

Development of a numerical flow channel with the Lattice Boltzmann method and application to highly swept wings at high angles of attack

vorgelegt von

M. Sc.
Justus Benad

an der Fakultät V – Verkehrs- und Maschinensysteme
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften

Dr.-Ing.

genehmigte Dissertation

Promotionsausschuss:

Vorsitzende:	Prof. Dr.-Ing. Sandra Klinge
Gutachter:	Prof. Dr. rer. nat. Valentin L. Popov
	Prof. Dr. Alexander E. Filippov

Tag der wissenschaftlichen Aussprache: 16. Dezember 2021

Berlin 2022

Deutsche Zusammenfassung

In der vorliegenden Arbeit wird ein numerischer Strömungskanal mit der Gitter-Boltzmann-Methode entwickelt und zur Untersuchung verschiedener stark gepfeilter Flügelgeometrien bei hohen Anstellwinkeln genutzt. Zu Beginn der Arbeit wird die Gitter-Boltzmann-Methode zusammen mit anderen numerischen Methoden diskutiert und im Hinblick auf derzeitige Trends bei der Entwicklung von Mikroprozessoren eingeschätzt. Es wird betont, dass numerische Techniken wie die Gitter-Boltzmann-Methode, welche auf einfachen parallelisierbaren Operation im Raum zurückgreifen, von den derzeitigen Entwicklung von Mikroprozessoren profitieren werden. Anschließend wird der Aufbau des numerischen Experiments mit der Gitter-Boltzmann-Methode beschrieben. Ein besonderer Fokus dabei liegt auf den numerischen Details welche die maximale Reynoldszahl beeinflussen, die simuliert werden kann. Drei exemplarische Studien werden mit dem entwickelten Aufbau durchgeführt: Als erstes wird ein endlicher ungepfeilter Flügel bei verschiedenen Anstellwinkeln untersucht. Danach wird der Einfluss von Pfeilung auf einen Flügel mit geringem Seitenverhältnis bei hohen Anstellwinkeln untersucht. In einer dritten Studie wird eine hochaufgelöste Simulation an einer exemplarischen Flying V Geometrie durchgeführt um einen ersten Eindruck des Strömungsfeldes hinter der Konfiguration bei hohen Anstellwinkeln zu erlangen. Dieser Studie wird bei einer Reynoldszahl von $Re = 4.6 \cdot 10^3$ und einem Anstellwinkel von $\alpha = 30^\circ$ durchgeführt. Resultate dabei sind unter anderem die Beobachtung von Instabilitäten in der Grenzschicht welche an der runden stark gepfeilten Vorderkante der Geometrie entstehen und sich nach innen hin zur Hinterkante des Flügels weiterbewegen, die Entdeckung von zwei charakteristischen freien Scherschichten auf der Oberseite des Flügels, die Beobachtung starker Interaktionen der Grundströmung und der zwei Scherschichten nahe der Flügelhinterkante, und die Entdeckung charakteristischer kohärenter Strukturen im Feld der Wirbelstärke nach einer starken Böe.

Abstract

In this work, a numerical flow channel is developed with the Lattice Boltzmann method and applied to study various swept wing geometries. At first, the Lattice Boltzmann method is discussed in the light of other numerical tools, and in the light of current development trends of microprocessors. It is concluded that numerical tools which consist of simple parallel operations in the spatial domain like the Lattice Boltzmann method are likely to benefit strongly from current development trends of microprocessors in the future. It is then decided to use the Lattice Boltzmann method for the creation of the numerical flow channel. The setup of this numerical experiment is explained in rich detail with a particular focus on the numerical details which influence the maximum Reynolds number which can be achieved. Finally, three exemplary studies are conducted with the developed setup. First, a finite unswept wing is investigated at various angles of attack. After this, the influence of sweep of a low aspect ratio wing at high angles of attack is investigated. In the third study, a high resolution simulation of an exemplary Flying V geometry is conducted in order to obtain a first impression of the flow field behind the configuration at high angles of attack. The results of this study are obtained at a Reynolds number of $Re = 4.6 \cdot 10^3$ and an angle of attack of $\alpha = 30^\circ$. Findings include the observation of instabilities in the boundary layer which form at the round highly swept leading edge and move inward towards the rear kink, the discovery of two characteristic free shear layers on the upper surface of the design, the observation of strong interactions of the main flow and the two shear layers close to the rear kink, and the discovery of turbulent coherent structures in the vorticity field after a strong initial gust.

Acknowledgements

I would like to express my great appreciation to Prof. Dr. Popov from the Department of System Dynamics and Friction Physics of the Technical University of Berlin for the supervision of this thesis, for his steadfast support, and for his encouragement to write about a topic which has long fascinated me. Further, I would like to thank the entire team of researchers at the Department of Friction Physics for a great working environment and much support over the past years. Many thanks also to all of my students. I have learned a lot from them over the past years.

I would like to offer my special thanks to various cooperation partners during my time as research assistant. I am extremely grateful for Dr. Roelof Vos and his team at Delft University of Technology for their continuation of the Flying V project. Their work was an inspiration to me and brought my old project back to me. I would also like to thank partners at Rolls-Royce Deutschland for a great cooperation project during my time at the TU Berlin, and I would like to thank Prof. Ken Nakano and his team at Yokohama National University for an inspiring cooperation over the past years.

Furthermore, my thanks goes to the great teachers I had in physics and engineering. I would like to thank Tom Swayne, Klaus Viebranz, Prof. Popov, Jean Roeder, and Klaus Bender.

Finally, I would like to thank my friends and family because nothing in this work would have been possible without their support, motivation, and enthusiasm. Most of all, I would like to thank Lisa Benad for her strong support throughout this work.

Contents

1	Introduction and motivation	1
2	Numerical techniques with local interactions in the spatial domain – four exemplary models	7
2.1	A single degree of freedom.....	8
2.1.1	Exemplary model	8
2.1.2	Discussion	12
2.1.2.1	Simplicity	12
2.1.2.2	Dimensionless formulation	13
2.1.2.3	Parallelization.....	13
2.2	A line of independent degrees of freedom.....	14
2.2.1	Exemplary model	14
2.2.2	Discussion	17
2.2.2.1	Parallelization.....	17
2.2.2.2	Simplicity	17
2.2.2.3	Problems.....	18
2.3	A 2D-array of degrees of freedom with local interactions	22
2.3.1	Exemplary model	23
2.3.2	Discussion	26
2.3.2.1	Simplicity	26
2.3.2.2	Parallelization.....	27
2.3.2.3	Further developments.....	27
2.4	A 3D-array of degrees of freedom with local interactions	28
3	Setup of the numerical flow channel.....	31
3.1	High fidelity model.....	31
3.2	The Buckingham pi theorem	32
3.3	Wind tunnels.....	34

3.4	Experimental setup	35
3.4.1	Dimensionless framework.....	36
3.4.2	Geometry.....	37
3.4.3	Available RAM and maximum Reynolds number.....	41
3.5	Algorithm.....	42
3.5.1	Initialization	43
3.5.2	Main algorithm.....	45
3.5.2.1	Dimensionless density and velocity field.....	45
3.5.2.2	Output.....	46
3.5.2.3	Equilibrium particle distribution	47
3.5.2.4	Collision	47
3.5.2.5	Streaming	48
3.5.2.6	Application of boundary conditions.....	48
3.5.2.6.1	Inflow	48
3.5.2.6.2	Wing.....	49
3.5.2.6.3	Tunnel walls	50
3.5.2.6.4	Outflow	50
3.5.3	Comments	51
3.6	Output	54
3.6.1	Vorticity field.....	54
3.6.2	Aerodynamic coefficients	55
4	Results	57
4.1	Finite wing	57
4.2	Influence of sweep.....	62
4.3	High resolution simulation	67
5	Conclusion and outlook.....	76
	References	78

1 Introduction and motivation

It is a warm summer day at Faßberg Air Base in Germany on July 14, 2020. A light breeze of air is blowing over fields of grass. A thin strip of concrete lies in the middle of this landscape. At one end, a small team of researchers have assembled, all of them wearing yellow vests which stand out, even from a far distance away. They are all gathered around a curiously shaped object in bright blue colors. As one of them moves away, the shape becomes more visible. Glistening in the sunlight, a scale model of a highly swept flying wing is being prepared to take off. Minutes pass, the researchers are busy on the model and on their computers. A team of cameramen have set out and prepare their equipment along the runway. Suddenly, the sound of powerful electric engines cuts through the silence. The airplane begins to accelerate down the runway. It gets faster and faster, and then, it begins to rotate and is off the ground. It rises quickly and enters a smooth right turn.



Figure 1: Take-off of a scale model of a Flying V aircraft configuration at Faßberg Air Base, Germany on July 14, 2020.

The scenario above refers to a test flight of a scale model of a Flying V aircraft configuration at Faßberg airbase in Germany in 2020. The Flying V is a new concept for an efficient aircraft. In that configuration, the pressurized passenger and cargo sections are arranged in

the shape of a V and located within a highly swept wing. Transition and outer wings extend the span of this wing at a lower sweep angle, see [Figure 2](#).

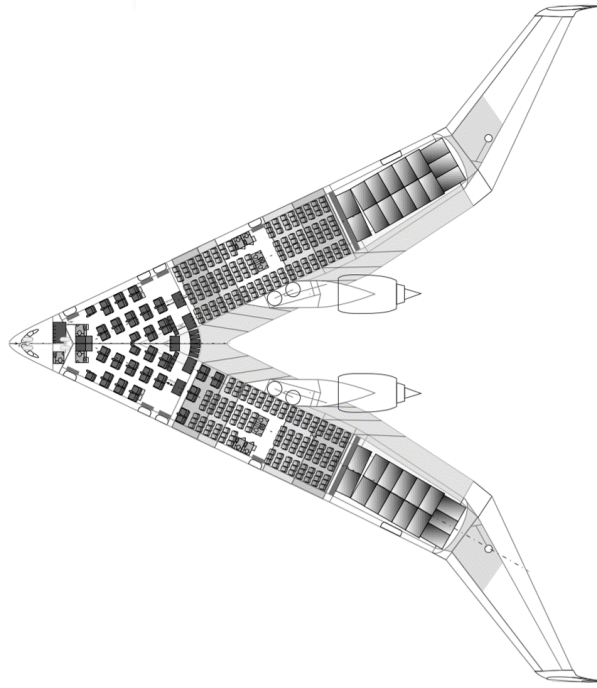


Figure 2: A Flying V aircraft configuration. Displayed here is an image from a recent study of a Flying V family concept ([Oosterom 2021](#)). A FV-1000 version is shown seating 378 passengers in a two class layout.

The Flying V concept was introduced in ([Benad 2014](#)) and ([Benad 2015, 1-3](#)). A subscale flight model with a wingspan of $b = 1.4$ m was built and flown demonstrating good handling qualities. In ([Faggiano, Vos et al. 2017](#)), a 25% higher lift to drag ratio was found for the Flying V when compared to the NASA common research model based on parametric optimization with Euler CFD. Estimation of structural weight in ([Claeys 2018](#)) showed a 17% decrease in “FEM weight” compared to A350-like aircraft based on an automated structural sizing algorithm coupled to a FEM solver. In ([Rubio Pascual and Vos 2020](#)), a feasible region for engine location on the Flying V was identified based on Euler CFD computations to minimize adverse aerodynamic interference between wing and engine. Works on the interior design of the Flying V were performed in ([Vink, Rotte et al. 2020](#)). Wind tunnel experiments on a 4.6%-scale half model were conducted in ([Palermo and Vos 2020](#)). The results showed that the aircraft can attain a maximum lift coefficient of $C_{L,max} = 1.02$ at an angle of attack of $\alpha = 35^\circ$. The pitching moment is negatively correlated to the angle of attack up to $\alpha = 19^\circ$. After this, a strong pitch-break occurs, making the aircraft statically unstable. The effectiveness of the control surfaces was found to be almost uninfluenced by the angle of attack. An aerodynamic model based on the wind

tunnel experiments was identified in (Ruiz Garcia, Vos et al. 2020). In (Viet 2019), oil flow visualizations on a 4.6%-scale half model demonstrated vortex formation for angles of attack above $\alpha = 10^\circ$ with varying patterns for increasing angle of attack. Large-Eddy CFD simulations conducted at (mvAero 2019) by van Egmond showed vortex formation on a subscale model with a different pattern than in previous wind tunnel experiments. Analysis of a full scale model did not show vortex formation at angles of attack up to $\alpha = 20^\circ$. Further studies on the Flying V have been conducted on the lateral handling qualities of the concept (Cappuyns 2019), and on the influence of the ground effect on the concept (Ankith John Santosh 2020). Further work on the engine integration of the Flying V has been performed in (Van Empelen and Vos 2021). Evacuation of the Flying V has been modelled in (Gebauer and Benad 2021) and (Hellmann 2020). A scale model of the Flying V with a span of $b = 3.06$ m was built by Brown, Ruiz García and Atherstone and flown successfully in 2020 (see Figure 1). A three-member family design for the Flying V was investigated in (Oosterom 2021). Various further research projects on the configuration are currently ongoing in fields such as aerodynamics, structures and manufacturing, flight dynamics and control, the environmental impact of the design, aircraft integration and airport operation.

The work in this particular thesis falls within the field of aerodynamic investigations of the unconventional Flying V shape at high angles of attack. On a conventional aircraft, high lift devices are used to change the shape of the wing at take-off and landing to generate the necessary lift at low speeds. The Flying V has a much larger wing area than a conventional aircraft of similar size. Therefore, increases in the angle of attack alone at low speeds may be sufficient to generate the necessary lift at low speeds.

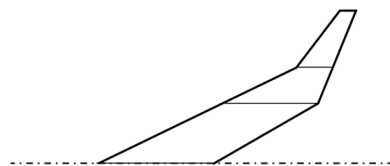


Figure 3: Characteristic geometry of a Flying V wing with a highly swept inner wing section, and an outer wing with lower sweep. The transition wing element in between has a leading edge sweep which follows the leading edge sweep of the inner wing, and a trailing edge sweep which follows the trailing edge sweep of the outer wing.

Yet, the aerodynamics at high angles of attack on a highly swept cranked wing of the shape it is used on the Flying V are to a large extent uninvestigated. In the history of aerospace,

there are virtually no planform shapes similar to that of the Flying V. The closest may be shape of an Arado E.555, a design from World War II, see (Herwig and Rode 2002). This flying wing also has a swept center wing and outer wings with lower sweep. Yet, there is no transition wing element between these two wing elements, as is characteristic for the Flying V. Also, with a sweep of about 45° , the center wing is not highly swept on the Arado design. The center wing sweep of the Flying V exceeds 60° . The Arado design was never built, nor is there any research data available. Examples for intensively studied wing planforms from the history of aerospace which share certain design features with the geometry of the Flying V are displayed in Figure 4.



Figure 4: Planform designs of examples for a) Blended wing body, b) Flying V, c) Delta wing, d) a highly swept wing.

Delta wings (Figure 4c) share the high leading edge sweep of the Flying V. The first Delta wings were built by Alexander Lippisch around 1940 (Storck 2003). Numerous airplanes have been designed with this wing layout throughout the following years, an example is the Convair F-102A. Generally, Delta wings have a sharp leading edge. The flow around Delta wings at high angles of attack is well understood, see for example (Erickson 1995, Cummings, Forsythe et al. 2003, Anderson 2017). Most characteristic are the leading edge vortices which form at the highly swept and sharp leading edge of the design. This is often desired for the creation of additional lift at high angles of attack. Although the Flying V shares the high leading edge sweep with the Delta wing, the Flying V has a thick and round leading edge. This is a major difference of the Flying V to common Delta wing designs and it will lead to different flow phenomena.

Blended wing bodies (Figure 4a) were intensively studied in (Liebeck, Page et al. 1998, Wakayama and Kroo 1998, Liebeck 2004, Qin, Vavalle et al. 2004). Aerodynamics at high angles of attack of the concept were investigated in (Vicroy 2009, Wisnoe, Nasir et al. 2009, Goldthorpe, Rossitto et al. 2010). Although the blended wing body shares some features with the Flying V, such as thicker profiles with a rounded leading edge in the middle section, the overall arrangement and planform still varies to a large extent, especially at the rear of

the design. Here again, different flow phenomena can be expected compared to the Flying V design.

Results of studies on highly swept wings (Figure 4d) show similarities with results of recent studies on the aerodynamics of the Flying V. Lift to drag relations obtained in wind tunnel experiments conducted on a 4.6% scale model of the Flying V in (Van Empelen and Vos 2021) are similar to experimental results obtained in (Cohen and Jones 1960) on a highly swept wing model as displayed in Figure 4d.

None of the geometries shown above displays all of the features of the highly swept cranked wing shape of the Flying V and while studies exist on the aerodynamics of the Flying V at high angles of attack, the understanding of the entire flow field behind the Flying V is still imprecise. Especially the vortex patterns which occur require further investigations. Indeed, as mentioned above, Large-Eddy CFD simulations have shown different results for vortex patterns than were obtained in wind tunnel experiments. In order to develop a full understanding of the flow behind the Flying V at high angles of attack there is a need for various more wind tunnel experiments and various numerical simulations. In order to understand Reynolds number effects, simulations should be conducted at a wide range of Reynolds numbers. Various research projects in this field are currently ongoing.

In this particular work, we model the flow around an exemplary highly swept cranked wing of a Flying V shape at a high angle of attack in a Lattice Boltzmann fluid. This investigation falls under the category of direct numerical simulations (DNS) at low Reynolds numbers. While this may help to develop an understanding of the flow field, let us emphasize strongly that further simulations with other models at high Reynolds numbers are required to assess the aerodynamics of the Flying V further.

In the present approach with the Lattice Boltzmann method, one models the flow directly and there are no simplifications to the flow as they occur in the Reynolds averaged Navier Stokes equations (RANS) or Large Eddy simulations which are commonly used to simulate flows at higher Reynolds numbers (Ferziger, Perić et al. 2002). The direct formulation used in this work may help to develop an understanding of the flow field around a shape which is to a large extent uninvestigated. It may be possible to draw comparisons with results

from other studies currently ongoing. It is also interesting to note that the tool which is developed in this work is fully scalable and can be applied without any further modifications on larger computers than they were used in the present work.

An additional thrust of this work, besides gaining further understanding of the Flying V, is to give an illustration of the simplicity and ease of use of the Lattice Boltzmann method.

The Lattice Boltzmann method has become a vast field of research over the past 30 years. (He and Luo 1997, Chen and Doolen 1998, Kang, Zhang et al. 2002, Aidun and Clausen 2010, Mohamad 2011, Guo and Shu 2013, Krüger, Kusumaatmaja et al. 2017). Researchers and engineers are drawn to this method due to its simplicity, and the way it allows for parallelization of computations.

We devote the next chapter to the introduction of the Lattice Boltzmann method from a technical perspective. Thereby, it shall be illustrated that the tool has a simple framework which is well suited for parallelization. This is shown by reviewing some equally simple numerical tools and by comparison of these tools with the Lattice Boltzmann method. It is highlighted that the simple parallelization of the Lattice Boltzmann method is well-suited to the current development trends in microprocessors. Thereby, it shall be illustrated that interest in the Lattice Boltzmann method is likely to increase further in the future.

After this first excursion, we return to our problem at hand and begin to develop the numerical flow channel to study highly swept wings at high angles of attack with the Lattice Boltzmann method. First, the setup of the numerical experiment is explained in detail with a particular focus on the numerical details which influence the maximum Reynolds number which can be achieved. The developed setup will be used to conduct three exemplary studies. First, a finite unswept wing is investigated at various angles of attack. Then, the influence of sweep of a low aspect ratio wing is investigated at high angles of attack. In the third study, a high resolution simulation of an exemplary Flying V geometry is conducted at a high angle of attack.

2 Numerical techniques with local interactions in the spatial domain – four exemplary models

Before we begin our numerical investigation of flows around wings, we will first introduce four exemplary numerical techniques. What all of these techniques share, is that the degrees of freedom in the spatial domain are decoupled, or are at least only influenced linearly by their direct neighbors. In this work, we call this “local interaction in the spatial domain”. The Lattice Boltzmann method which is applied for the creation of the numerical wind tunnel in this work, is an example of such a technique. Indeed, its local formulation is one of the major advantages of the method. Therefore, it is important to understand numerical techniques with local interactions in the spatial domain. For that sake, we will begin this work by presenting four such techniques. All of them were studied, implemented, and applied by the author of this thesis as a researcher at the Berlin University of Technology Institute of Mechanics.

The four exemplary models we will introduce are as follows:

1. A single “discretization point” in the spatial domain. Example: Normal and sideways oscillation of a single spring element
2. A line of independent degrees of freedom. Example: The Method of Dimensionality Reduction
3. A 2D-array of degrees of freedom with local interactions. Example: An evacuation simulation with cellular automata
4. A 3D-array of degrees of freedom with local interactions. Example: The Lattice Boltzmann method

After we have introduced each technique, we will enter a short discussion about the model. This way, we will be able to appreciate how these techniques are connected and what their advantages are. The entire chapter leads up to the introduction of the Lattice Boltzmann

method in the fourth section. There, it will be discussed in light of the previous three models and in light of current development trends in microprocessors.

2.1 A single degree of freedom

Let us organize the collection of tools we are about to discuss by the number of degrees of freedom in the model and how they are arranged. From a technical point of view, we begin with the simplest model of a single degree of freedom.

This is an almost trivial case, because there are no neighbors in the spatial domain to a single discretization point. Yet, we still include such a simple example at the beginning of this chapter. A discussion will follow afterwards.

Any simple Euler integration along a single thread in time could be used. The example which will be used here is the numerical model for a single spring whose motion is controlled at one end, while the other end may stick, slip, or jump on a surface. This model was investigated in (Popov, Popov et al. 2017, Benad, Nakano et al. 2018, Benad, Popov et al. 2018). We follow along with (Benad, Popov et al. 2018) in the next section.

2.1.1 Exemplary model

Consider an elastic body that is brought into contact with a flat elastic substrate and then subjected to a horizontal sliding movement with constant velocity which is superimposed, in case (I), with normal oscillations, and in case (II), with sideways oscillations. For the sake of a first simple investigation, it is possible to model this setup *as a single spring* with a normal stiffness k_N and a tangential stiffness k_T which is pressed onto a rigid plane. At the end of this section there will be a discussion about the implications of the simplification. The upper point P of the spring is subjected to a constant velocity v_0 in a horizontal direction. For the two cases of the investigation, the motion of the upper point P is superimposed with either normal or sideways oscillations. Between the lower point Q of the spring and the plane we assume that there is a friction force described by the simplest form of Coulomb's law with a constant coefficient of friction μ_0 .



Figure 5: An elastic body modeled as a spring with a normal and tangential stiffness is forced into a controlled movement at the upper point P, while the contact point Q follows according to the equilibrium conditions. Left: Schematic representation of the considered system for the case of normal oscillations, right: schematic representation of the considered system for the case of sideways oscillations. Images: (Benad, Popov et al. 2018)

A schematic view of the investigated system for the case (I) of normal oscillations is given in left image of Figure 5. The motion of the upper point of the spring P shall be given with

$$\begin{aligned} x_P &= v_0 t, \\ z_P &= z_N - z_0 \cos \omega t. \end{aligned} \quad (1)$$

As reference state, the unstressed state of the spring in the first moment of contact is chosen. In this state, the upper point P lies on the x -axis.

The motion of the immediate contact point Q is fully determined by l , which is the spring length projected onto the sliding plane. It can be determined with the following relations: When the contact point is sticking, it is

$$\dot{l} = v_0. \quad (2)$$

This relation remains valid as long as it is

$$l < \mu_0 (k_N/k_T) (z_N - z_0 \cos \omega t). \quad (3)$$

When (3) is violated, the contact point starts to slide and it is

$$l = \mu_0 (k_N/k_T) (z_N - z_0 \cos \omega t). \quad (4)$$

This relation remains valid as long as it is

$$\dot{l} < v_0. \quad (5)$$

When (5) is violated a phase of sticking starts again. An exception to the scheme given with relations (2)–(5) is when the point Q is jumping and not in contact at all. These phases always occur at times when the condition

$$\cos \omega t < (z_N/z_0) \quad (6)$$

is violated. During such times it is

$$l = 0. \quad (7)$$

The equations (2)–(7) which fully determine the motion of the system can be simplified with the following dimensionless variables and operators. It shall be

$$\tau = \omega t, \quad ' = \frac{d}{d\tau}, \quad \tilde{z}_0 = \frac{z_0}{z_N}, \quad \tilde{l} = \frac{l}{l_0}, \quad \tilde{v}_0 = \frac{v_0}{l_0 \omega}. \quad (8)$$

with $l_0 = z_N \mu_0 k_N / k_T$. The governing equations can now be rewritten: When the contact point is sticking, it is

$$\tilde{l}' = \tilde{v}_0. \quad (9)$$

This relation remains valid as long as it is

$$\tilde{l} < 1 - \tilde{z}_0 \cos \tau. \quad (10)$$

When (10) is violated, the contact point starts to slide and it is

$$\tilde{l} = 1 - \tilde{z}_0 \cos \tau. \quad (11)$$

This relation remains valid as long as it is

$$\tilde{l}' < \tilde{v}_0. \quad (12)$$

When (12) is violated, a phase of sticking starts again. The exception to this scheme of (9)–(12) is the jumping case which always occurs at times when the condition

$$\cos \tau < (1/\tilde{z}_0) \quad (13)$$

is violated. During such times it is

$$\tilde{l} = 0. \quad (14)$$

In the dimensionless governing equations (9)–(14) there are only two parameters, which are \tilde{z}_0 and \tilde{v}_0 . Therefore, *the motion of the system only depends on these two dimensionless variables.*

The macroscopic coefficient of friction is given by the average tangential force divided by the average normal force $\mu_{\text{macro}} = \langle k_T l \rangle / \langle F_N \rangle$. Using the dimensionless variables from above, it can be written as

$$\tilde{\mu}_{\text{macro}} = \frac{\mu_{\text{macro}}}{\mu_0} = \frac{\langle \tilde{l} \rangle}{\langle (|1 - \tilde{z}_0 \cos \tau| + 1 - \tilde{z}_0 \cos \tau)/2 \rangle} \quad (15)$$

and is therefore only a function of the dimensionless amplitude and the dimensionless sliding velocity:

$$\tilde{\mu}_{\text{macro}} = f(\tilde{z}_0, \tilde{v}_0). \quad (16)$$

On the left side of [Figure 6](#), this dependence is displayed as a contour plot and as a three-dimensional diagram.

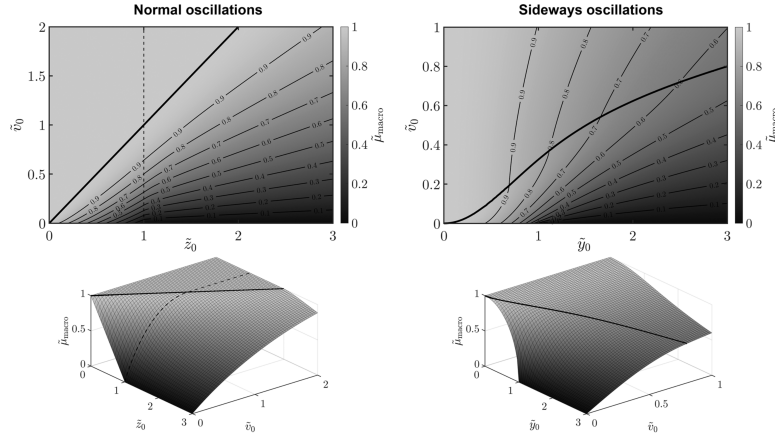


Figure 6: The macroscopic coefficient of friction as influenced by normal oscillations (*left*) and sideways oscillations (*right*). The results for the macroscopic coefficient of friction are displayed over the two system parameters, the dimensionless sliding velocity \tilde{v}_0 and the dimensionless oscillation amplitude (\tilde{z}_0 for normal, \tilde{y}_0 for sideways oscillations). In the upper graphs this dependence is shown as a 2D diagram in which the macroscopic coefficient of friction is displayed with a color scale and level lines. Each corresponding graph below gives an additional illustration of the dependence with a 3D surface in which the macroscopic coefficient of friction is displayed on the vertical axis. In all graphs, the region above the bold borderline is the region of continuous sliding of the contact point, while in the region below this line the contact point undergoes a stick-slip motion. The additional dashed line in the left graphs separates the region of no jumping (*left of the dashed line*) from the region in which the contact point is jumping (*region to the right of the dashed line*) Images: [\(Benad, Popov et al. 2018\)](#)

A schematic view of the investigated system for the case (II) of sideways oscillations is given on the right side of [Figure 5](#). The motion of the upper point of the spring P shall be given with

$$\begin{aligned} x_P &= v_0 t, \\ z_P &= z_N - z_0 \cos \omega t. \end{aligned} \quad (17)$$

The derivation for the macroscopic coefficient of friction runs similar to the derivation for normal oscillations, for details, see [\(Benad, Popov et al. 2018\)](#). With

$$\tau = \omega t, \quad ' = \frac{d}{d\tau}, \quad \tilde{y}_0 = \frac{y_0}{l_0}, \quad \tilde{l} = \frac{l}{l_0}, \quad \tilde{v}_0 = \frac{v_0}{l_0 \omega}. \quad (18)$$

and $l_0 = \mu_0 F_N / k_T$ we find again that the macroscopic coefficient of friction $\tilde{\mu}_{\text{macro}} = \mu_{\text{macro}} / \mu_0$ is only a function of the two system parameters:

$$\tilde{\mu}_{\text{macro}} = f(\tilde{y}_0, \tilde{v}_0). \quad (19)$$

This dependence is displayed as a contour plot and as a three-dimensional diagram on the right side of Figure 6. An additional schematic representation of the dependence is displayed in Figure 7. Here, dimensional quantities and relations are shown. Various results obtained for limiting cases in (Benad, Nakano et al. 2018) are displayed drawn into the parameter plane. Also, at four exemplary points, the sliding motion of the upper and lower point of the model are displayed.

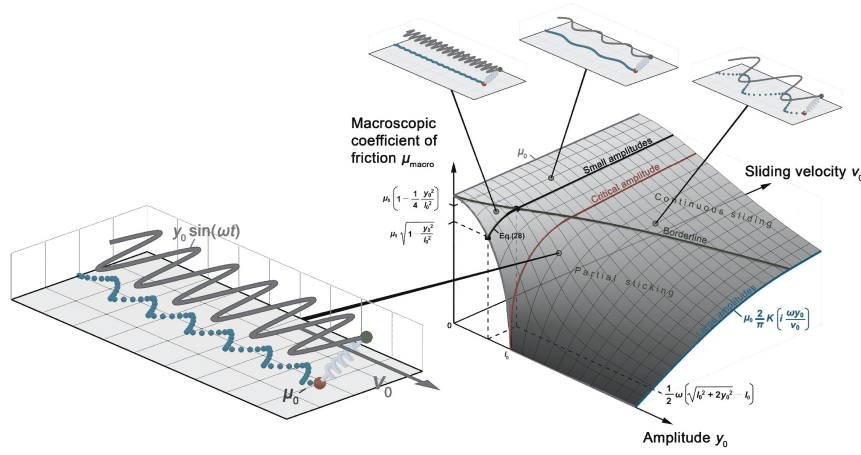


Figure 7: Schematic representation of the parameter plane for the case of sideways oscillations. Relations obtained in (Benad, Nakano et al. 2018) for limiting cases are drawn into the parameter plane.

2.1.2 Discussion

Let us now focus on several different aspects of the exemplary model which was just introduced.

2.1.2.1 Simplicity

Note first, that here we have a simple setup which contains just as much detail as necessary to include all parameters which are of interest. This should not be underestimated. Especially from a technical point of view the numerical model should be kept as simple as possible. Often, one may already see important trends and the risk of errors is low. Also, one already has a simple model to use as a starting point and for comparison with more

sophisticated models. For example, for the model above by author of this thesis, a more detailed follow up investigation was performed in (Pohrt 2020). In this study, the spatial resolution of the problem is increased from a single contact point, to an entire array of points which may either slide or stick on the surface. The investigation was carried out using a Boundary Element Method (Pohrt and Li 2014). This far more sophisticated tool confirmed the dependencies and trends of the present one-spring model with surprising accuracy, see for example Figure 8.

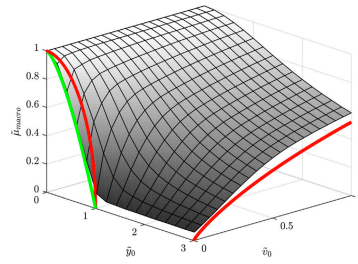


Figure 8: Results of a refined model of sideways oscillations by (Pohrt 2020) obtained with an spatial resolution of 32 x 32 contact points to include partial stick and slip. The limiting cases of the one-spring model are drawn into the diagram with red lines. The green line represents a more accurate estimation for small sliding velocities from the refined study. The red line on the right is approached for large oscillation amplitudes also in this more refined model. The gap in the diagram is only visible because the oscillation amplitudes are not displayed up to higher values in the diagram.

2.1.2.2 Dimensionless formulation

The second aspect we want to draw attention to is the dimensionless formulation of the tool above. In the dimensional governing equations in the model above there are eight parameters whose influence on the motion of the system we would have to investigate. The dimensionless formulations using (8) and (18) simplify this process. Note that the here found dimensionless style has been adopted in the follow up study by (Pohrt 2020). For such more sophisticated systems, the dimensionless style may not be immediately obvious without previous investigations such as the simplified one above. A formal procedure to find dimensionless formulations for more complicated systems is embodied in the Buckingham pi theorem which we will apply later for our investigation of flows around wings.

2.1.2.3 Parallelization

The third important aspect we want to highlight is parallelization. The model above is an example of a non-linear system. It can be solved numerically by integration in the time

domain. There is no explicit solution to the governing equations and the solution of the system in time can only be obtained by the knowledge of the status of the system in the past, which makes parallelization along the dimension of time impossible. All of the following numerical tools will have a higher spatial resolution than “one” and for them, parallelization will play an important role in the spatial domain. For the tool above, it was possible to make use of parallelization only in a more technical framework when conducting the parameter studies to obtain the parameter planes as they are displayed in [Figure 6](#) and [Figure 7](#).

2.2 A line of independent degrees of freedom

Let us now move on to a one-dimensional arrangement of independent degrees of freedom. In the exemplary model below this will be a Winkler foundation of independent spring elements. In examining this model, we will be able to highlight the benefits of parallelization through vectorization in the spatial domain. The exemplary model, to a certain extent, allows for this. Yet, the model will also offer us the opportunity to examine some cases where difficulties are encountered when the degrees of freedom cannot be treated independently.

2.2.1 Exemplary model

The exemplary model of this section will be the Method of Dimensionality Reduction (MDR). This is a simple tool for the calculation of contact forces between elastic and viscoelastic bodies. It is particularly easy to use for the simulation of axially symmetric contacts. Since it was first proposed in [\(Popov and Psakhie 2007\)](#) the MDR has been applied to a wide range of problems. The method maps a given three-dimensional contact problem to an equivalent contact problem of a transformed indentation profile with a one-dimensional elastic or viscoelastic foundation of independent elements. From a numerical perspective, the solution of the contact problem in the transformed MDR domain is fast and convenient due to the independent degrees of freedom in this domain.

The advantage of independent degrees of freedom is that operation can be performed parallel in the spatial domain. For an example, consider a two-dimensional profile in contact with a Winkler foundation, as displayed in [Figure 9](#).

Operations which have to be performed to determine which points are in contact, or how high the indentation of each point is, are local operations which do not depend on each other. This allows for parallelization. Programming languages offer convenient ways to deal with such problems. In the present case, for example, the indentation of the individual points can be obtained using a variety of *vectorized operations*, which are simple operations for arrays implemented to run parallel on multiple CPU threads or on GPUs. In the present example, let g be the one-dimensional array which contains the shape of the profile in [Figure 9](#). Let us also introduce the array $g_n = g - \min(g)$. If we then denote the indentation of the lowest point of the profile g_n with d , we find the indentation of all points of the profile which we save in the one-dimensional array $w1d$ with

$$w1d = (\text{abs}(d - g_n) + d - g_n) / 2. \quad (20)$$

Relation (20) contains only simple vectorized operations. This style can be adopted for all further operations in the spatial domain. Exemplary studies which apply this technique are ([Benad 2012](#), [Popov and Benad 2013](#), [Benad 2018](#)). For such simulations with many discretization points, it is essential that the core of the simulation can be vectorized in order to obtain results in an acceptable timeframe. For the Winkler foundation model, this is possible due to the decoupled degrees of freedom with techniques such as the one illustrated in (20).

In 2018, a study was performed by the author of this thesis on some of the numerical details of the MDR when applied to rotationally symmetric geometries ([Benad 2018](#)). The full MDR algorithm for the application on rotational symmetric geometries is described in rich detail for example in ([Willert 2020](#)). In the method, three transformations occur: The transformation of the three-dimensional profile $f(r)$ to a one-dimensional profile is

$$g(x) = |x| \int_0^{|x|} \frac{f'(r)}{\sqrt{x^2 - r^2}} dr, \quad (21)$$

the transformation of the one-dimensional foundation displacement $w_{1D}(x)$ to the three-dimensional normal surface displacement $w(r)$ is

$$w(x) = \frac{2}{\pi} \int_0^r \frac{w_{1D}(x)}{\sqrt{r^2 - x^2}} dx, \quad (22)$$

and the transformation of the one-dimensional force density $q(x)$ to the three-dimensional pressure distribution $p(r)$ is

$$p(r) = -\frac{1}{\pi} \int_r^\infty \frac{q'(x)}{\sqrt{x^2 - r^2}} dx. \quad (23)$$

Figure 9 shows results obtained in (Benad 2018) for a conic and parabolic indenter at an exemplary indentation depth d . It becomes apparent that already for as few as $N = 51$ discretization points a fairly good approximation of the analytical solutions can be achieved.

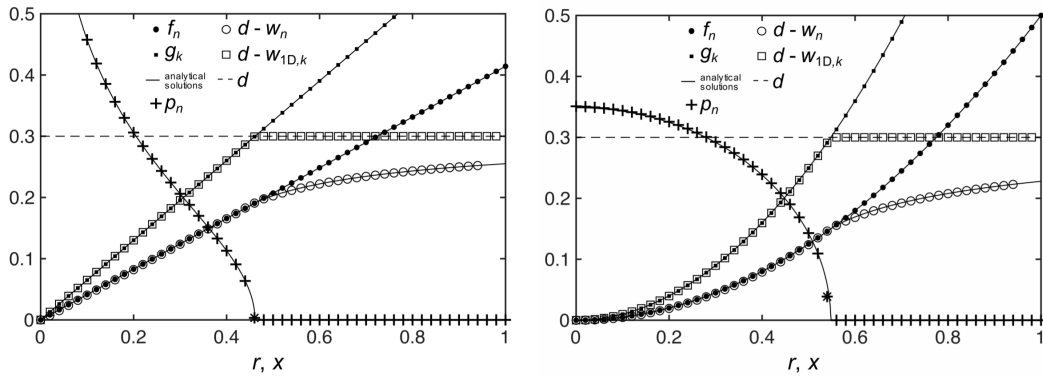


Figure 9: Results of the MDR transformations carried out with a numerical procedure described in (Benad 2018) for $N = 51$ discretization points, exemplary input parameters of $L = 1$, $E^* = 1$, $d = 0.3$ and an exemplary conic indenter (left) given with $f(r) = r \tan(\pi/8)$ and an exemplary parabolic indenter (right) given with $f(r) = r^2/2$. The pressure which is obtained at last discretization point within the contact area in this example is highlighted with a star. Image: (Benad 2018)

Before we enter the discussion, let us draw attention to one important aspect of this model: After the initial transformation (21) to the equivalent problem was performed, vectorization is possible. As shown in Figure 9, $g(x)$ simply has to be brought into contact with a Winkler foundation of independent spring elements. In this domain, we are free to use techniques such as (20). Various problems can be solved directly this transformed domain. If this is the case, a transformation such as (21) has to be performed only once at the beginning of the investigation. After this, the degrees of freedom of the problem are fully independent. However, there are also various problems where the transformations (21) to (23) have to be

repeated many times. This class of problems can be regarded as one where the degrees of freedom do depend on each other because the integrals (21) to (23) cannot be obtained using independent local operations in the spatial domain. In this case, numerical difficulties may be encountered in the method. We will discuss this aspect in rich detail in the following section.

2.2.2 Discussion

Let us now discuss several aspects of the described method above. Once again, we thereby hope to draw conclusions which will turn out to be useful in the development of the numerical flow channel in the subsequent sections.

2.2.2.1 Parallelization

Let us highlight this model once again as an example for simple parallelization of operations through vectorization in the spatial domain. As described in detail in [Section 2.2.1](#), vectorization in the spatial domain is easy to perform within the transformed MDR domain, when the profile is brought into contact with a Winkler foundation of independent spring elements.

2.2.2.2 Simplicity

As in [Section 2.1.2.1](#), let us first discuss the simplicity of the model at hand. The framework given with transformations (21) to (23) makes elegant use of the rotational symmetry of the problem. This causes a reduction of the degrees of freedom. As described above, in the transformed MDR space, a line has to be brought into contact with a one-dimensional Winkler foundation. Symmetry considerations will play an important role later in this work to reduce the number of degrees of freedom of a given problem.

As a further point, we should note that it is remarkable how well the simple MDR framework can be applied to practical problems, even if the contact problem under investigation is not fully rotationally symmetric. *Fabricants approximation*, as described in detail in [\(Barber 2018\)](#), makes it possible to approximate the solution of a non-rotationally symmetric contact with the solution of a rotationally symmetric problem. In two Master thesis projects supervised by the author of the present thesis in cooperation with

Rolls-Royce, results obtained with the MDR and Fabricants approximation are in very good agreement with results of finite element models and experiments: In (Diercks 2018), the MDR and Fabricants approximation returned the same results for the stress concentration in the contact region of turbine blade fir-tree connections as high fidelity finite element models, and in (Davison 2021), a wear model using Archard's law, the MDR and Fabricants approximation was sufficient to explain trends obtained with an experimental test rig for labyrinth seals in aero engines. With this knowledge, we can only repeat our conclusion from Section 2.1.2.1, which calls for a numerical model which should be kept as simple as possible. This way, one may already see important trends and the risk of errors is low. Also, one already has a simple model to use as a starting point and for comparison with more sophisticated models.

2.2.2.3 Problems

Problems in the numerical procedure are caused the moment the degrees of freedom start to depend on each other. For the present model, this is the case when transformations (21) to (23) have to be repeated many times, for example due to a continuously changing indentation profile as it appears in wear simulations, see (Dimaki, Dmitriev et al. 2014, Dimaki, Dmitriev et al. 2016, Li, Forsbach et al. 2018). The MDR transformations are given by Abel-like integral equations, and it is well known that their numerical treatment is challenging (Hansen and Law 1985, Murio, Hinestroza et al. 1992). From a technical point of view, the transformations state that the acquisition of a single transformed value in the one-dimensional array requires the knowledge of, at most, all other untransformed values in the array. In this work, we will call this a problem with *dependent degrees of freedom* in the spatial domain which require *non-local operations*.

We now follow along with (Benad 2018) to illustrate some of the difficulties which may be encountered with such problems. One technique for the implementation of the transformations (21) to (23) is a simple summation, as in

$$g_k = x_k h \left(\sum_{n=1}^{k-1} \left(\frac{f'_n}{\sqrt{x_k^2 - r_n^2}} \right) + \frac{f'_k}{\sqrt{h}} \right). \quad (24)$$

The singularity at $x = r$ is treated by the insertion of a single increment h . In (36), we consider a uniform discretization of $r \in [0, L]$ and $x \in [0, L]$ as shown in Figure 9 with N

points each and the same step size $h = \frac{L}{N-1}$, so that $r_n = h(n-1)$, $x_k = h(k-1)$, and $n, k \in \{1, 2, \dots, N\}$. Derivatives of a discretized indentation profile $f_n = f(r_n)$ can be obtained via central differences: $f'_n = (f_{n+1} - f_{n-1})/2h$, and $f''_n = (f_{n+1} - 2f_n + f_{n-1})/h^2$.

The method given with (36), however, delivers only very poor results when compared with other more sophisticated techniques. This can be seen in Figure 11, where the present technique is called “Method I”.

A far better technique for the implementation of the transformations is the use of an antiderivative. For the transformation to g_k , this translates to

$$g_k = x_k h \sum_{n=1}^k \left(\frac{\operatorname{atan} \frac{r_n}{\sqrt{x_k^2 - r_n^2}} - \operatorname{atan} \frac{r_{n-1}}{\sqrt{x_k^2 - r_{n-1}^2}}}{h} f'_n \right) \quad (25)$$

and for the transformation to p_n , one can use

$$p_n = -\frac{1}{\pi} h \sum_{k=n}^N \left(\frac{\log(\sqrt{x_{k+1}^2 - r_n^2} + x_{k+1}) - \log(\sqrt{x_k^2 - r_n^2} + x_k)}{h} q'_k \right) \quad (26)$$

The first derivatives can once more be obtained via central differences. As can be seen in Figure 11, this technique, referred to as “Method II” provides a much better accuracy than “Method I”.

The third technique which shall be mentioned here and called “Method III” is introduced and described in great detail in (Benad 2018). Please refer to this document for more information on this technique. The main idea is to avoid the singularity at $x = r$ through *partial integration* of the transformations (21) to (23). This leads to alternative formulations of the transformations in which the second derivative of the three-dimensional indentation profile and the deformed elastic foundation occur. Thus, singularities now occur at kinks of these profiles; however, they disappear in the numerical integration, similarly to “Method II” where the small increment h cancels out in equations (37) and (38). We shall also note that the singularity which is overcome in “Method II” occurs in the kernel. “Method III”, however, overcomes singularities which may occur through the shape of the indentation profile or the deformed one-dimensional foundation. Also, the singularity in “Method II”

always influences the transformation values at all discretization points whereas in “Method III” the singularities through kinks may leave transformation values at some discretization points uninfluenced. In [Figure 11](#), it can be seen that with “Method III” the number of discretization points can substantially be reduced to achieve the same accuracy as in “Method II”. However, it stands out that the maximum error in “Method III” is still fairly close to the maximum error in “Method II”. This relatively high maximum error of “Method III” is generally attained at the end of the contact area.

The previously described relatively high maximum error of “Method III” can be reduced in “Method IV” which is also introduced in [\(Benad 2018\)](#). Here, an additional discretization point is inserted at the end of the contact area. Three more points in the near surrounding have to be added to allow the computation of the derivatives. In [Figure 10](#), a detailed image of the additional points is displayed. In [Figure 11](#), it can be seen that techniques such as “Method IV” can help to further increase the accuracy of the method. Note that the particular technique “Method IV” with one additional discretization point is only the first step towards a more general refinement of the discretization towards the end of the contact area with additional points to increase the accuracy of the method even further.

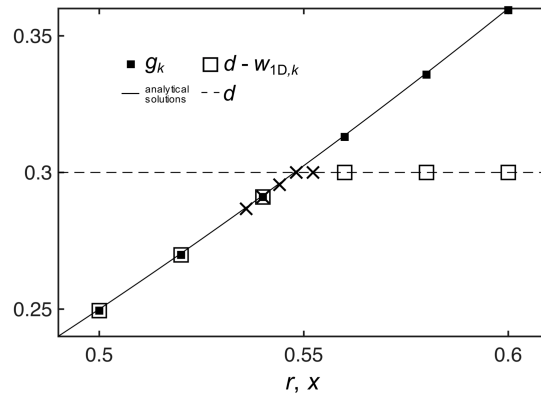


Figure 10: Detailed view of the graph in Fig. 1b, here with additional discretization points at the end of the contact area which are marked with black crosses. Image: [\(Benad 2018\)](#)

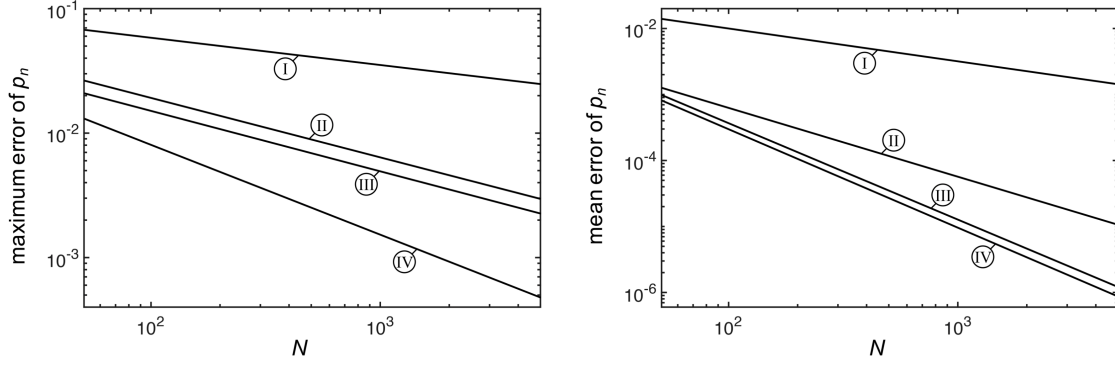


Figure 11: Upper limits of the maximum absolute error of p_n (left) and the mean absolute error of p_n (right) compared for the different numerical methods: “Method I” – insertion of h at singularity, “Method II” – implementation using the antiderivative, “Method III” – partial integration method, “Method IV” – partial integration method with small adjustment. As before, the curves are displayed for the exemplary inputs of $L = 1$, $E^* = 1$, $d = 0.3$ for the exemplary parabolic indenter given with $f(r) = \frac{r^2}{2}$. Image: (Benad 2018)

In addition to the investigation of the accuracy of the transformation methods it is also to investigate how they perform if they are used multiple times, for instance during wear simulations. It is interesting to note, that “Method II” seem to yield a high oscillating error during such simulations which cannot be seen with the newly introduced “Method III” and “Method IV”. To illustrate this behavior the corresponding image from (Benad 2018) is included below in Figure 12. For more details on this simulation please also refer to the complete paper.

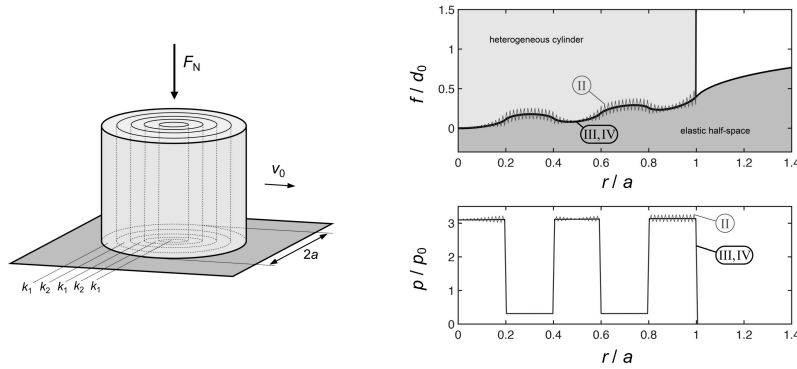


Figure 12: Left graph: A heterogeneous cylinder composed of rings of different material having the same elastic properties but different wear coefficients k_1 and k_2 is pressed onto an elastic half-space with the normal force F_N and moves tangentially with velocity v_0 . Right graph: Simulation results for the limiting profile and pressure after a long enough running-in process as obtained with Method II (a thin grey jagged line), and the techniques “Method III” and “Method IV” (smooth bold line) with $N = 201$ discretization points and a ratio of wear coefficients of $\frac{k_2}{k_1} = 10$. Image: (Benad 2018)

Let us now step back and examine the techniques which were just discussed. Although the problems which are encountered could be kept at bay, the numerical treatment was still somewhat challenging. Not only this, but also the computational effort is of course increased if the degrees of freedom depend on each other through transformations such as (21) to (23). This may not be challenging for a one-dimensional array such as in MDR simulations, but for more dimensional arrays as we will deal with in subsequent chapters such dependencies will indeed be difficult to handle. It is interesting to note, that although it is challenging, solutions may be found to mitigate such increases of computational effort. In a follow up study to the present model, see (Willert 2021), a way was found to apply the Fast Fourier Transformation to the transformations (21) to (23). This does not fully eliminate the rise in computational effort when compared operations with a single one-dimensional array of independent degrees of freedom, but it is still a large improvement when compared to the straight forward implementation of summations, as in (37) and (38).

As a closing statement of this discussion, let us emphasize that problems such as the one in this section with dependent degrees of freedom in the spatial domain requiring non-local operations seem to the author – although there may be ways to deal with them – to be extremely challenging and often difficult to implement.

Therefore, for the numerical flow channel which will be developed in subsequent chapters, we will use a different approach. We will apply a powerful procedure with *local interactions* in the spatial domain.

2.3 A 2D-array of degrees of freedom with local interactions

The third exemplary study which we will first follow closely and then discuss, is a recent preprint about the emergency evacuation of the novel Flying V aircraft configuration. With passenger compartments arranged in the shape of a V, the cabin geometry of a Flying V airplane differs significantly from the cabin geometry of the conventional tube and wing configuration (see Figure 13). Certification regulations state that the evacuation time of a civil passenger aircraft must not exceed 90 seconds when half of all doors are closed (CS-

25, (European Union Aviation Safety Agency 2020)). For flying wings in particular, this requirement has always been a topic of some concern (Martinez-Val 2007, Torenbeek 2013). An evacuation study of the Flying V was conducted by Julia Gebauer at the time a Master student at the Berlin Institute of Mechanics (Gebauer and Benad 2021). The work was supervised by the author of this thesis.

The inclusion of this study in the present will give us the opportunity to examine a simple model of a *two-dimensional* array of independent degrees of freedom. From a technical point of view, this is the next logical step after the previous two studies which we discussed.

Let us introduce the numerical model to investigate the emergency evacuation of the Flying V which was used in (Gebauer and Benad 2021). We follow along with this study in the next section. A detailed discussion of the model and how it relates to the numerical flow channel which will be developed in subsequent chapters of this thesis will follow after this.

2.3.1 Exemplary model

There are various techniques to simulate emergency evacuations, among them are cellular automaton models (Burstedde, Klauck et al. 2001), or the continuous social force model (Helbing and Molnar 1995). An exemplary evacuation software is airExodus, see (Galea, Blake et al. 2001). For a first preliminary analysis of the evacuation of the Flying V, a simulation tool was developed in (Gebauer and Benad 2021), which is the study we follow along with in this section. The developed tool is based on the technique of cellular automata with a floor field model, see (Burstedde, Klauck et al. 2001). Therein, a discrete domain is introduced, where each cell state can be empty (“zero”) or occupied (“one”). A passenger decides where to go by a probability calculated by layering different fields. In the present model, a single parameter r characterizing the level of random motion of the passengers during the evacuation process is introduced and calibrated to match evacuation times of existing airplane configurations. With the calibrated tool, multiple simulations are executed to compare the evacuation times of the Flying V and the Airbus A350-900 reference aircraft for different closed door configurations.

In order to create a simple simulation tool, cellular automata are used. The observed domain is the passenger cabin that is modeled by a grid consisting of square cells. The dimension for one cell was chosen with $40\text{ cm} \times 40\text{ cm}$ to model dimensions of seats and aisles as well as the space a pedestrian occupies, see (Torenbeek 1982, Weidmann 1993). The generated grids are shown in Figure 13.

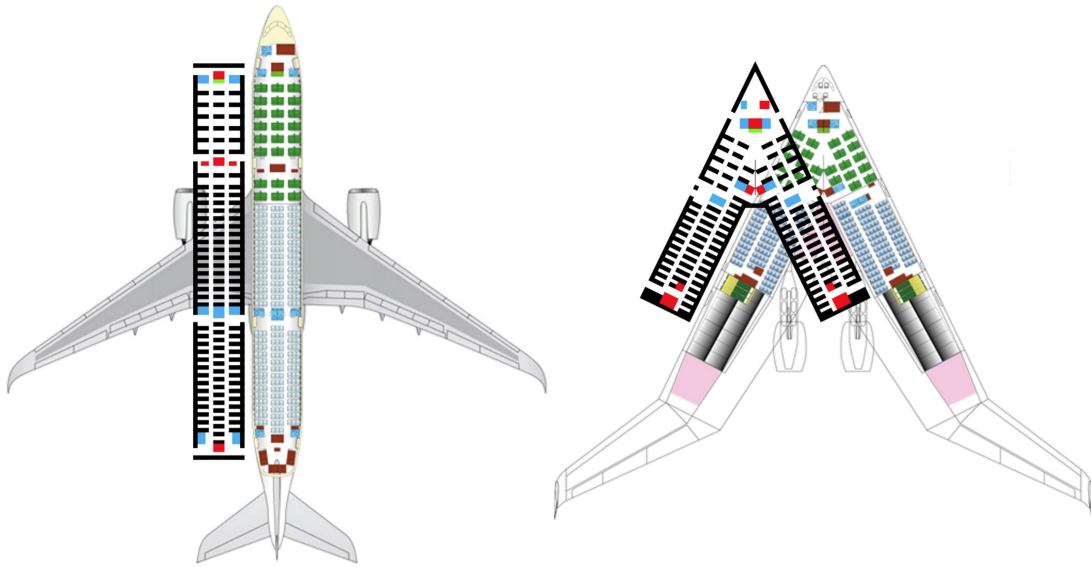


Figure 13: The cabin geometry of the Airbus A350-900 (left) and Flying V (right) modeled with a grid based on square cells.

The walls and seats receive the cell state “one”, which is permanent over the course of the simulation. The cell state for passengers changes over time. Only the closest adjacent cells are assumed to have an impact on a passenger in the present simulation. A single cell can only be occupied by one passenger in one time step. When multiple passengers have the same target cell, one passenger is chosen randomly. This passenger is allowed to move to this target cell while the other passengers are prohibited from moving at all. In order to define when a passenger can be seen as evacuated, boundary conditions need to be set. When a passenger enters an exit door, this passenger is considered evacuated and is ignored in the next time step. In addition, the transition between the legs of the Flying V is crucial. In the present preliminary model, two separate lattices are aligned with each leg of the V. Where both legs meet, transition conditions are applied.

The movement of a passenger depends on a transition probability p . In accordance with the floor field model (Burstedde, Klauck et al. 2001), this probability is calculated by layering different fields. Three different fields are taken into account: a gradient, distance and

direction field. The gradient field presents the urge of each passenger to reach the exit doors with the shortest way possible. The distance field adds an entirely random motion to the passengers. In the present model, its influence decreases linearly with the distance to each exit. The overall influence of the distance field can be adjusted by the single parameter s . Therefore, in the present study, this single parameter r is used to characterize the level of random motion of the passengers during the evacuation process. This parameter can be calibrated to match evacuation times of existing airplane configurations. Additionally, a correction field is applied to specific small areas with influence parameter q to guarantee that no passengers are not stuck in a dead end. The transition probability used in this simulation is

$$p_{ij} = \left[\left(r p_{d_{ij}} + (1 - r) p_{g_{ij}} \right) q + (1 - q) p_{c_{ij}} \right] (1 - w_{ij}) \quad (27)$$

with $i, j \in \{1, 2, 3\}$, where $p_{d_{ij}}$ represents the distance field, $p_{g_{ij}}$ the gradient field, $p_{c_{ij}}$ the correction field, and w_{ij} the wall grid, where a movement is prohibited. The variables i and j represent the adjacent cells that are considered for the calculation of the probability. In the preliminary model, all passengers will be moving with the same velocity of $v \approx 1.3$ m/s, which is the average velocity for a pedestrian (Weidmann 1993). In the present simulation, where a passenger walks with one cell per time step, this translates to a time step of approximately 0.3 s. In order to calibrate the simulation tool for the Airbus A350-900, data from trials or other evacuation models was researched. No values were found for the reference aircraft, but due to similarity in exit door arrangements and seat capacity, values presented in (Choochart and Thipyopas 2020) obtained from a simulation with airExodus applied to the Boeing 767 were used to calibrate the present preliminary model. From this study, a target evacuation time of 60 s could be derived for a case where all doors on the right side of the aircraft are closed. This time excludes the response time of crew members. Multiple simulations were run for the calibration. Based on the outcome of these simulations, the parameter was set to $r = 10^{-2}$. Various closed door configurations were examined for the Flying V and the reference aircraft. They are displayed in Figure 14.

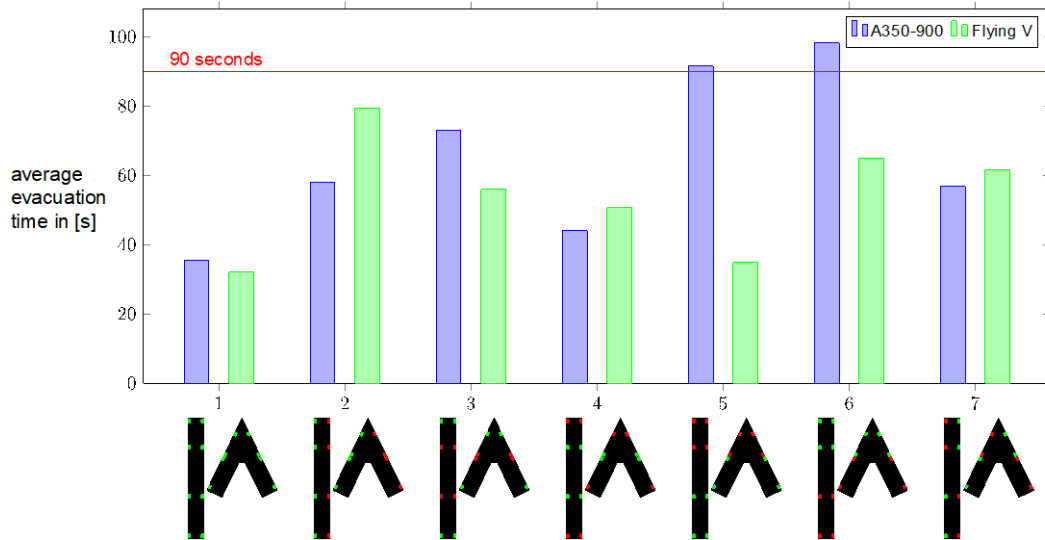


Figure 14: Average evacuation times for the Flying V and the reference aircraft displayed for various closed door configurations and a parameter $r = 10^{-2}$. Note that the displayed times are only of the evacuation process and exclude crew reaction times at the beginning of the evacuation. Note further, that these are preliminary results obtained with an extremely simple tool. One should exercise great caution with these results, especially with the actual quantitative values. The results indicate that the V shaped cabin has some advantages over the tube cabin if evacuation must take place only towards the front or only towards the rear of the aircraft (cases 5 and 6). For example, the tool showed a reduction in evacuation time of 62% for the Flying V when compared to the reference when half of all doors in the front of the aircraft are closed (case 5). When half of all doors in the back of the aircraft are closed (case 6), a reduction of 34% in evacuation time was obtained. This seems to indicate a similar trend as was obtained in a recent study (Isgrò 2020) where boarding times of the Flying V and the A350-900 were simulated using agent based modelling. In this study, a reduction of 30% in boarding time was obtained for the Flying V where passengers can proceed from the front to the back of the aircraft using four available aisles as opposed to two aisles in the reference aircraft. Results of the present study obtained for case 2 indicate that disadvantages in the evacuation process may occur when the passengers in the V shaped cabin need to evacuate solely towards one side of the aircraft. In this case, an increase of 37% in evacuation time for the Flying V was obtained when compared to the reference with the preliminary tool.

2.3.2 Discussion

Let us discuss the model from a technical point of view. Note that in the development of the model, we have already used some of the insights from the previous two discussions.

2.3.2.1 Simplicity

First, let us draw attention to the simplicity of the tool. Certification regulations state, that there must be a certain percentage of male and female passengers, old and young, etc. None of these aspects are accurately considered in the tool. All of this handled with the single parameter r which we use to calibrate the present simple model to match evacuation times obtained with more sophisticated models. Also, the cabin geometry is a far way from reality.

However, we have a first tool, which can serve as a first reference and as a starting point for more sophisticated studies.

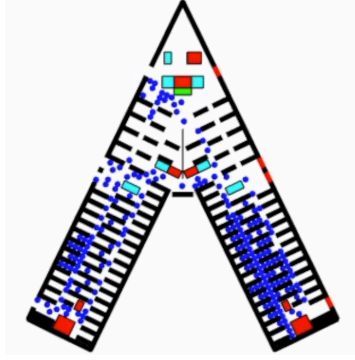
2.3.2.2 Parallelization

Second, with regard to parallelization, let us note that the model at hand can be fully vectorized in the spatial domain as described in previous discussions. In practice, this means that large numbers of simulations can be conducted and parameter studies can be carried out, for example to investigate various geometries. Simple vectorization is possible, first, because the decision a passenger makes depends only on his or her immediate neighbors, and second, because we do not follow each individual passenger as he or she moves through the airplane: The probability calculated with (27) is obtained in the same way for all grid points. Therefore, the tool requires only simple additions, subtractions, and shifts of two dimensional arrays. All operations which have to be performed to obtain (27) are local and do not depend on each other in the spatial domain. Yet, precisely because of this, no individual tracking of passengers is possible with this model, and the radius a passenger considers to make a decision is limited to the immediate neighbors.

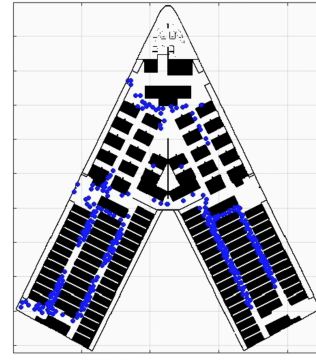
2.3.2.3 Further developments

Further developments of the evacuation tool which is discussed in this section exist. Major improvements to the tool were made by (Hellmann 2020). The main idea of this study is to increase the spatial resolution of the tool. A single passenger now occupies a lot more than one single cell, but many cells, all within a certain radius around a center cell. Around this area, there is an even larger radius which includes the cells used for the decision making process. Otherwise, the model runs the same way as the previously discussed preliminary tool. The model of Hellmann allows to consider far more detailed cabin geometries than the previous tool and the transition of the passengers through the airplane is a lot smoother than for the previous model. However, this comes at the cost of a much higher computational complexity. While for the model of Gebauer the complexity for operations in the spatial domain was simply $\mathcal{O}(N \times N)$, where $N \times N$ is the number of grid points in the two-dimensional array, the complexity rises to $\mathcal{O}(N \times N \times R \times R)$ for the model of Hellmann, where R is the radius around one passenger used for the decision making process. In the words of the previous two discussions: For the model of Hellmann, the degrees of freedom in the spatial domain depend on each other, at least within a certain radius around each cell.

The larger this radius gets, the more they depend on each other. If however R is one, we have the model of Gebauer with degrees of freedom in the spatial domain which are not fully independent, but they are only influenced by their neighbors. We call this a model with *local interactions* in the spatial domain.



Model of Gebauer



Model of Hellmann

Figure 15: Comparison of the model of Gebauer and the model of Hellmann. Results are shown for an exemplary time step during the evacuation simulation. The doors on the right of the aircraft are closed in both examples.

2.4 A 3D-array of degrees of freedom with local interactions

Let us now introduce the *Lattice Boltzmann Method*. It is this tool which we will use for the creation of the numerical flow channel in the next chapter.

Throughout this section, and in the next chapters of this work, we will draw much of our knowledge about the numerical tool from the book “The Lattice Boltzmann Method – Principles and Practice” (Krüger, Kusumaatmaja et al. 2017). In this book, the authors state: “[...] Researchers around the world are attracted to the Lattice Boltzmann Method for reasons such as its simplicity, its scalability on parallel computers, its extensibility, and the ease with which it can handle complex geometries. [...]” This echoes the impression the author of the present thesis has when regarding Lattice Boltzmann Method, especially in the light of the past three sections of this chapter.

The Lattice Boltzmann method is a tool which delivers second order accurate solutions of the compressible Navier-Stokes equation for low Mach numbers. The solution process for this is extremely simple: A three dimensional simulation is performed on a three dimensional lattice with equal spacing in each dimension. The numerical values on the lattice are stored in arrays. For example, the three dimensional field of the fluid density can be stored in a three dimensional array. In addition to a single value for the fluid density at each grid point, only 18 more values have to be stored at each discretization point for the simplest implementation of the Lattice Boltzmann method. Three of them represent the three components of the velocity field. The remaining 15 values represent particle distributions f_i moving in 15 different directions of the lattice. Simple operations have to be performed with the numerical values *directly at each grid point* (collision). Afterwards, the arrays have to be shifted by a single index to their neighboring lattice nodes (streaming) which yields the updated values for velocity, density and particle distributions.

We will describe the Lattice Boltzmann algorithm in rich detail in the following sections, particularly in [Section 3.5](#). For now, let us simply emphasize that the technique is a perfect example for a numerical procedure with *simple local interactions* in the spatial domain. Shifting an array, or performing local additions and subtractions – such operations do not depend on each other in the spatial domain. These operations can be fully vectorized and run in parallel. It is this which justifies our high interest in the Lattice Boltzmann method, especially in light of the past sections of this work.

This high interest is increased even more by current trends in the development of microprocessors. Examining [Figure 16a \(Rupp 2017\)](#), it becomes apparent that the number of transistors in a microprocessor increases exponentially over the past 50 years, a trend which is often described as *Moor's law*, see [\(Schaller 1997\)](#). From what we can observe in [Figure 16a](#), the number of transistors increases by a factor of 10 every 6 to 7 years. The very same trend can be observed for the computing power of microchips, often measured in floating point operation per second (FLOPS/sec). Trend data for this can be seen in [Figure 16b \(Rupp 2016\)](#). Additional graphs can be found in [\(Sun, Agostini et al. 2019\)](#). If we examine the rise of computing power not of single microchips, but of large computer clusters, we also observe the same trend. [Figure 16c](#) displays the development of floating point operations per second of the world's largest computers.

3 Setup of the numerical flow channel

In this section the setup of the numerical experiment will be explained. First, the high fidelity model for the flow will be introduced. We then ask which physical quantities determine the flow field around the wing. For this, we use again the procedure of dimensional analysis. Recall that this tool was also used in [Section 2.1](#). In the present section, we follow a procedure described in detail in [\(Anderson 2017\)](#). In the framework of the present study, the flow field around the wing is governed only by a single dimensionless parameter, and the geometry of the problem. We continue to describe this geometry and how it is modelled in rich detail. Subsequently, the simulation algorithm will be introduced. Finally, the output of the numerical experiment will be described.

3.1 High fidelity model

The high fidelity model of this work is given by the continuity equation

$$\frac{\partial \rho}{\partial t} + \operatorname{div} \rho \mathbf{v} = 0, \quad (28)$$

and the compressible Navier-Stokes equation

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \operatorname{grad} \mathbf{v} \right) = - \operatorname{grad} p + \eta \Delta \mathbf{v} + \left(\zeta + \frac{\eta}{3} \right) \operatorname{grad} \operatorname{div} \mathbf{v}, \quad (29)$$

see [\(Landau and Lifschitz 1971\)](#). η is the shear viscosity and ζ is the bulk viscosity. Often, η is referred to as dynamic viscosity which is related to the kinematic viscosity ν as

$$\eta = \nu \rho. \quad (30)$$

In its original form used in this work, the Lattice Boltzmann method solves [\(28\)](#) and [\(29\)](#) for weak compressibility, that is, errors will occur as the velocity of the fluid V approaches the speed of sound a . The problems under investigation in this work are well below this limit, they are at Mach numbers of $\operatorname{Ma} = V/a \approx 0.2$. Usually, such problems are treated entirely incompressible, see [\(Raymer 2012, Torenbeek 2013\)](#). It is often assumed that

compressibility effects will begin to manifest themselves at around $Ma \approx 0.4$, see for example (Schlichting and Truckenbrodt 1967, Schade and Kunz 2007).

Compressibility, however weak, requires the introduction of an equation of state to our high fidelity model. An equation of state which is frequently applied with the Lattice Boltzmann method and also used in the present work is the isothermal equation of state

$$p = \rho RT_0 . \quad (31)$$

The application of this relation with the Lattice Boltzmann method results in a bulk viscosity of $\zeta = 2\eta/3$ (Krüger, Kusumaatmaja et al. 2017).

3.2 The Buckingham pi theorem

If we expect the aerodynamic force R on a wing to depend on the freestream density ρ_∞ , the freestream velocity V_∞ , the size of the wing characterized by the chord length c , the freestream kinematic viscosity ν_∞ , the freestream speed of sound a_∞ , and the angle of attack α , that is

$$R = f(\rho_\infty, V_\infty, c, \nu_\infty, a_\infty, \alpha), \quad (32)$$

then dimensional analysis yields that R may be expressed in terms of a dimensionless force coefficient

$$C_R = \frac{R}{q_\infty S} \quad (33)$$

which only depends on the freestream *Reynolds number*

$$Re = \frac{V_\infty c}{\nu_\infty}, \quad (34)$$

the freestream *Mach number*

$$Ma = \frac{V_\infty}{a_\infty}, \quad (35)$$

and the angle of attack:

$$C_R = f(\text{Re}, \text{Ma}, \alpha). \quad (36)$$

In (33), S is a reference area characterizing the size of the body (such as the wing area), and $q_\infty = \rho_\infty V_\infty^2 / 2$ is the dynamic pressure.

The above also holds true for components of R , such as lift L and drag D . Both may be expressed in terms of dimensionless coefficients

$$C_L = \frac{L}{q_\infty S}, \quad C_D = \frac{D}{q_\infty S}, \quad (37)$$

which only depend on Re , Ma , and α :

$$\begin{aligned} C_L &= f(\text{Re}, \text{Ma}, \alpha), \\ C_D &= f(\text{Re}, \text{Ma}, \alpha). \end{aligned} \quad (38)$$

In fact, the above holds true not only for force coefficients but for all dimensionless field values ($\frac{V}{V_\infty}, \frac{p}{p_\infty}, \dots$) of the flow.

For full derivation of the statements above and further extensive and references on this topic see (Anderson 2017).

For the present work, the above means that the entire flow which will be investigated depends only on the geometry of the problem, the Reynolds number, and the Mach number. Moreover, as was mentioned in Section 3.1, compressibility effects will be extremely low for problems under investigation in this work with Mach numbers as low as $\text{Ma} \approx 0.2$. Therefore, we can conclude, that in the present work the flow under investigation for a given geometry only depends on a single dimensionless parameter, which is the Reynolds number.

3.3 Wind tunnels

We have now discussed the differential equations which govern the flow, and applied dimensional analysis to determine which parameters govern their solutions. We have also decided for a numerical tool to solve these equations for the parameters which are of interest. It is now almost time to develop the numerical experiment. Let us however pause one last time and examine a real wind tunnel.

The first flow channel for scientific investigations was built by Ludwig Prandtl ([Prandtl 1905](#)). This design inspired closed circuit wind tunnels, called Göttinger type wind tunnels. An exemplary wind tunnel of this type operational today is the Transonic Wind Tunnel Göttingen (see the colored image below).

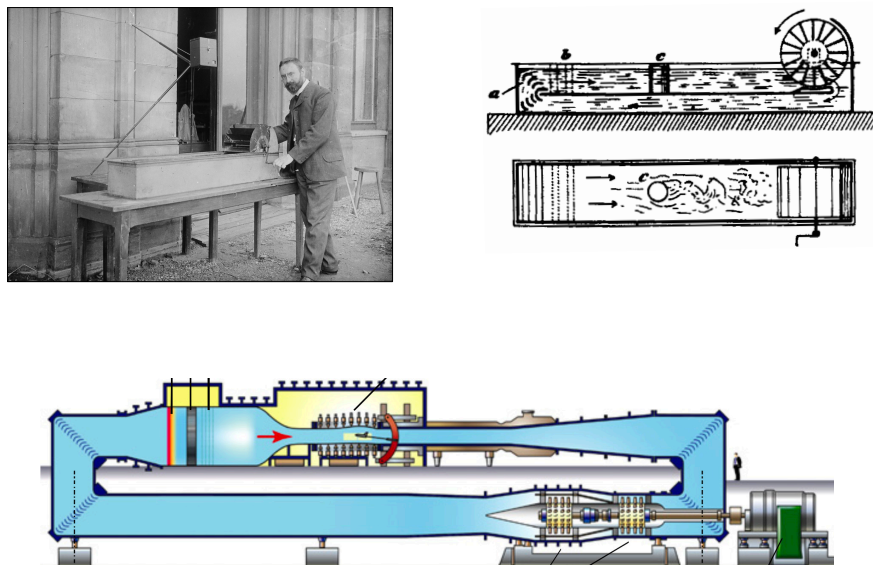


Figure 17: The left picture shows Ludwig Prandtl with his water tunnel in Hannover, Germany, in 1904 ([DLR 2021](#)). The original drawing of the apparatus is shown on the right ([Prandtl 1905](#)). This design inspired closed circuit wind tunnels, called Göttinger type wind tunnels. An exemplary wind tunnel of this type operational today is the Transonic Wind Tunnel Göttingen (TWG) (see the large image in the middle). The length of this tunnel is 46.5 m, the drive motor has a power of 12 MW, and the maximum Reynolds number which can be achieved in the test section ($1\text{ m} \times 1\text{ m} \times 4.5\text{ m}$) is $\text{Re} = 1.8 \times 10^6$ for a reference length of $l_{\text{ref}} = 0.1\text{ m}$.

3.4 Experimental setup

Now it is time to build the numerical wind tunnel of this work. An image of the wind tunnel is displayed together with some annotations in [Figure 18](#). The computations are performed on a workstation with an *AMD Ryzen Threadripper* processor with 24 cores capable of a maximum speed of 4.5 GHz. The available memory (RAM) for the computations is 256 GB. The full specifications of the workstation are displayed in [Table 1](#).

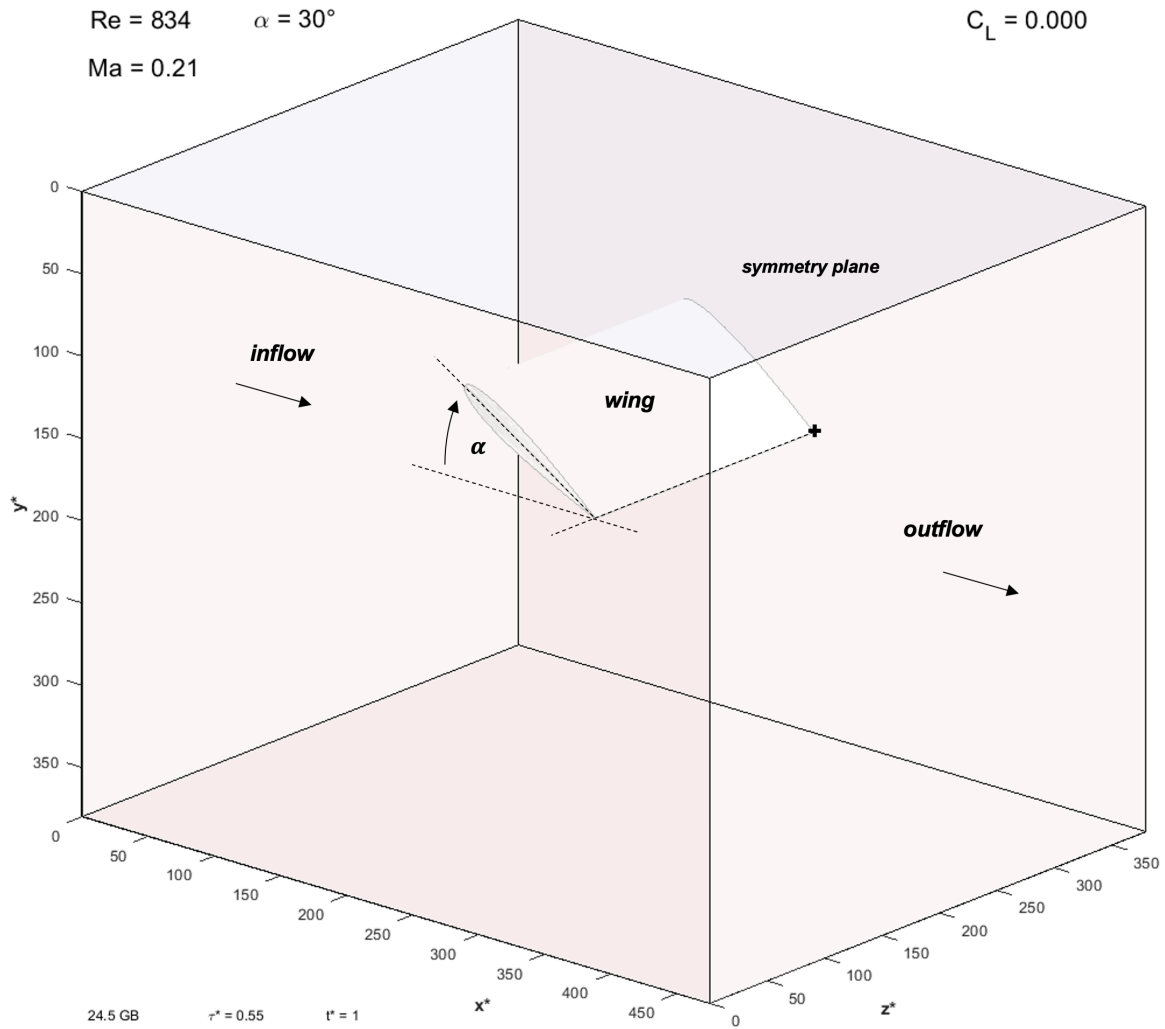


Figure 18: Setup of the numerical windtunnel. Annotations are displayed in bold italic print.

Processor	AMD Ryzen Threadripper 3960X 24-Core 3.80GHz (max: 4.5GHz)
RAM	256 GB DDR4 3200MHz
Mainboard	ASRock TRX40 Creator Mainboard
System	Windows 10 Pro
Matlab version	R2021a

Table 1: Workstation specifications

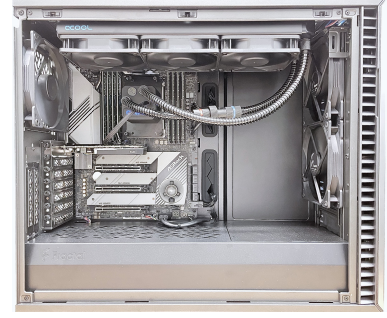


Figure 19: Image of the workstation

3.4.1 Dimensionless framework

The setup which is displayed in [Figure 18](#) makes use of a fully dimensionless framework. As we use this framework throughout this work, note that it is completely in line with the fundamental relations introduced in [Section 3.2](#).

Dimensionless variables are denoted with a star (*) throughout this work. The coordinate axes which are shown in [Figure 18](#) are dimensionless spatial coordinates. The dimensional coordinates (x, y, z) relate to their dimensionless counterparts (x^*, y^*, z^*) as

$$x = x^* \zeta_l, \quad y = y^* \zeta_l, \quad z = z^* \zeta_l, \quad (39)$$

where we choose

$$\zeta_l = \Delta x = \Delta y = \Delta z. \quad (40)$$

This means that

$$\Delta x^* = \Delta y^* = \Delta z^* = 1. \quad (41)$$

We use the same technique for the time t . It relates to its dimensionless counterpart t^* as

$$t = t^* \zeta_t, \quad (42)$$

where we choose

$$\zeta_t = \Delta t, \quad (43)$$

which means that

$$\Delta t^* = 1 . \quad (44)$$

Units as they are given with (41) and (44) are called *lattice units* (Krüger, Kusumaatmaja et al. 2017).

The dimensional fluid density ρ relates to its dimensionless counterpart ρ^* as

$$\rho = \rho^* \zeta_\rho . \quad (45)$$

We choose

$$\zeta_\rho = \rho_\infty , \quad (46)$$

which means that

$$\rho_\infty^* = 1 . \quad (47)$$

In the present dimensionless framework, all other variables can now be related to their dimensionless counterparts through combinations of ζ_l , ζ_t and ζ_ρ . For example, for the velocity field, we have

$$\mathbf{v} = \mathbf{v}^* \zeta_l / \zeta_t . \quad (48)$$

For the pressure field, we have

$$p = p^* \zeta_\rho \zeta_l^2 / \zeta_t^2 , \quad (49)$$

for the kinematic viscosity, it is

$$\nu = \nu^* \zeta_l^2 / \zeta_t , \quad (50)$$

and for the speed of sound

$$a = a^* \zeta_l / \zeta_t . \quad (51)$$

3.4.2 Geometry

Let us position a half model of the wing we seek to investigate in our wind tunnel as shown in Figure 18. Half models are commonly used to investigate flows around symmetric wings, see for example (Van Empelen and Vos 2021).

The intersection of the trailing edge of the wing and its symmetry plane is shown in [Figure 18](#) with a black cross (+). The x^* and y^* coordinates of this point position the wing inside the tunnel. The axis which runs through this point in z^* direction is highlighted with a dashed line in [Figure 18](#). The wing may be turned around this axis which sets the angle of attack α .

We will go into detail about the boundary conditions of the numerical simulation in [Section 3.5](#). Here, let us only state briefly that at the inflow plane we set the inflow velocity V_∞^* and the inflow density ρ_∞^* . At the outflow plane, we assume that the field values of velocity, density and pressure change very little. The three boundary planes which are left will be modelled as solid but frictionless walls. The boundary condition we apply here is often referred to as free-slip boundary condition. We will also discuss this in [Section 3.5](#).

The entire numerical setup of the geometry is created with the software environment of *Matlab R2021a*. All necessary calculations and visualizations of the results are performed with this tool as well.

Let us now discuss the wing geometry we place into our wind tunnel. A major operation before any calculations can be performed is to obtain the discretization points which lie within the geometry we seek to investigate. Airfoil sections used in this work are exclusively from the four digit NACA airfoil series. Wing elements are created as ruled surfaces between the airfoil sections. The creation of such a surface can be achieved using a simple `alphaShape` object within Matlab. In further developments of this work, it may be necessary to investigate methods to import geometry files from CAD programs. In the present work however, simple geometries were created directly within Matlab. Grid points which lie within an `alphaShape` can be obtained with the `inShape` function in Matlab. This is by no means a trivial operation and can be very time consuming. In order to mitigate any time penalties encountered through the operation, we have limited the search region for points which lie within the wing geometry to a small box which encloses the wing, see [Figure 20](#).

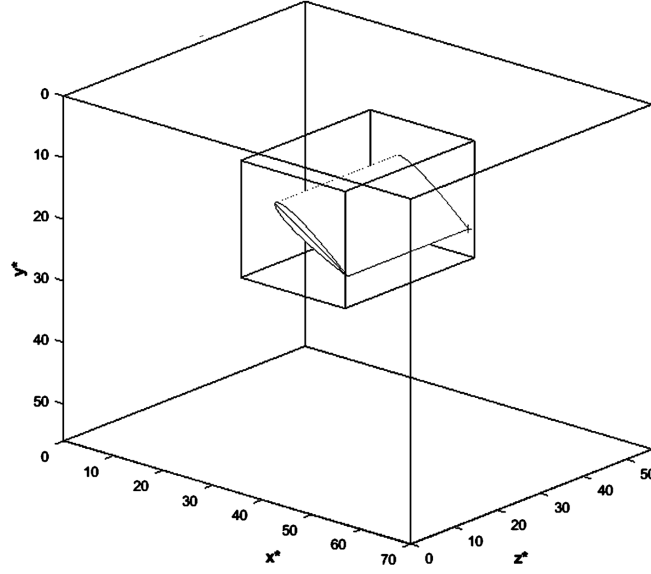


Figure 20: Discretization points which lie inside the wing are searched inside a small box which fully encloses the wing.

Times for using the `inShape` function on this setup were found to be on the order of a single iteration step of the main algorithm for the grid sizes investigated in this work. RAM requirements were found to be on the order of 50% of the RAM required in the main algorithm for the grid sizes investigated in this work. With these benchmarks, and the notion that this function has to be run only a single time before the main algorithm is entered, it was deemed a viable technique for the scope of this work.

Discretization points which lie inside an exemplary wing geometry are displayed in [Figure 21](#) for two different discretization resolutions. The images are enlarged versions of [Figure 20](#) to show only the wing and its enclosing box, not the entire wind tunnel. The half model which is displayed in the images on the left side has a half span $b^*/2 = 28$ discretization points. This is half of the width of the entire simulation region, which was chosen with $B^* = 56$ discretization points. The images on the right side show the same geometry, but a resolution which is ten times higher. Here we have a half model with a half span of $b^*/2 = 280$ discretization points lying inside an overall simulation area with a width of $B^* = 560$ discretization points.

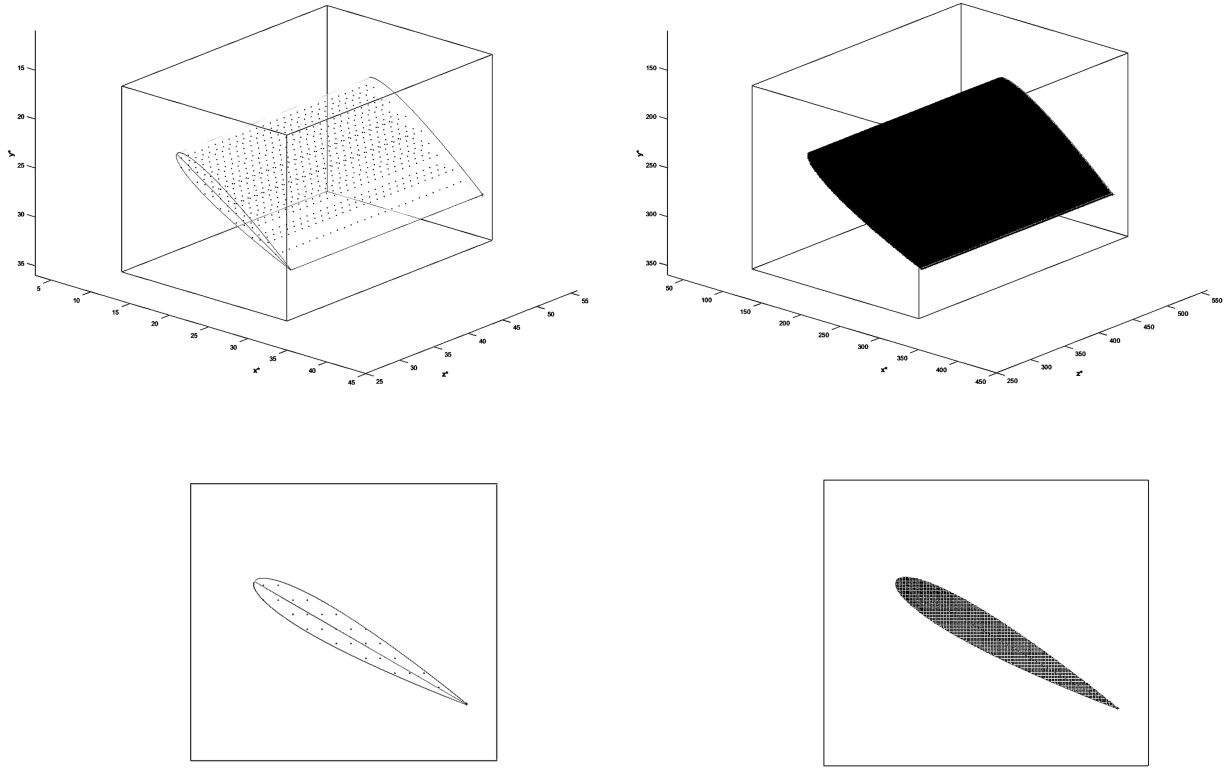


Figure 21: Two different resolutions of the same wing geometry are shown. The images on the right side show a resolution which is ten times higher than the resolution shown on the left. Note that the images are enlarged versions of Figure 20 to show only the wing and its enclosing box, not the entire wind tunnel. The half model which is displayed in the images on the left side has a half span $b^*/2 = 28$ discretization points. This is half of the width of the entire simulation region, which was chosen with $B^* = 56$ discretization points. The images on the right side show a half model with a half span of $b^*/2 = 280$ discretization points lying inside an overall simulation area with a width of $B^* = 560$ discretization points.

Before we end our description of the experimental setup with comments on the general system size and the available RAM, let us go back one more time to the introductory image of the experimental setup, [Figure 18](#). We have now discussed the axis and dimensions in this image, and we have discussed the geometry which is shown. We have yet to comment on the values which are shown in the header and the footer of this image, and will appear again in most images of the results of this work. On the top left, we see the two dimensionless governing parameters of the simulation, Re and Ma . We will only change Re in our simulations. Slight changes of the low Mach number Ma will have no influence on the results within the scope of this work as was explained in [Section 3.2](#). Then, in the same upper left corner, we display the angle of attack α . It influences the geometry of the problem and thus, it certainly influences the results. Many results will be displayed as a dependence

on α . In the top right corner, one of the results of the simulation, the lift coefficient C_L , is displayed (see more in [Section 3.6.2](#)). In the footer we show first the amount of calculated RAM for the simulation (see next section), second, the relaxation parameter τ^* , and third, current timestep of the simulation t^* .

3.4.3 Available RAM and maximum Reynolds number

The available amount of RAM dictates the maximum Reynolds number which can be achieved with the present setup. As will be discussed later in this work, the maximum size investigated in this work was a wind tunnel with a length in x^* direction of $S^* = 700$ discretization points, a height in y^* direction of $H^* = 560$ discretization points, and a width in z^* direction of $B^* = 560$ discretization points. This results in a maximum size of

$$N_{\max} = S^* \times H^* \times B^* = 700 \times 560 \times 560 = 220 \times 10^6 \quad (52)$$

discretization points in the wind tunnel. For a D3G15 lattice, we have to store 15 populations f in each point (see next section). Furthermore, the computation of an equilibrium distribution f_{eq} which is required for the simulation runs fully vectorized in each time step when an additional dimension with three entries is added for the computation of the term $\mathbf{v}^*(\mathbf{x}^*, t) \cdot \mathbf{c}_i$. (This also, we will discuss in rich detail later in the next section of this work.) This gives a total amount of

$$N_{\max, \text{store}} = N_{\max} \times 15 \times 3 = 220 \times 10^6 \times 45 = 1 \times 10^{10} \quad (53)$$

elements which have to be stored in one time step. From this, we can obtain the calculated amount of maximum RAM with

$$\text{RAM}_{\max, \text{calculated}} = N_{\max, \text{store}} \times 8 \text{ B} = 80 \times 10^9 \text{ B} = 80 \text{ GB} \quad (54)$$

In numerous simulations of systems with various size, it was found that the actual amount of RAM which is needed by Matlab R2021a for a smooth execution of the developed code scales with three times the value obtained with the calculated value in equation (54), that is

$$\text{RAM}_{\max} = \text{RAM}_{\max, \text{calculated}} \times 3 = 240 \text{ GB}. \quad (55)$$

This value was found to be a sufficient margin below the available amount of 256 GB on the workstation to ensure an efficient and robust execution of the simulation.

We are now able to estimate the maximum Reynolds number which can be achieved with our experimental setup. Let us assume that the largest wing we will be able to place into the wind tunnel has a reference chord of 35% of the tunnel length S^* :

$$l_{\max}^* = S^* \times 0.35 = 252. \quad (56)$$

Let us also assume that the lowest relaxation parameter τ^* for which we are able to conduct stable and accurate simulations will be (see [Section 3.5.3](#) for more details)

$$\tau_{\min}^* = 0.52. \quad (57)$$

If we then choose a high Mach number, which is, however, still below the limit at which compressibility effects begin to matter (see [Section 3.1](#)),

$$\text{Ma}_{\max} = 0.21, \quad (58)$$

we obtain our dimensionless inflow velocity with (compare [\(35\)](#) and [\(69\)](#))

$$V_{\infty, \max}^* = \text{Ma}_{\max} a^* = 0.21 \times \sqrt{1/3} = 0.12. \quad (59)$$

With

$$\nu^* = a^{*2} (\tau^* - 1/2). \quad (60)$$

for the viscosity (see [Section 3.5.3](#)), we then obtain a maximum Reynolds number of

$$\text{Re}_{\max} = \frac{V_{\infty, \max}^* l_{\max}^*}{\nu_{\min}^*} = \frac{V_{\infty, \max}^* l_{\max}^*}{a^{*2} (\tau_{\min}^* - 1/2)} = \frac{0.12 \times 252}{1/3 \times (0.52 - 1/2)} = 4.6 \times 10^3. \quad (61)$$

3.5 Algorithm

In the last section, we have described the experimental setup of our wind tunnel. We introduced the dimensionless framework and we described the geometry of the problem. We have also gained a first understanding of how the Reynolds number scales with the available amount of memory. Let us now introduce the simulation procedure which is used to obtain the field values of the flow inside the wind tunnel.

3.5.1 Initialization

At $t^* = 0$, we set the velocity field to zero, that is,

$$\mathbf{v}^*(\mathbf{x}^*, t^* = 0) = 0. \quad (62)$$

In practice, this means the creation of a four-dimensional array which consists entirely out of zeros. The first three dimensions of the array are the spatial dimensions of the wind tunnel. The fourth dimension is used to store the velocity components in x^* , y^* and z^* direction. In total, this array has $S^* \times H^* \times B^* \times 3$ entries. We further initialize the fluid density with

$$\rho^*(\mathbf{x}^*, t^* = 0) = \rho_\infty^* = 1, \quad (63)$$

compare (47). In practice, we create a three-dimensional array which consists entirely of ones. In total, the array has $S^* \times H^* \times B^*$ entries.

Note that there are various ways to initialize the velocity and density field. For example, one could use a constant velocity in x^* -direction, that is, $\mathbf{v}^*(\mathbf{x}^*, t^* = 0) = V_\infty^* \mathbf{e}_x$ and again, $\rho^*(\mathbf{x}^*, t^* = 0) = 1$. Like (62) and (63), this solves our high fidelity model (28) and (29). However, it violates the no-slip boundary condition at the wing we place into our wind tunnel. With (62) and (63), we fulfill both our high fidelity model (28) and (29) and all the boundary conditions which we place upon our model. The author of the present thesis has observed that this zero-velocity initialization scheme allows for more stable simulations when compared to schemes with a constant non-zero initial velocity component in the entire flow field. The phrase “more stable simulations” translates to a lower possible relaxation parameter τ^* and thus, higher Reynolds numbers Re , as we will see in the following sections.

In the Lattice Boltzmann method, $f_i(\mathbf{x}^*, t^*)$ represents the density of particles moving in the various directions of the lattice. We require an initialization of this so-called particle distribution, and also of the equilibrium particle distribution $f_{eq,i}(\mathbf{x}^*, t^*)$. The latter describes the particle distribution in a fluid without internal friction. We will come back to this in more detail later. At this stage, let us only note that in the main algorithm, through particle motions and collisions, the particle distributions undergo a continuous relaxation towards their equilibrium state where there is no friction. Indeed, a motion of a fluid with particle distributions which are always entirely equal to the equilibrium particle distribution,

$f_i(\mathbf{x}^*, t^*) = f_{\text{eq},i}(\mathbf{x}^*, t^*)$, fulfills the *Euler momentum equation*, which is (29) without any viscous terms.

In the Lattice Boltzmann Method, the equilibrium particle distribution is obtained with

$$f_{\text{eq},i}(\mathbf{x}^*, t^*) = w_i \rho^* (1 + 2\vartheta_1 \Pi_c + \vartheta_2 \Pi_c^2 - \vartheta_1 \Pi), \quad (64)$$

where it is

$$\vartheta_1 = \frac{1}{2a^{*2}}, \quad \vartheta_2 = \frac{1}{2a^{*4}}, \quad (65)$$

and

$$\Pi_c = \mathbf{v}^* \cdot \mathbf{c}_i, \quad \Pi = \mathbf{v}^* \cdot \mathbf{v}^*. \quad (66)$$

Therein, we have

$$w_i = [8 \ 8 \ 8 \ 8 \ 8 \ 8 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 16]/72 \quad (67)$$

and

$$\begin{aligned} c_{x^*,i} &= [1 \ -1 \ 0 \ 0 \ 0 \ 0 \ 1 \ -1 \ 1 \ -1 \ 1 \ -1 \ -1 \ 1 \ 0], \\ c_{y^*,i} &= [0 \ 0 \ 1 \ -1 \ 0 \ 0 \ 1 \ -1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ 0], \\ c_{z^*,i} &= [0 \ 0 \ 0 \ 0 \ 1 \ -1 \ 1 \ -1 \ -1 \ 1 \ 1 \ -1 \ 1 \ -1 \ 0]. \end{aligned} \quad (68)$$

This is the D3Q15 velocity set. It is called like this because we have a three-dimensional lattice and 15 velocity vectors \mathbf{c}_i with corresponding values w_i . Other common three-dimensional sets are the D3Q19 or D3Q27 velocity sets. In the present work, we will use the D3Q15 set given above, because it is the set with the smallest number of lattice vectors and thus, uses the least amount of memory. We will review this decision at various stages throughout this work.

For the present isothermal model (see Section 3.1), it is $a = \sqrt{1/3} \Delta x / \Delta t$. With (51), we then have

$$a^* = \sqrt{1/3}. \quad (69)$$

In order to obtain the equilibrium distribution at $t^* = 0$, we can insert (62) and (63) into (64). This yields

$$f_{\text{eq},i}(\mathbf{x}^*, t^* = 0) = w_i. \quad (70)$$

In practice, we create a five-dimensional array. As before, the first three dimensions are the spatial coordinates. The fourth dimension is left open for the time being. We will need it later for an efficient computation. The fifth dimension is used to store 15 values at each grid point. For example, the first value at each grid point is $w_1 = \frac{8}{72}$. In the initialized array, we have to store this same value $S^* \times H^* \times B^*$ -times, once at each grid point. Thus, the entire array for the equilibrium distribution with 15 values at each grid point has $S^* \times H^* \times B^* \times 1 \times 15$ entries in total.

A resting fluid with zero-velocity fulfills both (29) and the Euler momentum equation, or in other words, the particle distribution is fully in equilibrium. Therefore, at $t^* = 0$, we have

$$f_i(\mathbf{x}^*, t^* = 0) = f_{\text{eq},i}(\mathbf{x}^*, t^* = 0) = w_i. \quad (71)$$

The array we initialize this way to store the particle distributions has also $S^* \times H^* \times B^* \times 1 \times 15$ entries.

3.5.2 Main algorithm

We enter the main algorithm with the calculation of the dimensionless density and velocity field. As we just initialized both of these fields, we could indeed enter the main algorithm at a later stage and thus save a few steps in the very beginning. However, in order to be consistent with current literature, we begin with the density and velocity field.

3.5.2.1 Dimensionless density and velocity field

The particle distribution $f_i(\mathbf{x}^*, t^*)$ represents the density of particles moving in the various directions given with the vectors \mathbf{c}_i . Therefore, the overall density of the fluid is

$$\rho^*(\mathbf{x}^*, t^*) = \sum_i f_i(\mathbf{x}^*, t^*). \quad (72)$$

Note that in the present framework laid out in Section 3.4.1, all of these quantities are dimensionless. In order to avoid confusion with notations from other authors however, we do not write out the star (*) together with f . It is common to simply write f , even if a framework is used where these values are dimensionless.

In practice, (72) is a summation of all elements in the fifth dimension of the array we use to store $f_i(\mathbf{x}^*, t^*)$. The resulting array has $S^* \times H^* \times B^*$ entries.

In addition to the dimensionless density field, we also obtain the dimensionless velocity field at the beginning of the main algorithm. Here, we simply have

$$\mathbf{v}^*(\mathbf{x}^*, t^*) = \frac{\sum_i \mathbf{c}_i f_i(\mathbf{x}^*, t^*)}{\rho^*(\mathbf{x}^*, t^*)}. \quad (73)$$

using the vectors \mathbf{c}_i of the given velocity set. In practice, we predefine the vectors \mathbf{c}_i before we enter the main algorithm. We do so leaving the first three dimensions open. We use the fourth dimension to store the three velocity components in x^* , y^* and z^* direction. We use the fifth dimension to store all 15 vectors. In total, the predefined array to store \mathbf{c}_i has $1 \times 1 \times 1 \times 3 \times 15$ entries. For the computation of (73), we perform an element wise multiplication of this array with the array we use to store $f_i(\mathbf{x}^*, t^*)$. This yields a temporary array with $S^* \times H^* \times B^* \times 3 \times 15$ entries. A summation in the fifth dimension of this array is performed, and then an element wise division with the array we use to store $\rho^*(\mathbf{x}^*, t^*)$. We then have our array for the fluid velocity field with $S^* \times H^* \times B^* \times 3$ entries.

Let us note that the use of the large temporary array with $S^* \times H^* \times B^* \times 3 \times 15$ entries is not required. One could also perform the operations for the three velocity components after each other which would yield a maximum array size of $S^* \times H^* \times B^* \times 1 \times 15$ in the computation of (73). Depending on the framework used for the implementation, this may be a slower operation, because the three operation are independent from another and one should allow for parallelization of these operations. In the present work, we have therefore decided to use the implementation with the larger maximum array size of $S^* \times H^* \times B^* \times 3 \times 15$ elements in order to save computational time. However, if one has a restriction on RAM more then on computational time, one should consider the implementation with a smaller maximum array size. Further below, there will be another calculation step and the main algorithm when we will have to make this decision. We will point it out when it arises.

3.5.2.2 Output

Now that we have our dimensionless density and velocity field, we can proceed with the next step in the main algorithm. The next step is the output of these dimensionless field values (green box in Figure 22). They can be used in further calculations, plotted in some way, written to a hard drive, etc. We will describe this output in more detail in Section 3.6.

3.5.2.3 Equilibrium particle distribution

After the output, the main algorithm continues with the computation of the equilibrium distribution (64). Both parameters ϑ_1 and ϑ_2 can be predefined before the main algorithm is entered. Π_c and Π are given with (66). Recall that the array we use to store the fluid velocity field $\mathbf{v}^*(\mathbf{x}^*, t^*)$ has $S^* \times H^* \times B^* \times 3$ entries. For the computation of $\Pi = \mathbf{v}^* \cdot \mathbf{v}^*$, we perform an element wise multiplication of this array with itself, and then take the sum of all elements in the fourth dimension. The resulting array for Π has $S^* \times H^* \times B^*$ entries. Let us now recall that the array we use to store \mathbf{c}_i has $1 \times 1 \times 1 \times 3 \times 15$ entries. For the computation of $\Pi_c = \mathbf{v}^* \cdot \mathbf{c}_i$ we perform an element wise multiplication of this array with the array for $\mathbf{v}^*(\mathbf{x}^*, t^*)$. This yields a temporary array with $S^* \times H^* \times B^* \times 3 \times 15$ entries. After a summation in the fourth dimension of this array, we have our final array for Π_c with $S^* \times H^* \times B^* \times 1 \times 15$ entries. In this form, both arrays for Π_c and Π can then be used with element wise additions, subtractions and multiplications to obtain (64). In that calculation, w_i takes the shape of a predefined array with $1 \times 1 \times 1 \times 1 \times 15$ entries. The resulting array for the equilibrium distribution $f_{\text{eq},i}(\mathbf{x}^*, t^*)$ has $S^* \times H^* \times B^* \times 1 \times 15$ entries.

Here again, let us note that the use of the large temporary array with $S^* \times H^* \times B^* \times 3 \times 15$ entries is not required. In order to obtain $\Pi_c = \mathbf{v}^* \cdot \mathbf{c}_i$ with less RAM, one could perform operations after one another, thereby reducing the required memory but limiting possibilities for parallelization. In the present work we have decided to use the larger array in order to save computational time.

3.5.2.4 Collision

The next step in the Lattice Boltzmann algorithm is collision. For the particle distribution after the collision, $f_{\text{col},i}(\mathbf{x}^*, t^*)$, we have

$$f_{\text{col},i}(\mathbf{x}^*, t^*) = \omega_p f_i(\mathbf{x}^*, t^*) + \omega f_{\text{eq},i}(\mathbf{x}^*, t^*), \quad (74)$$

where

$$\omega = \frac{1}{\tau^*} \quad (75)$$

and

$$\omega_p = 1 - \omega . \quad (76)$$

τ^* is the dimensionless relaxation time. We discuss this simulation parameter in [Section 3.5.3](#). ω and ω_p are predefined before the main algorithm is entered. In relation (74), we use the BGK collision operator ([Bhatnagar, Gross et al. 1954](#)). This is the simplest collision operator in the Lattice Boltzmann method. Other more refined operators are the TRT or MRT model, see ([Krüger, Kusumaatmaja et al. 2017](#)). In the present work, we apply the BGK operator in this work because of its simplicity. We discuss this decision in more detail at various stages throughout the following work.

In practice, (74) is an addition of two arrays after they were each multiplied by a scalar. The arrays in (74) have $S^* \times H^* \times B^* \times 1 \times 15$ entries.

3.5.2.5 Streaming

After collision, the resulting distributions of particles are shifted to their neighboring grid points according to the direction given with \mathbf{c}_i , that is

$$f_i(\mathbf{x}^* + \mathbf{c}_i, t^* + 1) = f_{\text{col},i}(\mathbf{x}^*, t^*). \quad (77)$$

In practice, this is performed by a change of indices of the array, which is a trivial operation orders of magnitude faster than any summation or multiplication with the array elements.

3.5.2.6 Application of boundary conditions

Before we can close the loop so that the algorithm can run over and over again, we have to include some boundary conditions. The step is performed at the end of the Lattice Boltzmann algorithm.

It is worth noting, that with the initialization of a resting fluid given above, we could have entered the main algorithm at the present step. Nothing will have changed for the initialized distributions and dimensionless fields over the previous steps during the first iteration. Once we impose boundary conditions, this will change.

3.5.2.6.1 Inflow

At the opening of the wind tunnel, we impose a constant velocity in x^* direction, that is

$$\mathbf{v}^*(x^* = 1, y^*, z^*, t^*) = V_\infty^* \mathbf{e}_x. \quad (78)$$

In the present algorithm, this can be achieved by setting

$$f_i(x^* = 1, y^*, z^*, t^* + 1) = f_{\text{col},i}(x^* = 1, y^*, z^*, t^*) - 2w_i \rho_\infty^* \frac{\mathbf{c}_i \cdot V_\infty^* \mathbf{e}_x}{a^{*2}}, \quad (79)$$

which translates to

$$\begin{aligned} f_1(x^* = 1, y^*, z^*, t^* + 1) &= f_{\text{col},2}(x^* = 1, y^*, z^*, t^*) + 4w_2 \rho_\infty^* V_\infty^* \vartheta_1, \\ f_{j_k}(x^* = 1, y^*, z^*, t^* + 1) &= f_{\text{col},g_k}(x^* = 1, y^*, z^*, t^*) + 4w_8 \rho_\infty^* V_\infty^* \vartheta_1, \end{aligned} \quad (80)$$

where $j_k = [7 \ 9 \ 11 \ 14]$ and $g_k = [8 \ 10 \ 12 \ 13]$ (Ladd 1994, Ladd and Verberg 2001).

3.5.2.6.2 Wing

In the Lattice Boltzmann method, a resting solid surface, such as the surface of our wing, can be modelled by the application of the bounce back technique (Frisch, Hasslacher et al. 1986, Cornubert, d'Humières et al. 1991, Ziegler 1993, Ginzbourg and Adler 1994, Ladd 1994). In Section 3.4.2, we have described our technique for the selection of the discretization points of the lattice which lie within the wing geometry (see Figure 21). Let us denote the positions of grid points of the outside boundary to this selection with \mathbf{x}_b^* . Here, “outside boundary” means all those grid points outside the geometry from which an internal grid point can be reached by a single step along one of the vectors \mathbf{c}_i . In the bounce back technique, distributions which leave the outside boundary \mathbf{x}_b^* at a time t^* during the streaming step for the inside of the geometry are reflected back in the direction $\mathbf{c}_{\bar{i}} = -\mathbf{c}_i$ and arrive at time $t^* + 1$ at the node \mathbf{x}_b^* from which they originally came. We have

$$f_{\bar{i}}(\mathbf{x}_b^*, t^* + 1) = f_{\text{col},i}(\mathbf{x}_b^*, t^*). \quad (81)$$

In practice, we perform the standard streaming step given in Section 3.5.2.5 for all grid points. Afterwards, we replace the values we obtain with (81) for the grid points at the outside boundary \mathbf{x}_b^* . Note that with this technique, we store the array with $S^* \times H^* \times B^* \times 1 \times 15$ entries for $f_{\text{col},i}(\mathbf{x}^*, t^*)$ and the array with $S^* \times H^* \times B^* \times 1 \times 15$ entries for $f_i(\mathbf{x}^*, t^* + 1)$ at the same time during one iteration. In theory, this is not necessary, because we only need values of $f_{\text{col},i}$ directly at \mathbf{x}_b^* to perform the collision step. We will encounter this in a similar manner at the tunnel walls and we have already seen it in the previous section where we needed some old values for $f_{\text{col},i}$ at the tunnel inflow region. For simplicity and to reduce the risk of errors, in the present work,

we did indeed keep the entire array with $S^* \times H^* \times B^* \times 1 \times 15$ entries for $f_{\text{col},i}(\mathbf{x}^*, t^*)$ in addition to the array for $f_i(\mathbf{x}^*, t^* + 1)$ with the same amount of elements. (Recall that we already decided for the storage of a large temporary array with $S^* \times H^* \times B^* \times 3 \times 15$ elements at an earlier stage in this work.) However, if one has a strong limit on RAM, one can avoid this large array. In that case, one should also consider only keeping copies of those fragments of $f_{\text{col},i}$ which are necessary to apply boundary conditions in order keep the memory usage as low as possible.

3.5.2.6.3 Tunnel walls

At the tunnel walls, we apply a free-slip boundary condition. In the Lattice Boltzmann method, this technique is similar to the bounce back technique. Now, only the normal velocity component is reflected, that is $c_{j,n} = -c_{i,n}$. The new distributions at \mathbf{x}_b^* are given by

$$f_j(\mathbf{x}_b^* + \mathbf{c}_{j,t}, t^* + 1) = f_{\text{col},i}(\mathbf{x}_b^*, t^*), \quad (82)$$

where $\mathbf{c}_{j,t} = \mathbf{c}_{i,t}$ is the tangential velocity of the distributions. A symmetry boundary condition, as we apply on one of the walls of our wind tunnel (see [Figure 18](#)), can also be achieved using this technique ([Succi 2001](#), [Da Silva 2008](#), [Falcucci, Aureli et al. 2011](#)).

Let us note, that it is well-known that walls in close proximity of the test object influence the aerodynamic forces on this object. The smaller the object compared to the tunnel cross section, the smaller are these interference effects. Simple rules exist to transform results such as lift or drag coefficient to the freestream case without any surrounding walls. An extensive analysis of this matter can be found here ([Glauert 1933](#)).

3.5.2.6.4 Outflow

At the outflow plane of our numerical windtunnel, we perform a simple operation. After streaming with relation (77), $f_i(\mathbf{x}^*, t^* + 1)$ remains undetermined at the outflow plane for some i because there are no distributions to stream into the tunnel from the outside. We simply fill these distributions by copying all missing elements at $f_i(\max(x^*), y^*, z^*, t^* + 1)$ from their neighbors at $f_i(\max(x^*) - 1, y^*, z^*, t^* + 1)$. We will discuss this technique in the next section.

3.5.3 Comments

We have now introduced the entire simulation algorithm. Let us take a moment to examine it once more. A schematical representation of the algorithm can be seen below.

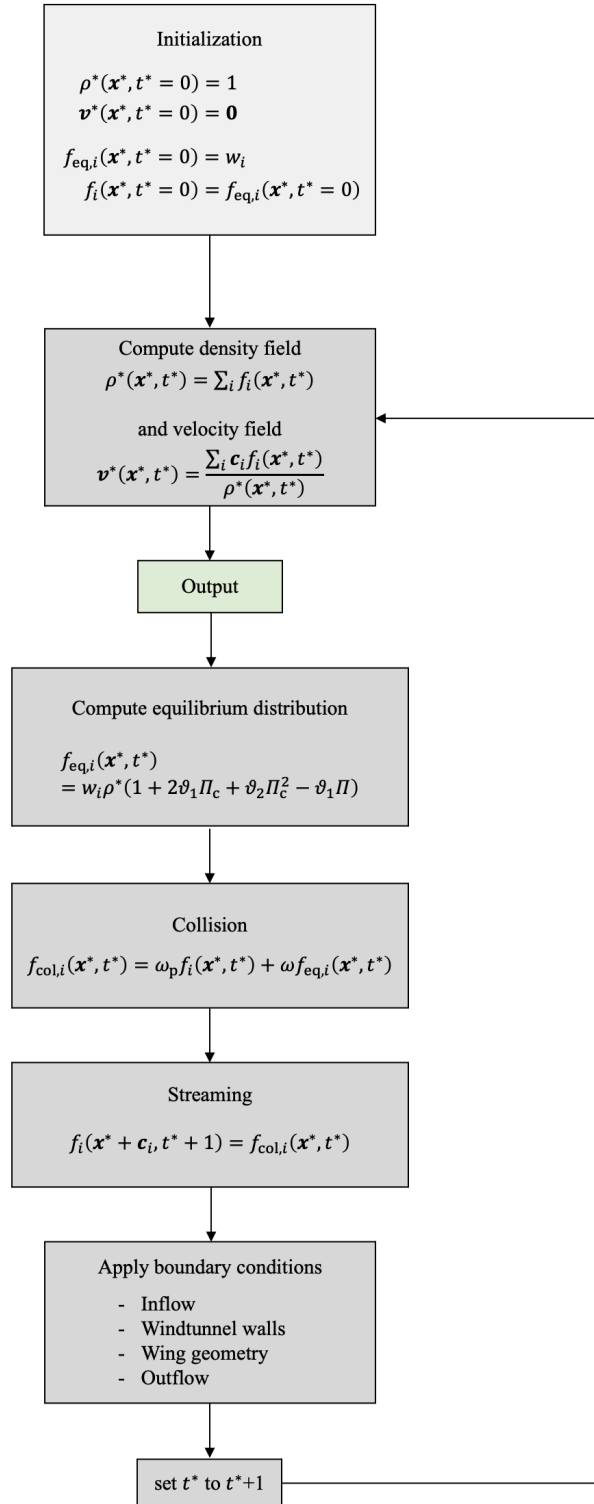


Figure 22: Simulation algorithm

With the experimental setup described in the previous section, and the simulation algorithm from this section, our numerical wind tunnel is now almost ready to be used and to be applied to various problems which are of interest. However, before we proceed to this, let us make a few more statements.

First of all, please note again that all the calculation steps which have to be performed during an iteration in the simulation algorithm are simple local interactions in the spatial domain. Compare again with Section 2 and note, that such a framework allows for simple parallelization which is well-suited to current trends in the development of microprocessors.

Second, it shall be emphasized here, that the algorithm which was just presented delivers second order accurate solutions for the high fidelity model given with (28) and (29). Validation of this has been performed by numerous authors in a variety of studies. Examples are (Ten Cate, Nieuwstad et al. 2002, Li, Shock et al. 2004). Works on the validations of the boundary conditions applied in this work are for example (Aharonov and Rothman 1993, Chen, Martinez et al. 1996, Krüger, Kusumaatmaja et al. 2017). Two examples supervised by the author of the present thesis are (Beinlich 2021, Müller 2021). An example for validation with experimental data at high Reynolds numbers for turbulent flows is given in (Barad, Kocheemoolayil et al. 2017). In the following we will “use” this algorithm and expect results of a physical nature, both qualitatively and quantitatively. Wherever possibilities for comparisons with known behavior of fluids exist, this expectation is confirmed. Wherever we expect to reach limits for the application of the simulation technique, we point this out.

Third, note again that we have used a completely dimensionless formulation of the algorithm. This framework is fully in line with the fundamental relations introduced in Section 3.2. This will allow us to present the results in a compact dimensionless form. In this form, the dimensionless field values for the flow only depend on the freestream Reynolds number.

As a fourth last point, let us devote some attention to the simulation parameter τ^* . The solutions to our high fidelity model which are delivered by the presented algorithm, are for a fluid with a viscosity as given by

$$\nu^* = a^{*2}(\tau^* - 1/2). \quad (83)$$

Therein, a^* is a constant given by (69). Thus, there is a direct connection of the viscosity of the fluid with the relaxation parameter. If we denote a dimensionless reference length in our windtunnel with l^* (this could be the number of discretization points of the center wing chord), we have for the Reynolds number

$$\text{Re} = \frac{V_\infty^* l^*}{\nu^*} \quad (84)$$

Therein, $V_\infty^* = \text{Ma } a^*$ is generally given through the choice of a Mach number as high as possible, but still well below the limit at which compressibility effects begin to matter (see [Section 3.1](#)). Therefore, in order to achieve a high Reynolds number for a given discretization of the wind tunnel, there is no option but to choose the relaxation time as low as possible. Yet, at some point when $\tau^* \rightarrow 1/2$, the Lattice Boltzmann method becomes unstable. A common low value for τ^* for which we still have stable simulations is $\tau^* = 0.52$. How close τ^* can actually be to 0.5 for a given problem is slightly influenced by the choice of V_∞^* . Other slight influences may be the choice of boundary conditions and the problem geometry, see ([Krüger, Kusumaatmaja et al. 2017](#)). Once we have found a τ^* as low as possible for our problem at hand, we have no option but to increase the resolution of our simulation in order to model flows with higher Reynolds numbers. With higher Reynolds numbers, fine turbulent structures begin to occur in the flow. As the Reynolds number increases even further, the smallest length scales of these structures become even smaller and smaller ([Schlichting and Truckenbrodt 1967](#), [Anderson 2017](#)), see also ([Oberleithner, Sieber et al. 2011](#)). In a way, the Lattice Boltzmann method “demands” that we have a sufficient resolution of these fine details of the flow, otherwise the tool becomes unstable. To the author of this thesis, it seems remarkable that the tool is so well-connected to this nature of flows.

3.6 Output

The numerical wind tunnel is now ready to be used. This section shows how the results will be displayed.

3.6.1 Vorticity field

We will display results for the flow field in terms of an iso-surface through the dimensionless vorticity field color coded by the local Mach number. We will explain this visualization in the following:

After a simulation, we have full knowledge of the dimensionless velocity field $\mathbf{v}^*(x^*, y^*, z^*, t^*)$. The vorticity field can be obtained from this with

$$\boldsymbol{\xi}^* = \text{rot } \mathbf{v}^*, \quad (85)$$

compare (Anderson 2017). We have remained within our dimensionless framework in (85).

The dimensional vorticity $\boldsymbol{\xi} = \text{rot } \mathbf{v}$ can be recovered with

$$\boldsymbol{\xi} = \boldsymbol{\xi}^* / \zeta_t. \quad (86)$$

With the dimensionless vorticity field $\boldsymbol{\xi}^*(x^*, y^*, z^*, t^*)$ at hand, we are now able to display an iso-surface at a constant absolute dimensionless vorticity through this field. In the present work, we choose

$$|\boldsymbol{\xi}^*| = 0.01 \quad (87)$$

for all visualizations. It was found that with this value all the features in the flow which are of interest become visible, and this over a remarkable range of Reynolds numbers from 10^1 to 10^4 . We will discuss this choice frequently throughout the following section. The resulting iso-surface, we color with the local Mach number $\text{Ma} = |\mathbf{v}^*|/\mathbf{a}^*$. We choose the code given below for all visualizations.

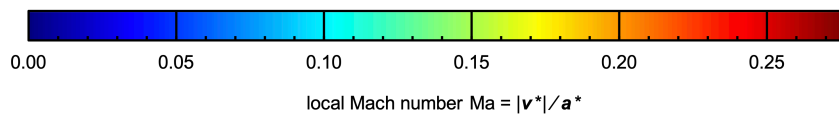


Figure 23: An iso-surface through the dimensionless vorticity field is colored by the local Mach number in this work. The color code used for this throughout all visualizations of this work is displayed in this figure.

3.6.2 Aerodynamic coefficients

The lift coefficient C_L is obtained as follows. With (37), we have

$$L = \frac{1}{2} \rho_{\infty} V_{\infty}^2 C_L S \quad (88)$$

for the lift L . In the present simulation, we will obtain the lift as a summation of the pressure distribution over the surface of our wing. For now, let us write this as

$$\sum p \Delta x \Delta y = \frac{1}{2} \rho_{\infty} V_{\infty}^2 C_L S. \quad (89)$$

In the dimensionless framework of the present work, we then have with (39), (45), (48), and (49)

$$\sum p^* \zeta_{\rho} \zeta_l^4 / \zeta_t^2 = \frac{1}{2} \rho_{\infty}^* \zeta_{\rho} V_{\infty}^{*2} \zeta_l^2 / \zeta_t^2 C_L S^* \zeta_l^2, \quad (90)$$

which is

$$\sum p^* = \frac{1}{2} \rho_{\infty}^* V_{\infty}^{*2} C_L S^*, \quad (91)$$

or

$$C_L = \frac{\sum p_b^* - \sum p_t^*}{\frac{1}{2} \rho_{\infty}^* V_{\infty}^{*2} S^*} \quad (92)$$

when displayed for the coefficient of lift. In (92), we have now displayed pressure distribution over the wing in terms of the pressure p_b^* on the bottom and the pressure p_t^* on the top of the wing. In the present isothermal Lattice Boltzmann model, pressure can be obtained via

$$p^* = p_0^* + a^{*2} \rho'^*, \quad (93)$$

where p_0^* is some constant reference pressure, such as the atmospheric pressure (Krüger, Kusumaatmaja et al. 2017). p_0^* is not relevant for our investigations as we will see in the following. ρ'^* is given via

$$\rho^* = \rho_0^* + \rho'^* \quad (94)$$

and describes the density fluctuations about some mean density ρ_0^* in the flow, for example the inflow density. Here again, the value of ρ_0^* is not relevant for our investigations, as we will see in the following. Inserting (93) into (92), we have

$$C_L = \frac{\sum(p_0^* + a^{*2}\rho_b'^*) - \sum(p_0^* + a^{*2}\rho_t'^*)}{\frac{1}{2}\rho_\infty^* V_\infty^{*2} S^*}, \quad (95)$$

which simplifies to

$$C_L = \frac{a^{*2}(\sum \rho_b'^* - \sum \rho_t'^*)}{\frac{1}{2}\rho_\infty^* V_\infty^{*2} S^*}. \quad (96)$$

Inserting (94), we have

$$C_L = \frac{a^{*2}(\sum \rho_b^* - \sum \rho_t^*)}{\frac{1}{2}\rho_\infty^* V_\infty^{*2} S^*}. \quad (97)$$

for the coefficient of lift.

Note that here we have only computed the coefficient of lift due to pressure which acts on the wing, similar to experiments where the coefficient of lift is measured with the pressure measured through small holes in the wing. In the real world, there may be some influence of surface friction on the wing which influences the lift coefficient. It is often assumed that surface friction only influences the drag coefficient, however, there are, albeit small, influences on lift. It is possible to take this into account in the Lattice Boltzmann method. However, this topic is beyond the scope of the present work. Here, we only use the pressure lift coefficient as a measure to check convergence of simulations and to check the quality of the results. In future works however, the tool may be extended to include full lift-, drag- and moment coefficients in the output.

In Section 3.2, we have seen that the lift coefficient only depends on the geometry of the problem, the freestream Reynolds number. This can be illustrated with (97). Even if we try to change ρ_∞^* for example (in this work we usually have set it to 1) we do not influence the Reynolds number and the results will remain exactly the same. ρ_b^* and ρ_t^* will just change accordingly. Indeed, ρ_∞^* is a mere numerical scaling parameter for the entire simulation and has no influence on the results. We will show this in more detail in the following section.

4 Results

In the past section, the experimental setup of our numerical wind tunnel was developed. Let us now use this setup to investigate some wing geometries which are of interest.

At first, a simple unswept wing with a symmetrical section will be investigated at a low Reynolds number. This will give us the opportunity to become familiar with the visualization and to observe some first results.

Afterwards, we will investigate the influence of wing sweep on the flow field at a given high angle of attack and a slightly higher Reynolds number than in the previous example.

The third exemplary study which we will perform is a single high resolution simulation on an exemplary Flying V geometry at a high angle of attack. This simulation took 20 days with the present setup. Ultimately, as mentioned in the introduction of this work, it will take a whole collection of numerical tools and quite a few real experiments to develop an understanding of the flow field around shapes as unconventional as the Flying V. The simulations in this work seek to add to this.

4.1 Finite wing

Let us first consider an unswept wing as shown in [Figure 24](#). Consider an unswept, untapered wing with symmetrical NACA 0014 sections and an aspect ratio of $\Lambda = b^*/c^* = 3.33$, where b^* is the span of the wing, and c^* is the chord. The flow around this wing is investigated at angles of attack ranging from $\alpha = 0^\circ$ to $\alpha = 60^\circ$ and a Reynolds number of $Re = 307$.

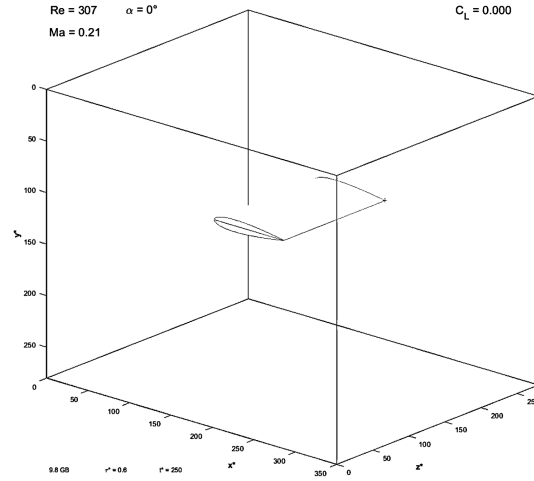


Figure 24: Geometry of the first investigation

Figure 25 shows exemplary results of this investigation at an angle of attack of $\alpha = 25^\circ$ after $t^* = 4000$ times steps. Let us first examine the iso-surface which is now wrapped around the wing geometry. As explained in detail in Section 3.6.1, this is a surface at a constant vorticity of $|\xi^*| = 0.01$ colored by the local Mach number as given in Figure 23. Prandtl's boundary layer theory states, that friction is only relevant in a small layer close to the wing's surface. The surrounding air behaves as potential flow. The visualization of the results in this work is inspired by this theory. Although we have, in the present example, a very low Reynolds number and Prandtl's theory was originally created for wings at much higher Reynolds numbers, we can already see in the present example some aspects of this theory. Note that the iso-surface at constant vorticity follows the wing's geometry very well. Vorticity is a measure for the level of friction within the fluid. For a potential flow for example, it is $\xi^* = \text{rot } \mathbf{v} = \mathbf{0}$, that is, there is no friction among the fluid elements. In the boundary layer, we have much higher friction, or higher vorticity. In the creation of the present work, we have experimented with a few values $|\xi^*|$ for the creation of the iso-surface. If we increases this value, the surface moves closer to the wing's first slowly, then rapidly, until it vanishes. If we decrease this value, the surface moves away from the wing, first slowly, then rapidly. For very small values of $|\xi^*|$ the wing geometry is not distinguishable anymore in the iso-surface. Although we have small Reynolds numbers, we can already observe these trends which are typical for a boundary layer. In a way, the displayed iso-surface *encloses* a significant amount of the boundary layer. Of course there can be no full enclosure, even at higher Reynolds numbers. There is always some amount

of friction among the fluid elements. However, this is highest in a layer close to the surface of the wing, even in the present example. This layer follows the geometry of the wing very well. As expected, behind the wing tip, we observe some increased vorticity due to the wing tip vortex which arises due to the creation of lift with the finite wing.

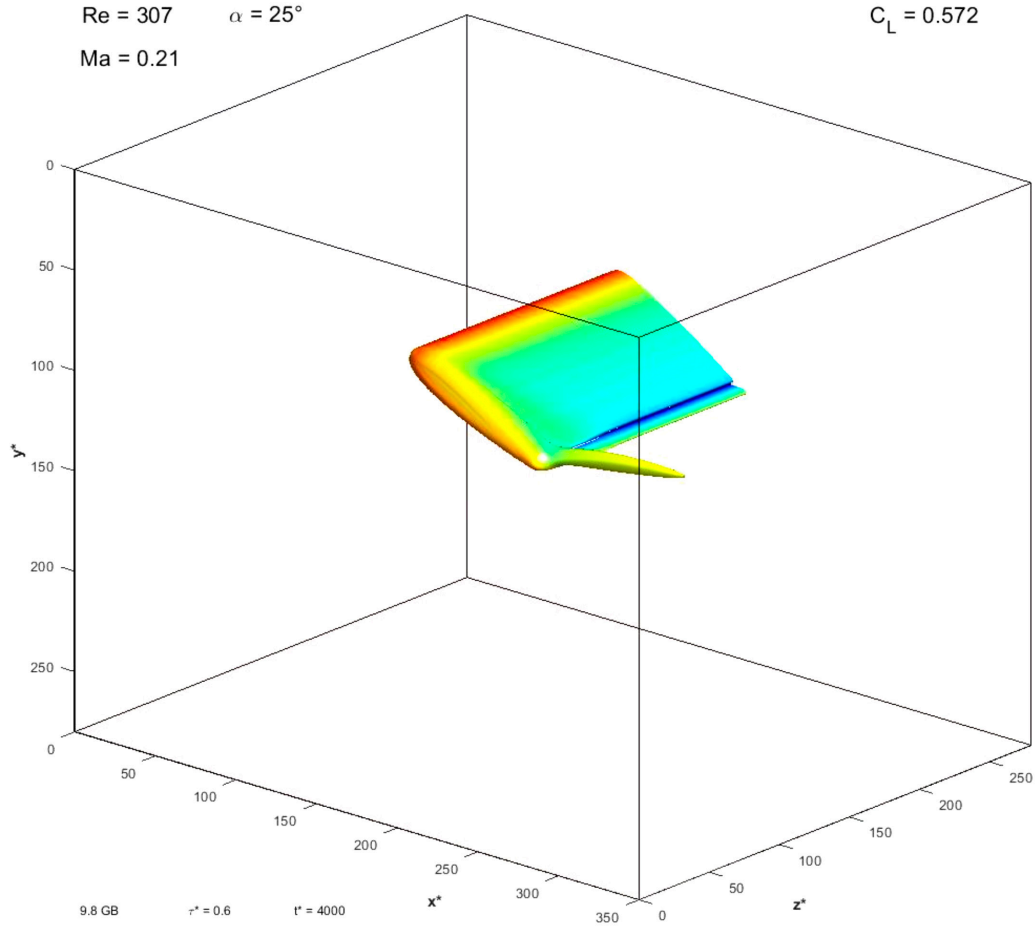


Figure 25: Iso-surface of constant vorticity $|\xi^*| = 0.01$ colored by the local Mach number displayed after $t^* = 4000$ timesteps for the first investigation of a finite wing with NACA 0014 sections and an aspect ratio $\Lambda = 3.33$ at a Reynolds number of $Re = 307$ for an angle of attack $\alpha = 25^\circ$.

Note also, that the surface which encloses the wing is fairly smooth, as are the colors of the local Mach number displayed on this surface. This is a full laminar flow where the fluid elements move in well-ordered layers, as we would expect at such low Reynold numbers.

We can further observe, that the local Mach numbers are highest at the nose of the wing and that we have a definite decrease in Mach number at the upper surface of the wing. At this stage one should note that with our iso-surface, we are still somewhat within the boundary layer, that is, fairly close to the wing, and here velocities are generally a bit smaller.

Let us now examine the time evolution of the lift coefficient in the present example. The result displayed in [Figure 25](#) for an angle of attack $\alpha = 25^\circ$ is well-converged. No change in the iso-surface or colors could be observed from a time step of $t^* = 1500$ onward. A way to measure this is the pressure lift coefficient we have introduced in [Section 3.6.2](#). Below, in [Figure 26](#), we show the pressure lift coefficient displayed over the simulation time for the present example at an angle of attack of $\alpha = 25^\circ$.

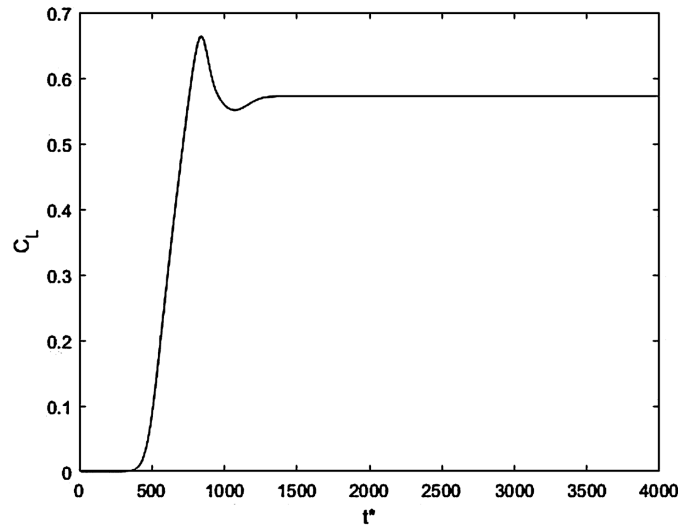


Figure 26: Pressure lift coefficient C_L displayed over the simulation time for the first investigation of a finite wing with NACA 0014 sections and an aspect ratio $\Lambda = 3.33$ at a Reynolds number of $Re = 307$ for an angle of attack $\alpha = 25^\circ$. Three graphs are plotted in the figure, one for $\rho_\infty^* = 1$, one for $\rho_\infty^* = 10$ and another one for $\rho_\infty^* = 100$. As expected, the results do not depend on this parameter so that only a single line is visible.

First, we observe no lift, as the incoming fluid takes its time to reach the wing geometry which thus far was surrounded by a resting fluid. At around $t^* = 400$, we observe a gradual increase in the lift coefficient which becomes steep at around $t^* = 500$. The maximum value of C_L is reached at around $t^* = 820$. At this time, the incoming flow reaches the trailing edge of the wing. A trailing edge vortex forms and dissolves again as the lift coefficient converges at around $t^* = 1500$. Note that such an overshooting of the lift coefficient before a lower converged value is reached is typically encountered as an aircraft flies through a gust, see for example the experimental results in [\(Kramer 1932\)](#).

Let us further note that the curve displayed in [Figure 26](#) are actually three curves for three different densities ρ_∞^* displayed on top of each other. We have conducted one simulation with the standard value of our setup of $\rho_\infty^* = 1$, and two more simulations with values of

$\rho_\infty^* = 10$ and $\rho_\infty^* = 100$. As expected and explained in [Section 3.6.2](#), the three curves lie exactly on top of each other. The dimensionless density in the Lattice Boltzmann method is a mere numerical scaling parameter. Also recall that we have found out with the fundamental relations in [Section 3.2](#) that our results will only depend on the Reynolds number $Re = V_\infty c / \nu_\infty$. The density does not appear here, therefore, there is no dependence on it in the dimensionless results. This is reflected in the results above.

A single simulation from above until time step $t^* = 4000$ takes about six hours with the present setup. Several of these simulations were conducted at angles of attack ranging from $\alpha = 0^\circ$ to $\alpha = 60^\circ$. The results for the lift coefficients after $t^* = 4000$ times steps are displayed in [Figure 27](#). Each one of the simulations had long converged before the end of the simulation time. Let us now examine the results. We do not expect the constant lift slope $C_{L,\alpha} = 2\pi/(1 + 2/\Lambda)$ obtained in theoretical potential flow solutions ([Anderson 2017](#)). This would correspond to a case where $Re \rightarrow \infty$. For lower Reynolds numbers, the lift slope is generally lower than that ([Spedding and McArthur 2010](#)), but we would still expect a linear dependence of the lift coefficient on the angle of attack, until at some point the maximum lift is reached and the lift coefficient begins to drop again, see for example the experimental results obtained in ([Taira and Colonius 2009](#)). For a thick profile such as the NACA0014 as chosen in this work, we would expect this stalling behavior at large angles of attack to be slow, rather than a sudden drop in the lift coefficient ([Schlichting and Truckenbrodt 1967](#)). With these expectations, we are not disappointed in the results below. We have highlighted the linear dependence at lower angles of attack, and we can observe a slow drop in the lift coefficients at higher angles of attack.

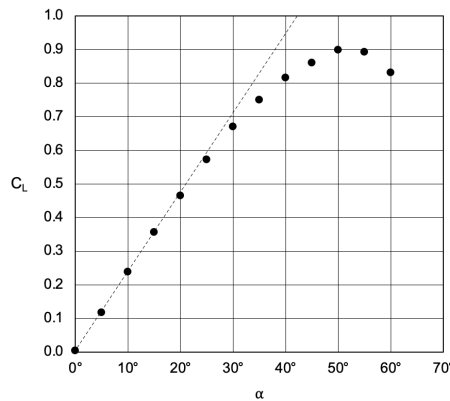


Figure 27: Pressure lift coefficients after $t^* = 4000$ timesteps for simulations at angles of attack ranging from $\alpha = 0^\circ$ to $\alpha = 60^\circ$ for the first investigation of a finite wing with NACA 0014 sections and an aspect ratio $\Lambda = 3.33$ at a Reynolds number of $Re = 307$. The dashed line is drawn into the figure to illustrate a linear dependence at low angles of attack.

4.2 Influence of sweep

Let us now use the developed numerical wind tunnel for a second investigation. We consider a swept, untapered wing with symmetrical NACA 0014 sections and an aspect ratio of $\Lambda = b^*/c^* = 2$. The flow around this wing is investigated at an angle of attack $\alpha = 30^\circ$ and the Reynolds number is $Re = 417$. Eight different sweep angles ϕ are examined. In the present work, ϕ describes the sweep of the leading edge. $\phi = 0^\circ$ corresponds to a straight leading edge such as in the previous investigation. In the following, we will also frequently make use of a parameter we call the *sweep ratio* \mathcal{R} . We introduce this parameter as

$$\mathcal{R} = \tan \phi . \quad (98)$$

The eight sweep ratios we will investigate in the following range from $\mathcal{R} = 0$ to $\mathcal{R} = 2.8$, which corresponds to a range of sweep angles from $\phi = 0^\circ$ to $\phi = 70.3^\circ$. While we set these various sweep ratios, parameters such as the span b^* , the chord c^* , and the wing area, here $S^* = b^*c^*$, remain the same.

Let us first examine exemplary results of a simulation with $\mathcal{R} = 1.6$. The converged flow field is displayed in [Figure 28](#). A simulation such as this one takes one and a half days with the present setup.

Note first, that we observe a laminar flow, that is, the flow particles move in well-ordered layers over the surface of the wing.

Second, note that these results we show are indeed converged results. No changes in the lift coefficient and the vorticity field are visible from this moment onward, that is, we have a stationary solution for our flow field. At even larger angles of attack, we might expect unstationary behavior at some point, but for the present investigation the results remain stationary.

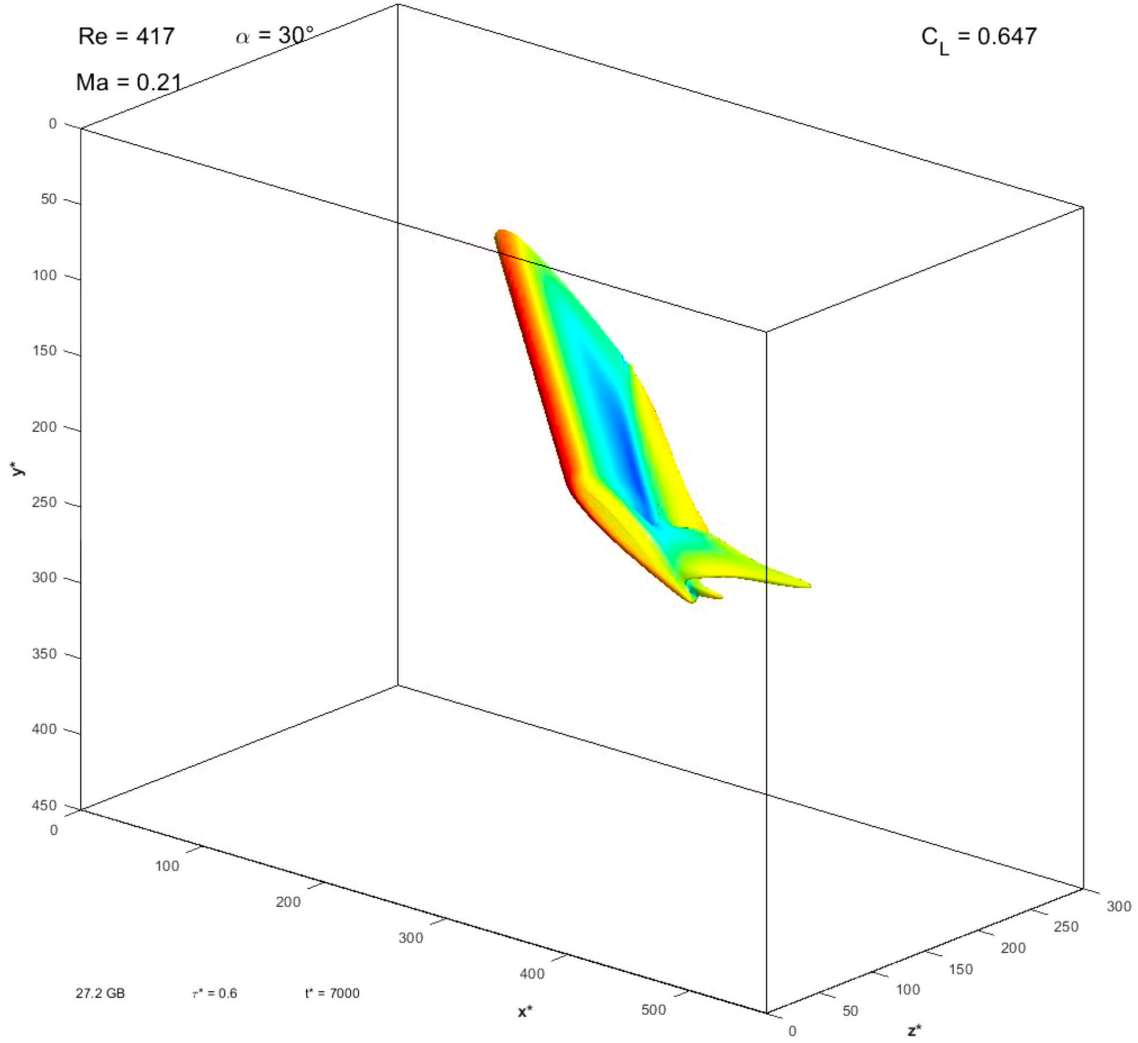


Figure 28: Iso-surface of constant vorticity $|\xi^*| = 0.01$ colored by the local Mach number displayed after $t^* = 7000$ timesteps for a finite wing with NACA 0014 sections and an aspect ratio $\Lambda = 2$ at an angle of attack $\alpha = 30^\circ$ and a Reynolds number of $Re = 417$. The sweep ratio is $\mathcal{R} = 1.6$, the sweep angle is $\phi = 58.0^\circ$.

Third, let us note that the surface of constant vorticity we use to visualize the flow field still aligns well with the surface of the wing. Although we observe a variety of different local velocities, the vorticity field is given to a large extent by the surface of the wing.

Let us draw attention to two regions of the field where the vorticity field differs from the wings geometry.

The first region is the tip region of the wing. As we have observed in the previous investigation, we have some concentrated vorticity in this area due the location of the tip vortex which arises from the creation of lift with the wing (Anderson 2017). On the swept

wing in the present example, even more shear surfaces arise in this tip region. For larger sweep angles, these shear surfaces grow more and more sophisticated and begin to merge with other shear surfaces of the wing. We will discuss this further below when we investigate other sweep angles.

The second region where we observe a region of vorticity other than a mere boundary contour of the wing is the region at the trailing edge of the wing. This is the characteristic free shear layer which arises as the flows on the upper and lower surface of the wing which have different directions and speeds begin to intermingle after they have passed the trailing edge ([Schlichting and Truckenbrodt 1967](#)).

Let us now observe this image from behind, that is, in the $y^* - z^*$ plane. And let us also not only examine a single sweep ratio, but all eight investigated sweep ratios. The converged flow fields are displayed in [Figure 29](#).

With this visualization we can observe in detail how the vorticity field is influenced by the geometry of the wing. Let us first examine the shear layers. It becomes apparent that the free shear layer on the trailing edge of the wing grows as the sweep is increased. For the unswept configuration, we have a very small shear layer at the trailing edge of the wing, see marker **I**. In fact, it is connected to the tip vortex **II** as we can see in the top image. In this first image, note also that above the tip vortex, the iso-surface around the wing edge is not smooth but features a distinct line **III**. As the sweep ratio is increased to $\mathcal{R} = 0.8$, we first note that we have actually two separate shear layers, the first one, marked with **I**, directly connected to the trailing edge of the wing, while the second one, here **II**, is connected to the original tip vortex which has moved slightly upwards. This distinction becomes more and more apparent as the wing sweep is increased further. One more feature to note in the third image is the occurrence of a region of low velocity in the boundary layer on the upper surface of the wing. In the fourth image, this region has become even more pronounced with a clear lower boundary at a line we denote with **IV** and another clear oblique boundary to the right which we denote with **V**. With even more sweep, $\mathcal{R} = 1.6$, the dark blue region of lower velocity begins to extend to the tip vortex where it meets with line **III**. Meanwhile the lower boundary **IV** of the region of low speed has become quite distinct and slowly moves upwards. We can only conclude that a vortex is induced by the

trailing edge which rotates counter clockwise. As the sweep is increased the influence of this vortex on the flow field of the upper surface becomes more and more pronounced. For $\mathcal{R} = 2.0$, **IV** moves up even further. Below this line, we have higher flow velocities due to the influence of this trailing edge vortex. Directly at line **IV** where the vortex meets the main flow around the wing, we can even witness the formation of another small free shear layer. The other previously discussed shear layers **I** and **II** are now visible more clearly than ever, with a clear gap between them at the wing tip. In fact, it becomes apparent that shear layer **II** is now the clear boundary between the trailing edge vortex, which rotates counter clockwise and the tip vortex, which rotates in clockwise direction. This phenomenon remains visible for all higher sweep angles and we will discuss it in more detail throughout this work. Already with the present simulation it is illustrated that for highly swept wings at high angles of attack we have these influences on the flow field: A trailing edge vortex which rotates in counter clockwise direction, and some sort of tip vortex which rotates in clockwise direction. In the present situation, where both meet we find shear layer **II**. When the sweep is increased even more to $\mathcal{R} = 2.4$, this shear layer connects with the shear layer along line **IV**, which is to say, the region of low velocity becomes the small footprint of those two influences. The influence of the trailing edge vortex over the wing intensifies for even larger sweep angles as can be seen for the case of $\mathcal{R} = 2.8$. Here we have also sketched the discussed phenomena for the flow using some rounded arrows to illustrate rotation directions of the flow. At this stage, we should note that the overall lift of the wing results in a clockwise rotation of the flow due to the downwash behind the wing. We observe the same here, yet we also observe the trailing edge vortex working against this at the given high angle of attack. This influence rises for higher sweep angles. It is interesting to note that we also observe a decrease in the pressure lift coefficient for higher sweep angles, see [Figure 30](#). The trailing edge vortex may not be the only cause of this behavior, but it may well be of influence.

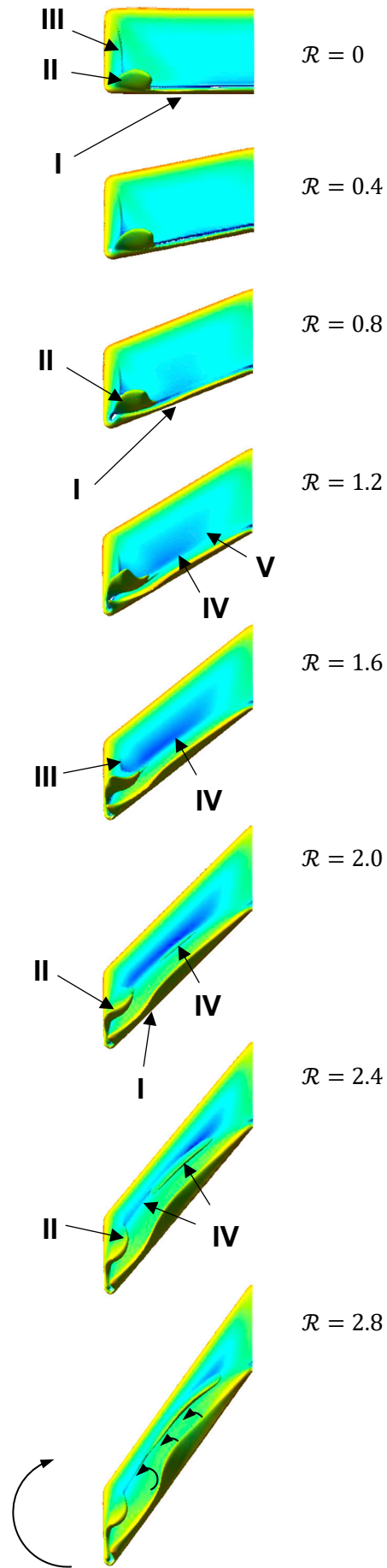


Figure 29: Setup from Figure 28 seen from behind. Converged results are shown after $t^* = 7000$ timesteps for sweep ratios ranging from $\mathcal{R} = 0$ to $\mathcal{R} = 2.8$. The annotations are explained in the text above.

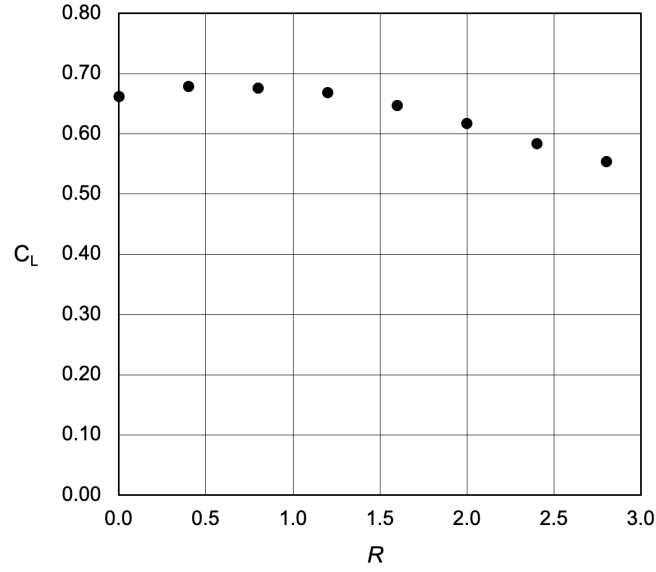


Figure 30: Pressure lift coefficients after $t^* = 7000$ timesteps for simulations with sweep ratios ranging from $\mathcal{R} = 0$ to $\mathcal{R} = 2.8$ of a finite wing with NACA 0014 sections and an aspect ratio $\Lambda = 2$ at an angle of attack $\alpha = 30^\circ$ and a Reynolds number of $Re = 417$.

4.3 High resolution simulation

Let us now conduct a single high resolution simulation. Therein, we will set the Reynolds number as high as possible with the given setup. As wing layout we will investigate an exemplary shape of the Flying V. The wing layout can be seen in [Figure 31](#). The wing which is displayed consists out of three elements: A highly swept middle section, a transition section, and an outer wing element with lower sweep. The leading edge sweep of the transition section is equal to the leading edge sweep of the middle section, and the trailing edge sweep of the transition section is equal to the trailing edge sweep of the outer wing section. This design will be investigated at an angle of attack of $\alpha = 30^\circ$.

The setup for our high-resolution simulation can be seen in the image below.

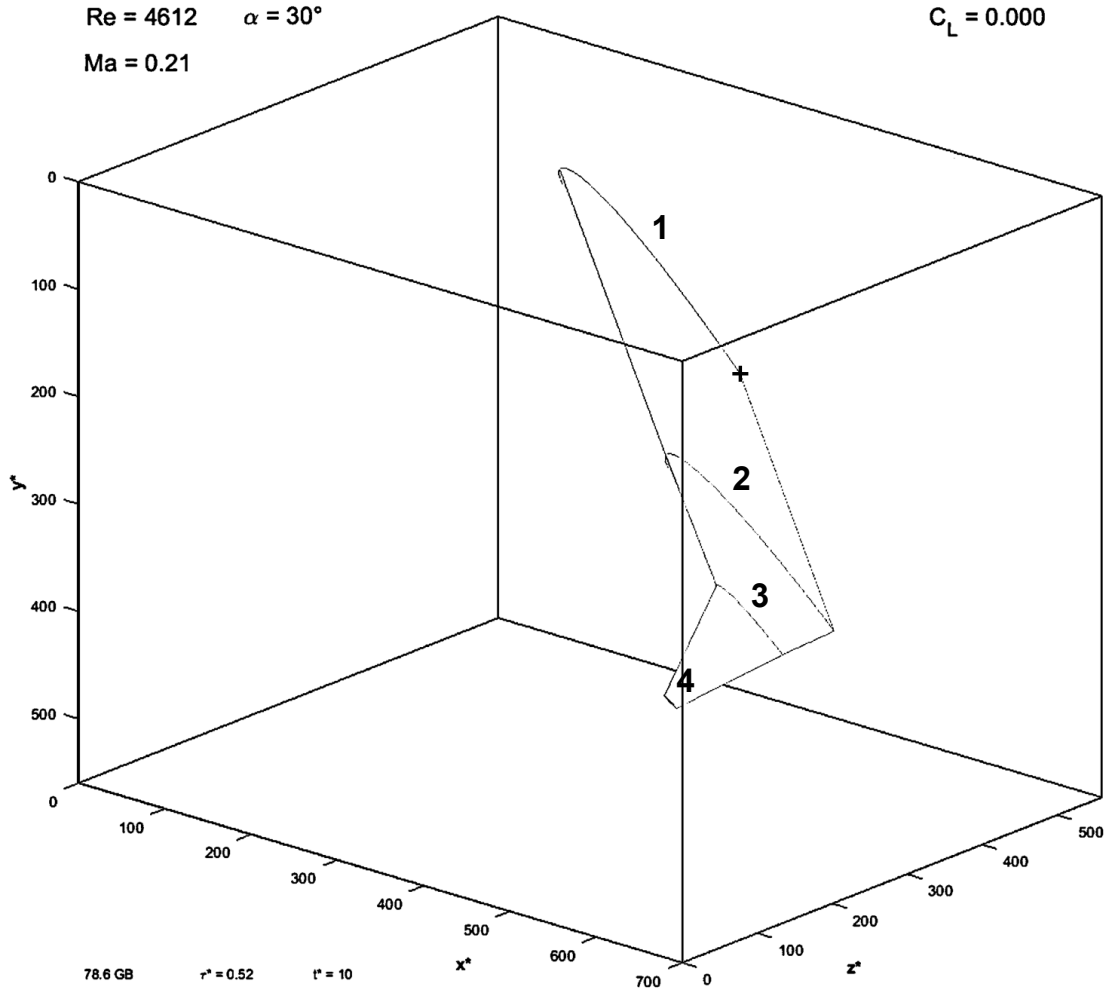


Figure 31: Wing geometry for the high resolution investigation

The span b^* of the investigated model is chosen with $1.45 B^*$. The geometry is given by four profile sections. As profiles we choose symmetric four digit NACA sections.

The intersection $+$ of the trailing edge with the first profile section **1** in the symmetry plane is located at $x^* = 0.4 S^*$, $y^* = 0.48 H^*$. The first profile has a thickness of 13.5%. As local angle of incidence of the first profile we choose $\alpha_l = 4^\circ$. The length of the chord of the first profile section is chosen with $0.31 b^*$.

The trailing edge point of the second profile section **2** is located $0.35 b^*$ behind $+$ in x^* direction, and $0.01 b^*$ above it in negative y^* direction. In negative z^* direction, it is at a distance of $0.2 b^*$ from $+$. The second profile also has a thickness of 13.5%. As local angle of incidence, we choose $\alpha_l = 1^\circ$. The length of the chord is chosen with $0.28 b^*$.

The trailing edge point of the third profile section **3** is located $0.36 b^*$ behind **+** in x^* direction, and $0.02 b^*$ above it in negative y^* direction. In negative z^* direction, it is at a distance of $0.3 b^*$ from **+**. The profile has a thickness of 11%. As local angle of incidence, we choose $\alpha_l = 1.1^\circ$. The length of the chord is chosen with $0.11 b^*$.

The trailing edge point of the outermost profile section **4** is located $0.38 b^*$ behind **+** in x^* direction, and $0.03 b^*$ above it in negative y^* direction. In negative z^* direction, it is at a distance of $0.5 b^*$ from **+** (half-span). The profile has a thickness of 10%. As local angle of incidence, we choose $\alpha_l = 2^\circ$. The length of the chord is chosen with $0.02 b^*$.

After the geometry was constructed this way, the entire wing is turned around **+** along the z^* axis by an angle of $\alpha = 30^\circ$.

Let us now investigate the results. Let us begin this by looking at the time evolution of the lift coefficient during the simulation, see [Figure 32](#). $t^* = 20000$ timesteps were simulated, this corresponds to a calculation time of roughly 20 days.

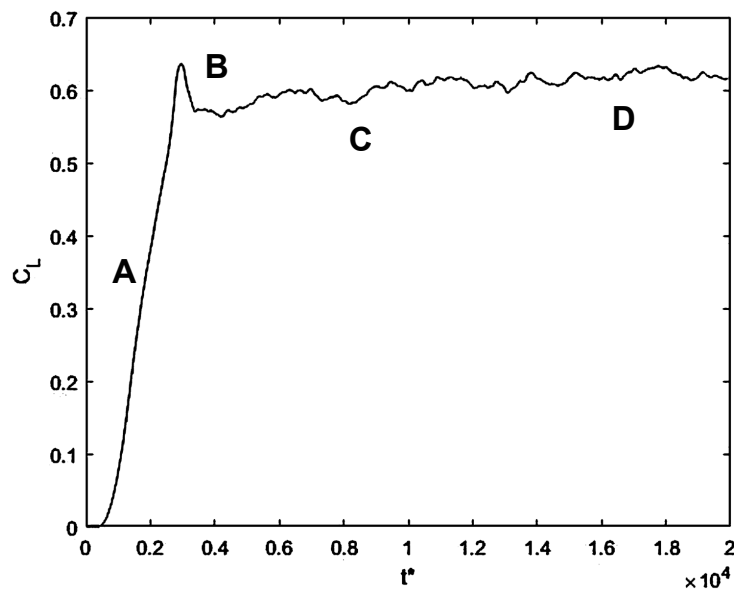


Figure 32: Time evolution of the pressure lift coefficient for the high resolution simulation. Discretization points: 220 million, timesteps: 20 thousand. The annotations are explained in the text above.

The simulation begins with a resting fluid, that is, the pressure lift coefficient is zero at the beginning. As the incoming flow begins to sweep over the wing geometry, the lift

coefficient begins to rise. We have marked an exemplary positions during this stage with **A**. The corresponding iso-surface of the vorticity can be seen in [Figure 33](#). Note already at this stage, that there are some waves in the iso-surface which form at the leading edge of the highly swept region of the wing.

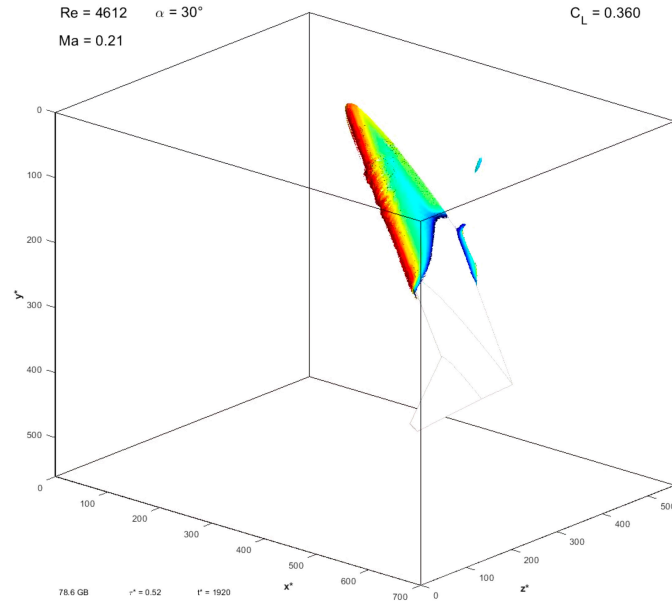


Figure 33: Iso-surface of constant vorticity $|\xi^*| = 0.01$ colored by the local Mach number displayed after $t^* = 1920$ timesteps for the high-resolution simulation.

As the flow continues, the lift coefficient rises even higher. It reaches its maximum value the moment the flow reaches the outer trailing edge. We mark this point with **B**. Right after this, a trailing edge vortex begins to form. This stage is displayed in [Figure 34](#) below.

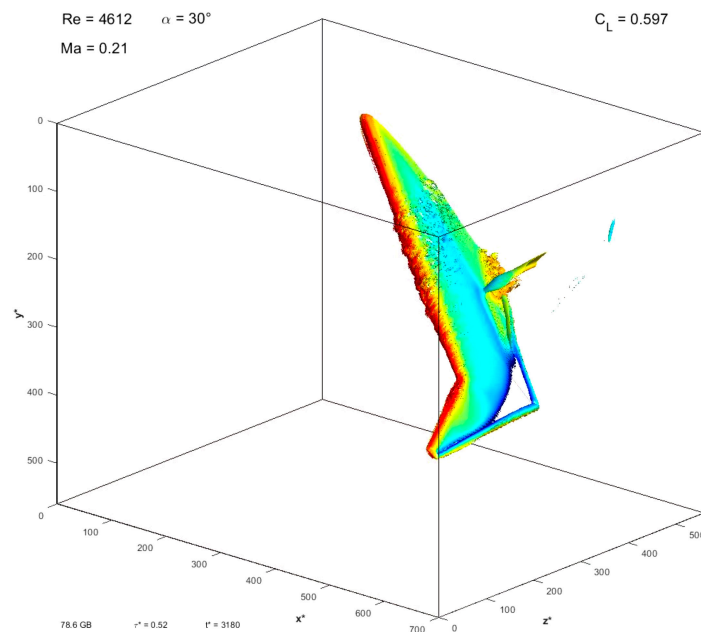


Figure 34: Iso-surface of constant vorticity $|\xi^*| = 0.01$ colored by the local Mach number displayed after $t^* = 3180$ timesteps for the high-resolution simulation.

Most remarkable in the coming timesteps is the growth of the small waves which now begin to appear everywhere on the wing geometry. They do not only originate from the leading edge but begin to appear all over of the wing's surface. These waves slowly move to the back of the aircraft. After a while, no new waves originate In the front of the geometry and the flow in this region begins to calm. Everywhere else, the waves remain present for quite a while. We call this stage **C**. An exemplary snapshot from this stage at $t^* = 7450$ is displayed in the image below.

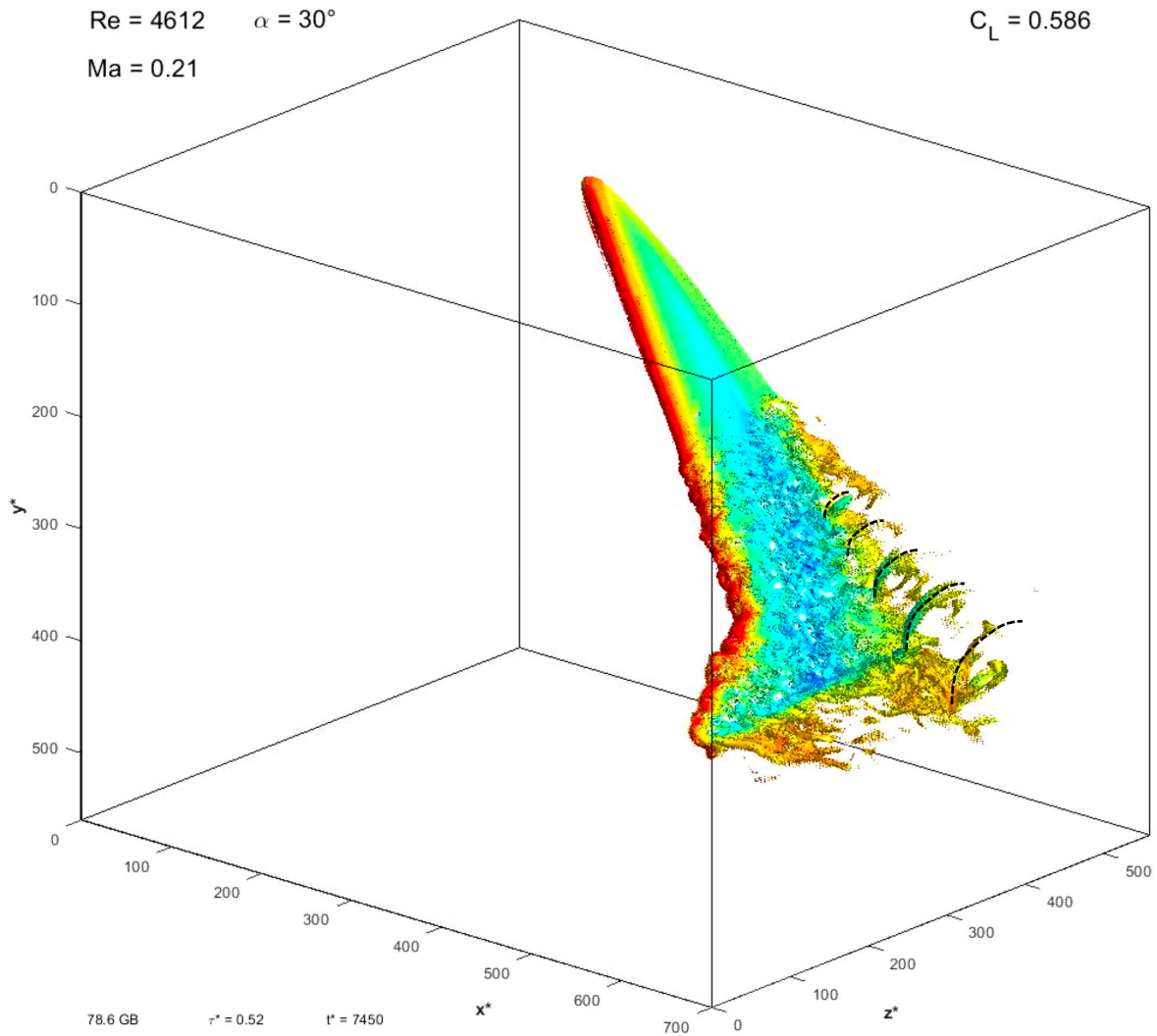


Figure 35: Iso-surface of constant vorticity $|\xi^*| = 0.01$ colored by the local Mach number displayed after $t^* = 7450$ timesteps for the high-resolution simulation.

Note also the coherent structures which begin to appear more to the rear of the aircraft. We have highlighted these structures with dashed black lines.

This flow field in Figure 35 is interesting, because through the abrupt start of the flow the level of turbulence is higher than one would expect at the present Reynolds number. The influence of the rough beginning starts to decrease as the simulation moves on and the flow begins to calm further in certain regions. In other regions, instabilities remain present and distinct features begin to show. At around $t^* = 14000$ most of these features do not change anymore. Until now, the lift coefficient was steadily increasing, now this tendency begins to cease. The image displayed below at $t^* = 16450$ is a typical snapshot of a flow where no features of the main flow change anymore. Naturally, there are oscillations, but these tend to repeat themselves in various ways. We call this stage **D**. Let us now discuss some of the features of this flow.

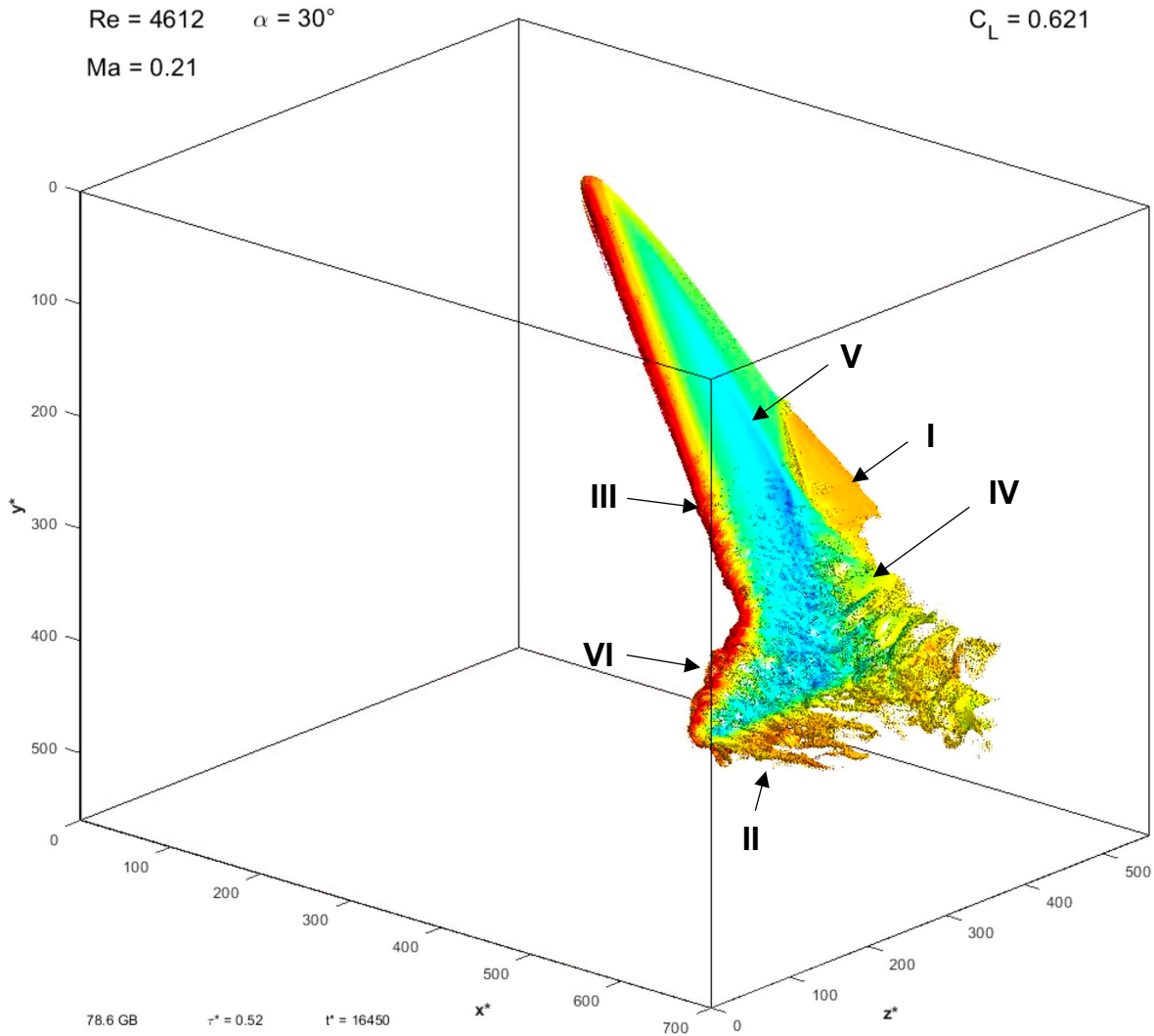


Figure 36: Iso-surface of constant vorticity $|\xi^*| = 0.01$ colored by the local Mach number displayed after $t^* = 16450$ timesteps for the high-resolution simulation.

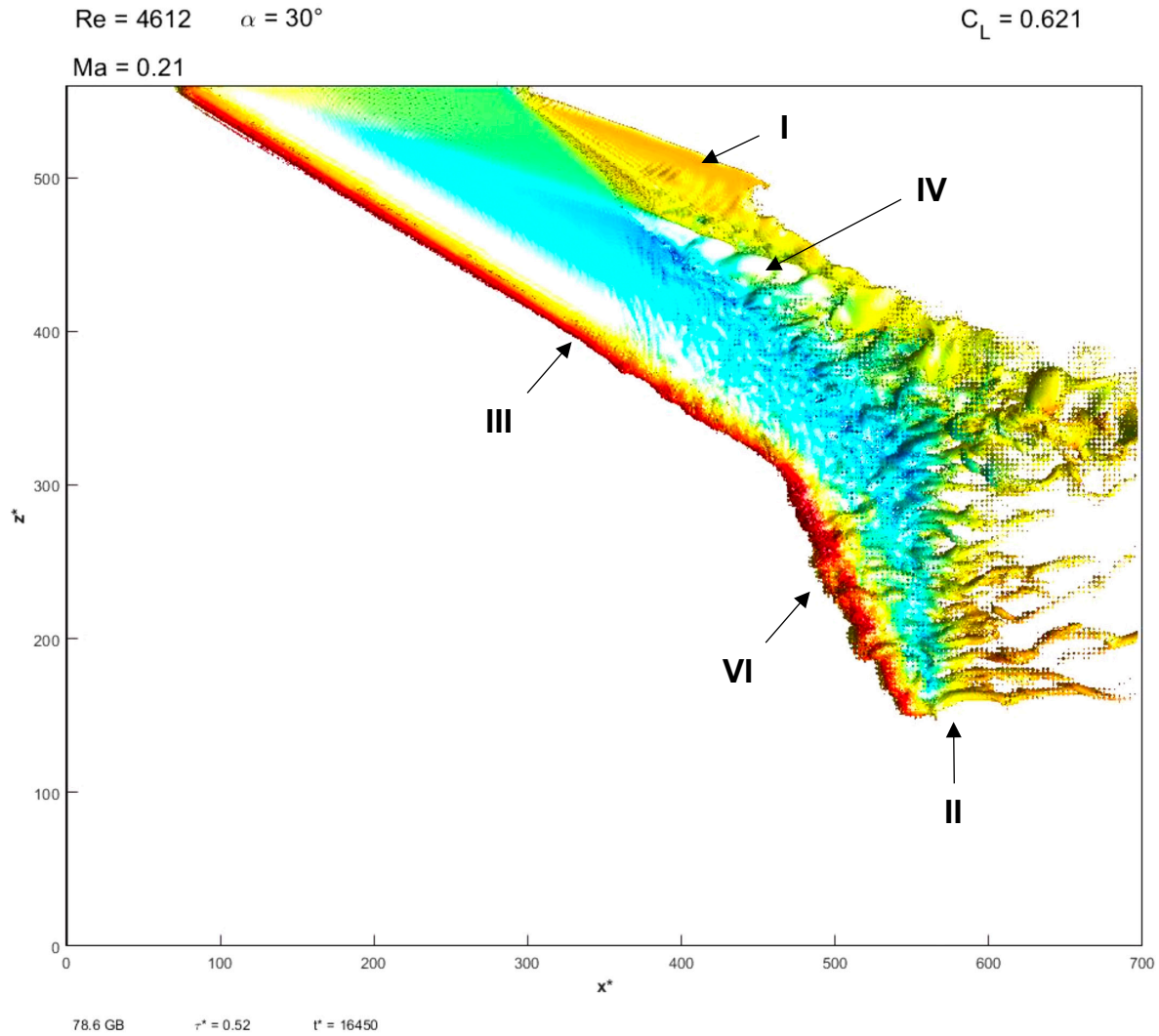


Figure 37: Iso-surface of constant vorticity $|\xi^*| = 0.01$ colored by the local Mach number displayed after $t^* = 16450$ timesteps for the high-resolution simulation, top view.

First, note the free shear layer **I** at the highly swept trailing edge of the design. We already know this shear layer from our previous study with smaller Reynolds numbers. In the present study, waves appear in the shear layer. Further down the flow, the shear layer dissolves in some turbulent fashion. It induces a vortex quite similar to the previous study. The increased flow velocities due to the presence of this vortex can be seen best in [Figure 38](#) in yellow right above the beginning of the shear layer. We have sketched this motion with some rounded arrows. The flow in this yellow region on the becomes unstable as well and begins to intermingle with the instable free shear layer further down the flow. Above this region, we find the familiar shear layer **IV**. This layer forms due to the counteracting influences of the trailing edge vortex we just described, and the main flow which sweeps over the wing as roughly sketched with the dashed line in [Figure 38](#). Contrary to previous

investigations, this shear layer also becomes unstable. The cause of this is visible quite nicely in the simulations. In the present setup, waves begin to form in the boundary layer at the leading edge of the design, see **III**. These waves remind of Tollmien–Schlichting waves which are a classical phenomenon in laminar turbulent transitions ([Tollmien 1931](#), [Schlichting 1933](#), [Schlichting and Gersten 2006](#)). Some of these waves form high up front on the wing. They move in small packages inward over the surface of the wing until they meet the beginning of the shear layer **IV**. This leads to the occurrence of characteristic coherent structures in the shear layer. Further down the flow, the shear layer intermingles strongly with the trailing edge vortex, the influence of the main flow, and with vortices which form more to the outside of the wing and follow the classical clockwise rotation.

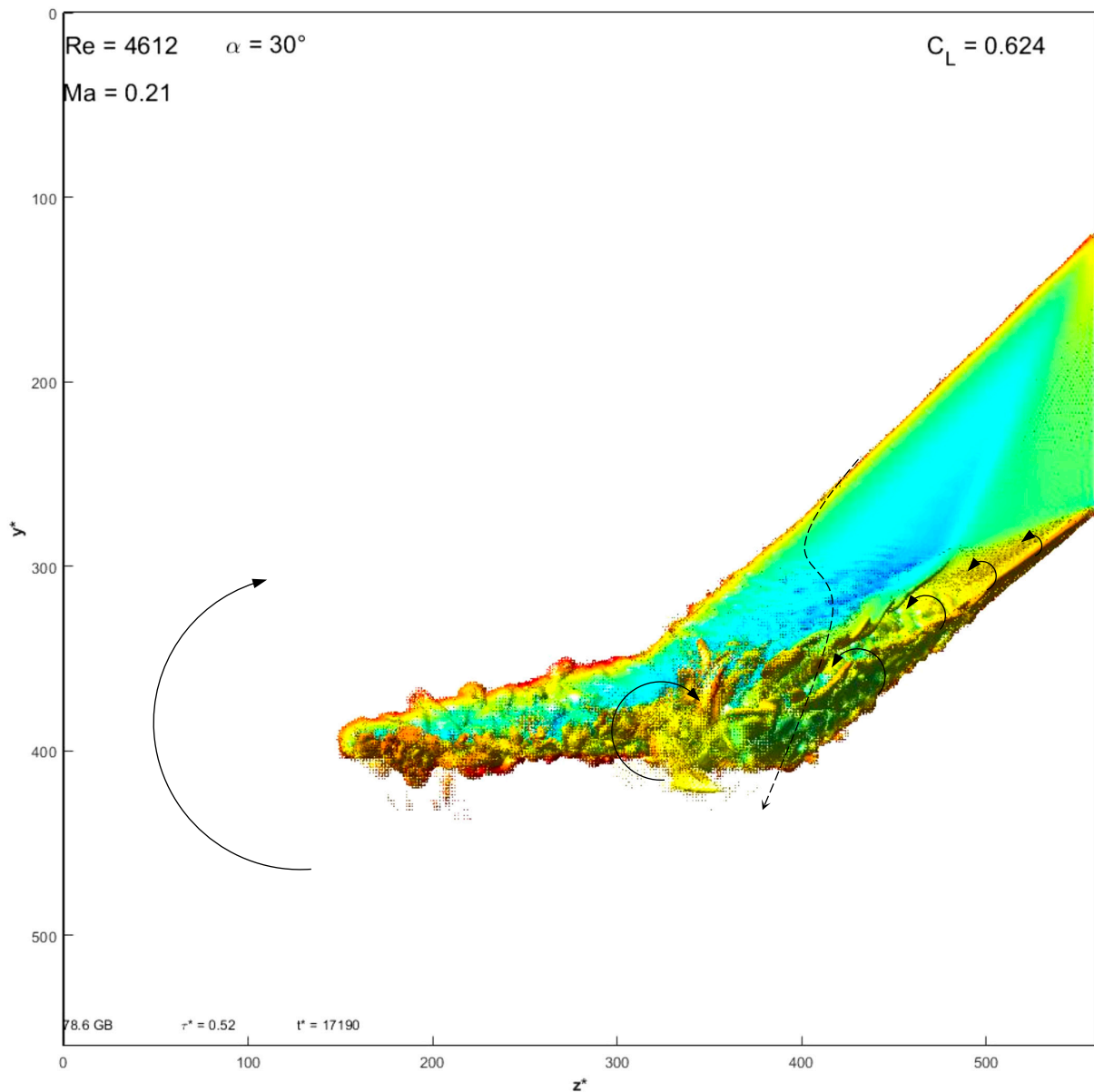


Figure 38: Iso-surface of constant vorticity $|\xi^*| = 0.01$ colored by the local Mach number displayed after $t^* = 17190$ timesteps for the high-resolution simulation, rear view.

Further to the outside of the wing the free shear layer **II** at the trailing edge separates in some distinct vortices. Some of these vortices begin to form at the leading edge **VI**. The motion of these vortices was observed to be highly irregular. Their formation seems to originate at a region close to the king in the leading edge. From there, they move to the outside along the leading edge in a highly disordered fashion. Wind tunnel experiments at higher Reynolds number indicated the presence of a leading edge vortex which forms in this region and bursts at high angles of attack. The irregular fashion we observe here in the present study could be a forerunner of this. Right at the wing tip, we observe that the tip vortex forms as expected. There are a few oscillations, however the overall visibility of this feature remains quite steady. Furthermore, we observe small vortex fragments which separate from the wing at the trailing edge and are then caught by the downwash of air behind the wing.

The present study is the beginning of further experiments which can now be conducted with the developed wind tunnel. Especially simulations with even higher resolutions will be of great interest to describe some of the features which were explained above in even more detail and to investigate how they change as the Reynolds number is increased even further. Also, various investigations of the present highly swept flying wing geometry are currently underway by other authors. It will be very interesting to compare the results and to thereby develop a full understanding of the flow field around the geometry at high angles of attack.

The conclusion in the next section contains various aspects which will be interesting for future works on developed flow channel. As a closing argument at this stage, let us draw attention again to the simplicity of the tool at hand and the remarkable level of detail in the results which is visible above.

5 Conclusion and outlook

In this work, a numerical flow channel was developed with the Lattice Boltzmann method and applied to study various swept wing geometries. At first, the Lattice Boltzmann method was discussed in the light of other numerical tools, and in the light of current development trends of microprocessors. It was then decided to use the Lattice Boltzmann method for the creation of the numerical flow channel. The setup of this numerical experiment was explained with a particular focus on the numerical details which influence the maximum Reynolds number which can be achieved. Finally, the developed setup was used to conduct three exemplary studies. First, a finite unswept wing was investigated at various angles of attack. Then, the influence of sweep of a low aspect ratio wing at high angles of attack was investigated. In the third study, a high resolution simulation of an exemplary Flying V geometry was conducted in order to obtain a first impression of the flow field behind the configuration at high angles of attack.

The results of this effort are as follows:

First, it was highlighted that numerical techniques with simple local numerical interactions in spatial domain allow for efficient computations. It was emphasized that such techniques are well-suited for parallelization.

Second, it was highlighted that there is a strong trend of an increase of parallel operations in the development of microprocessors. It was made clear that if we expect this trend to continue for some years, numerical tools which consist of simple parallel operations in the spatial domain like the Lattice Boltzmann method, are likely to benefit from this in the future.

Third, insights of how large arrays can be handled and arranged with the Lattice Boltzmann method were presented. Implications this has on the maximum amount of memory required by the method were discussed. For these details, please see [Section 3.5](#).

Fourth, the flow field behind a highly swept wing at low Reynolds numbers was described as a function of the sweep ratio. For these results, please see [Section 4.2](#).

Fifth, a first description of the flow field behind a Flying V configuration at a high angle of attack was offered. The results of this study at $Re = 4.6 \cdot 10^3$ and an angle of attack of $\alpha = 30^\circ$ are, a) the observation of waves in the boundary layer which form at the leading edge and move inward towards the rear kink, see [Figure 37](#), b) the discovery of two characteristic free shear layers **I** and **IV**, see [Figure 36](#), c) the observation of strong interactions of the main flow and the two shear layers close to the rear kink, see [Figure 38](#), refer to [Section 4.3](#) for a more detailed description, and d) the discovery of turbulent coherent structures in the vorticity field as seen in [Figure 35](#) after a strong initial gust.

Recommendations for future work on this particular last investigation include more studies with the present tool at other angles of attack, the investigation of higher resolutions, the investigation of geometry changes of the design, and the comparison of the results of the present study with wind tunnel experiments. It is interesting to note that the tool which is developed in this work is fully scalable and can be applied without any further modifications on larger computers than in the present work to model higher Reynolds numbers in future works.

Recommendations for future work on the flow channel are the inclusion of other collision operators (TRT / MRT models). Such more refined operators may help to further reduce the relaxation time and thus may allow for the simulation of higher Reynolds numbers at a given amount of memory ([Krüger, Kusumaatmaja et al. 2017](#)). A further recommendation to extend the range of application of the flow channel is an investigation on how to include compressibility effects, see for example ([Fares, Wessels et al. 2014](#)).

In closing, let us draw attention again to the simplicity of the tool at hand and the remarkable level of detail in the results which is visible above.

References

- Aharonov, E. and D. H. Rothman (1993). "Non-Newtonian flow (through porous media): A lattice-Boltzmann method." *Geophysical Research Letters* 20(8): 679-682.
- Aidun, C. K. and J. R. Clausen (2010). "Lattice-Boltzmann method for complex flows." *Annual review of fluid mechanics* 42: 439-472.
- Anderson, J. (2017). *Fundamentals of Aerodynamics*, McGraw-Hill Education.
- Ankith John Santosh, A. (2020). "Numerical Investigation of the Influence of Ground Effect on the FV Aircraft: Influence of Ground Effect on the Flying V Aircraft."
- Barad, M. F., J. G. Kocheemoolayil and C. C. Kiris (2017). "Lattice Boltzmann and Navier-stokes cartesian cfd approaches for airframe noise predictions." 23rd AIAA Computational fluid dynamics conference.
- Barber, J. R. (2018). *Contact mechanics*, Springer.
- Beinlich, L. (2021). Implementierung eines 3D Lattice-Boltzmann-Verfahrens zur numerischen Strömungssimulation im Rechteckkanal, Bachelorarbeit, Technische Universität Berlin.
- Benad, J. (2012). "On the dependence of the static friction force between a rigid, randomly rough fractal surface and a viscoelastic body on the normal force." *Physical Mesomechanics* 15(5): 300-302.
- Benad, J. (2014). "Luftfahrzeug." Patent application DE102014201040A1, Assignee: Airbus Operations GmbH.
- Benad, J. (2015). "The Flying V - A new aircraft configuration for commercial passenger transport." *Deutscher Luft- und Raumfahrtkongress*, Rostock, 2015.
- Benad, J. (2018). "Fast numerical implementation of the MDR transformations." *Facta Universitatis, Series: Mechanical Engineering* 16(2): 127-138.
- Benad, J., K. Nakano, V. L. Popov and M. Popov (2018). "Active control of friction by transverse oscillations." *Friction* 7(1): 74-85.
- Benad, J., M. Popov, K. Nakano and V. Popov (2018). "Stiff and soft active control of friction by vibrations and their energy efficiency." *Forschung im Ingenieurwesen* 82(4): 331-339.
- Bhatnagar, P. L., E. P. Gross and M. Krook (1954). "A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems." *Physical review* 94(3): 511.
- Burstedde, C., K. Klauck, A. Schadschneider and J. Zittartz (2001). "Simulation of pedestrian dynamics using a two-dimensional cellular automaton." *Physica A: Statistical Mechanics and its Applications* 295(3-4): 507-525.
- Cappuyns, T. (2019). *Handling Qualities of a Flying V Configuration*, Master thesis, Delft University of Technology.
- Chen, S. and G. D. Doolen (1998). "Lattice Boltzmann method for fluid flows." *Annual review of fluid mechanics* 30(1): 329-364.

- Chen, S., D. Martinez and R. Mei (1996). "On boundary conditions in lattice Boltzmann methods." *Physics of Fluids* 8(9): 2527-2536.
- Choochart, P. and C. Thipyopas (2020). "Study of passenger evacuation from the Airbus A330-300 aircraft." *Multidisciplinary Digital Publishing Institute Proceedings* 39(1): 25.
- Claeys, M. (2018). "Flying V and Reference Aircraft Structural Analysis and Mass Comparison."
- Cohen, D. and R. T. Jones (1960). *High speed wing theory*, Princeton University Press.
- Cornubert, R., D. d'Humières and D. Levermore (1991). "A Knudsen layer theory for lattice gases." *Physica D: Nonlinear Phenomena* 47(1-2): 241-259.
- Cummings, R. M., J. R. Forsythe, S. A. Morton and K. D. Squires (2003). "Computational challenges in high angle of attack flow prediction." *Progress in Aerospace Sciences* 39(5): 369-384.
- Da Silva, A. (2008). "Numerical studies of aeroacoustic aspects of wind instruments."
- Davison, M. (2021). Investigation of material and porosity effects during labyrinth seal rubs in aero engines, Master thesis, Technische Universität Berlin.
- Diercks, P. (2018). Ermüdungsfestigkeit von Tannenbaumverbindungen korrodierter Turbinenscheiben. Department of System Dynamics and Friction Physics, Masterarbeit, Technische Universität Berlin.
- Dimaki, A., A. Dmitriev, Y. Chai and V. Popov (2014). "Rapid simulation procedure for fretting wear on the basis of the method of dimensionality reduction." *International Journal of Solids and Structures* 51(25-26): 4215-4220.
- Dimaki, A., A. I. Dmitriev, N. Menga, A. Papangelo, M. Ciavarella and V. L. Popov (2016). "Fast high-resolution simulation of the gross slip wear of axially symmetric contacts." *Tribology Transactions* 59(1): 189-194.
- DLR (2021). "Ludwig Prandtl with his fluid test channel." from https://www.dlr.de/content/en/images/2017/4/ludwig-prandtl-with-his-fluid-test-channel_29031.html.
- Erickson, G. E. (1995). "High angle-of-attack aerodynamics." *Annual review of fluid mechanics* 27(1): 45-88.
- European Union Aviation Safety Agency (2020). "Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes CS-25."
- Faggiano, F., R. Vos, M. Baan and R. Van Dijk (2017). "Aerodynamic design of a Flying V aircraft." 17th AIAA Aviation Technology, Integration, and Operations Conference, 2017.
- Falcucci, G., M. Aureli, S. Ubertini and M. Porfiri (2011). "Transverse harmonic oscillations of laminae in viscous fluids: a lattice Boltzmann study." *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 369(1945): 2456-2466.
- Fares, E., M. Wessels, R. Zhang, C. Sun, N. Gopalaswamy, P. Roberts, J. Hoch and H. Chen (2014). "Validation of a lattice-Boltzmann approach for transonic and supersonic flow simulations." 52nd Aerospace Sciences Meeting.
- Ferziger, J. H., M. Perić and R. L. Street (2002). *Computational methods for fluid dynamics*, Springer.
- Frisch, U., B. Hasslacher and Y. Pomeau (1986). "Lattice-gas automata for the Navier-Stokes equation." *Physical review letters* 56(14): 1505.

- Galea, E., S. Blake and P. Lawrence (2001). "The airEXODUS evacuation model and its application to aircraft safety", CMS Press.
- Gebauer, J. and J. Benad (2021). "Flying V and Reference Aircraft Evacuation Simulation and Comparison." arXiv preprint arXiv:2102.06502.
- Ginzbourg, I. and P. Adler (1994). "Boundary flow condition analysis for the three-dimensional lattice Boltzmann model." *Journal de Physique II* 4(2): 191-214.
- Glauert, H. (1933). *Wind tunnel interference on wings, bodies and airscrews*, Aeronautical Research Council London (UK).
- Goldthorpe, S., K. Rossitto, D. Hyde and K. Krothapalli (2010). "X-48b blended wing body flight test performance of maximum sideslip and high to post stall angle-of-attack command tracking." *AIAA atmospheric flight mechanics conference*.
- Guo, Z. and C. Shu (2013). *Lattice Boltzmann method and its application in engineering*, World Scientific.
- Hansen, E. W. and P.-L. Law (1985). "Recursive methods for computing the Abel transform and its inverse." *JOSA A* 2(4): 510-520.
- He, X. and L.-S. Luo (1997). "Theory of the lattice Boltzmann method: From the Boltzmann equation to the lattice Boltzmann equation." *Physical review E* 56(6): 6811.
- Helbing, D. and P. Molnar (1995). "Social force model for pedestrian dynamics." *Physical review E* 51(5): 4282.
- Hellmann, R. (2020). *Evacuation simulation of a Flying V aircraft using Cellular Automata*, Bachelor thesis, Technische Universität Berlin.
- Herwig, D. and H. Rode (2002). *Geheimprojekte der Luftwaffe, Bd.2, Strategische Bomber 1935-1945*.
- Isgro, F. (2020). "Fuselage Design Studies to Improve Boarding Performance of Novel Passenger Aircraft." Master thesis, supervisor: La Rocca G, Delft University of Technology.
- Kang, Q., D. Zhang and S. Chen (2002). "Unified lattice Boltzmann method for flow in multiscale porous media." *Physical review E* 66(5): 056307.
- Kramer, M. (1932). "Increase in the maximum lift of an airplane wing due to a sudden increase in its effective angle of attack resulting from a gust."
- Krüger, T., H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva and E. M. Viggien (2017). "The lattice Boltzmann method." Springer International Publishing 10(978-3).
- Ladd, A. and R. Verberg (2001). "Lattice-Boltzmann simulations of particle-fluid suspensions." *Journal of statistical physics* 104(5): 1191-1251.
- Ladd, A. J. (1994). "Numerical simulations of particulate suspensions via a discretized Boltzmann equation. Part 1. Theoretical foundation." *Journal of fluid mechanics* 271: 285-309.
- Landau, L. and E. Lifschitz (1971). *Lehrbuch der Theoretischen Physik IV: Hydrodynamik*, Akademie Verlag: Berlin.
- Li, Q., F. Forsbach, M. Schuster, D. Pielsticker and V. L. Popov (2018). "Wear analysis of a heterogeneous annular cylinder." *Lubricants* 6(1): 28.

- Li, Y., R. Shock, R. Zhang and H. Chen (2004). "Numerical study of flow past an impulsively started cylinder by the lattice-Boltzmann method." *Journal of fluid mechanics* 519: 273-300.
- Liebeck, R., M. Page and B. Rawdon (1998). "Blended-wing-body subsonic commercial transport." 36th AIAA aerospace sciences meeting and exhibit.
- Liebeck, R. H. (2004). "Design of the blended wing body subsonic transport." *Journal of Aircraft* 41(1): 10-25.
- Martinez-Val, R. (2007). "Flying wings. A new paradigm for civil aviation?" *Acta Polytechnica* 47(1).
- Mohamad, A. (2011). *Lattice Boltzmann Method*, Springer.
- Müller, A. (2021). *Numerische Simulation und Analyse der zweidimensionalen Umströmung eines ruhenden Zylinders mit der Gitter-Boltzmann-Methode*, Bachelorarbeit, Technische Universität Berlin.
- Murio, D. A., D. Hinestroza and C. E. Mejía (1992). "New stable numerical inversion of Abel's integral equation." *Computers & Mathematics with Applications* 23(11): 3-11.
- mvAero (2019). <https://mvaero.nl/>.
- Oberleithner, K., M. Sieber, C. Nayeri, C. O. Paschereit, C. Petz, H.-C. Hege, B. R. Noack and I. Wygnanski (2011). "Three-dimensional coherent structures in a swirling jet undergoing vortex breakdown: stability analysis and empirical mode construction." *Journal of fluid mechanics* 679: 383-414.
- Oosterom, W. (2021). *Flying-V Family Design*, Master thesis, Delft University of Technology.
- Our World in Data (2020). "Supercomputer Power (FLOPS), 1993 to 2020." from <https://ourworldindata.org/grapher/supercomputer-power-flops>.
- Palermo, M. and R. Vos (2020). "Experimental aerodynamic analysis of a 4.6%-scale Flying-V subsonic transport." *AIAA Scitech 2020 Forum*.
- Pohrt, R. (2020). "Friction influenced by vibrations: A refined contact-mechanics view on lateral and rotational oscillations."
- Pohrt, R. and Q. Li (2014). "Complete boundary element formulation for normal and tangential contact problems." *Physical Mesomechanics* 17(4): 334-340.
- Popov, M. and J. Benad (2013). "Numerische Untersuchung von Emission in einem Rollkontakt mit rauen Oberflächen." *Reibung, Schmierung und Verschleiß. Forschung und praktische Anwendungen, Tribologie-Fachtagung*, 54: 1/1-1/6.
- Popov, M., V. L. Popov and N. V. Popov (2017). "Reduction of friction by normal oscillations. I. Influence of contact stiffness." *Friction* 5(1): 45-55.
- Popov, V. L. and S. G. Psakhie (2007). "Numerical simulation methods in tribology." *Tribology International* 40(6): 916-923.
- Prandtl, L. (1905). "Über Flüssigkeitsbewegung bei sehr kleiner Reibung." *Sonderdruck aus den „Verhandlungen des III. Internationalen Mathematiker-Kongresses, Heidelberg 1904“*. Druck und Verlag von B. G. Teubner, Leipzig: 484-491.
- Qin, N., A. Vavalle, A. Le Moigne, M. Laban, K. Hackett and P. Weinerfelt (2004). "Aerodynamic considerations of blended wing body aircraft." *Progress in Aerospace Sciences* 40(6): 321-343.

- Raymer, D. P. (2012). "Aircraft design: a conceptual approach (AIAA Education Series)." Reston, Virginia.
- Rubio Pascual, B. and R. Vos (2020). "The effect of engine location on the aerodynamic efficiency of a Flying-V aircraft." AIAA Scitech 2020 Forum.
- Ruiz Garcia, A., R. Vos and C. de Visser (2020). "Aerodynamic model identification of the Flying V from wind tunnel data." AIAA Aviation 2020 Forum.
- Rupp, K. (2016). "FLOPs per Cycle for CPUs, GPUs and Xeon Phi." from <https://www.karlrupp.net/2016/08/flops-per-cycle-for-cpus-gpus-and-xeon-phi/>.
- Rupp, K. (2017). "42 Years of Microprocessor Trend Data." from <https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/>.
- Schade, H. and E. Kunz (2007). Strömungslehre, Walter de Gruyter.
- Schaller, R. R. (1997). "Moore's law: past, present and future." IEEE spectrum 34(6): 52-59.
- Schlichting, H. (1933). "Zur entstehung der turbulenz bei der plattenströmung." Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse 1933: 181-208.
- Schlichting, H. and K. Gersten (2006). Grenzschicht-theorie, Springer-Verlag.
- Schlichting, H. and E. A. Truckenbrodt (1967). Aerodynamik des Flugzeuges: Erster Band Grundlagen aus der Strömungsmechanik Aerodynamik des Tragflügels (Teil I), Springer-Verlag.
- Spedding, G. and J. McArthur (2010). "Span efficiencies of wings at low Reynolds numbers." Journal of Aircraft 47(1): 120-128.
- Storck, R. (2003). Flying Wings-Die historische Entwicklung der Nurflügelflugzeuge der Welt. Bonn: Bernard & Graefe Verlag.
- Succi, S. (2001). The lattice Boltzmann equation: for fluid dynamics and beyond, Oxford university press.
- Sun, Y., N. B. Agostini, S. Dong and D. Kaeli (2019). "Summarizing CPU and GPU design trends with product data." arXiv preprint arXiv:1911.11313.
- Taira, K. and T. Colonius (2009). "Three-dimensional flows around low-aspect-ratio flat-plate wings at low Reynolds numbers." Journal of fluid mechanics 623: 187-207.
- Ten Cate, A., C. Nieuwstadt, J. Derksen and H. Van den Akker (2002). "Particle imaging velocimetry experiments and lattice-Boltzmann simulations on a single sphere settling under gravity." Physics of Fluids 14(11): 4012-4025.
- Tollmien, W. (1931). "Grenzschichttheorie." Handbuch Experimentalphysik 4: 241-287.
- Torenbeek, E. (1982). Synthesis of subsonic airplane design: an introduction to the preliminary design of subsonic general aviation and transport aircraft, with emphasis on layout, aerodynamic design, propulsion and performance, Delft University Press, Kluwer Academic Publishers.
- Torenbeek, E. (2013). Advanced Aircraft Design: Conceptual Design, Analysis and Optimization of Subsonic Civil Airplanes, John Wiley and Sons, Ltd.

- Van Empelen, S. and R. Vos (2021). "Effect of Engine Integration on a 4.6%-Scale Flying-V Subsonic Transport." AIAA Scitech 2021 Forum.
- Vicroy, D. (2009). "Blended-wing-body low-speed flight dynamics: summary of ground tests and sample results." 47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition.
- Viet, R. (2019). "Analysis of the flight characteristics of a highly swept cranked flying wing by means of an experimental test."
- Vink, P., T. Rotte, S. Anjani, C. Percuoco and R. Vos (2020). "Towards a hybrid comfortable passenger cabin interior for the Flying V aircraft." International Journal of Aviation, Aeronautics, and Aerospace 7(1): 1.
- Wakayama, S. and I. Kroo (1998). "The challenge and promise of blended-wing-body optimization." 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization.
- Weidmann, U. (1993). "Transporttechnik der Fußgänger: Transporttechnische Eigenschaften des Fußgängerverkehrs, Literaturlauswertung." IVT Schriftenreihe 90.
- Willert, E. (2020). "Die Methode der Dimensionsreduktion in der Kontaktmechanik." Stoßprobleme in Physik, Technik und Medizin, Springer: 95-111.
- Willert, E. (2021). "FFT-based Implementation of the MDR transformations for homogeneous and power-law graded materials." Facta Universitatis, Series: Mechanical Engineering.
- Wisnoe, W., R. E. M. Nasir, W. Kuntjoro and A. M. I. Mamat (2009). "Wind tunnel experiments and CFD analysis of Blended Wing Body (BWB) Unmanned Aerial Vehicle (UAV) at mach 0.1 and mach 0.3. International Conference on Aerospace Sciences and Aviation Technology", The Military Technical College.
- Ziegler, D. P. (1993). "Boundary conditions for lattice Boltzmann simulations." Journal of statistical physics 71(5): 1171-1177.