UNIVERSITY OF TWENTE.

Faculty of Engineering Technology

Instituto Tecnológico de Aeronáutica

Field of Aeronautical Projects, Structures and Aerospace Systems

A simple semi-strong solution for the integral boundary layer equations using high-order Galerkin methods

UT Graduation Number 449

Master of Science Thesis

Mechanical Engineering

Aeronautical and Mechanical Engineering

Gerben Johannes Siegersma

s2017016

g.j.siegersma@student.utwente.nl

21 August 2024

Department:
Engineering Fluid Dynamics
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

A thesis submitted to

University of Twente
Department of Engineering Fluid Dynamics

Instituto Tecnológico de Aeronáutica
Field of Aeronautical Projects, Structures and Aerospace Systems, EAM-1

In partial fulfillment of the requirements for the degrees of
Mechanical Engineering
and
Aeronautical and Mechanical Engineering

by

Gerben J. Siegersma

Certified by

Prof. Rodrigo Costa Moura	Supervisor
Dr. Ir. Hüseyin Özdemir	Supervisor
Prof. Dr. Ir. Cornelius H. Venner	Chairman
Prof. André F. de Castro da Silva	Reader
Dr. Ir. Richard J.A.M. Stevens	Reader

Foreword and acknowledgments

Brazil has been good to me. It has brought me experiences and memories that I will carry with me for the rest of my life, and cherish for good. Apart from the delicious food, the interesting and at times weird and inconsistent language, and the cities, monuments and nature of epic and beautiful proportions that the country has to offer, Brazil made me grow as a person too. I have encountered a new-found discipline and motivation, that I didn't know I had in me; it is a great feeling to disprove myself in the moments that I thought I would not be able to achieve what I have achieved until now. Above all, I am grateful, and I still find myself thinking at times, 'wow, I'm actually in Brazil'.

Rodrigo, eu me diverti bastante tendo você como meu orientador. Lembro-me de discussões que adiantaram o trabalho, e aprecio seus comentários espertos e apoio quando eu precisava. Além disso, de ser um bom orientador e além de melhorar o meu português, falando de assuntos como aeronáutica e religião, além de tudo isso, as figurinhas apropriadas que você sempre teve, para qualquer situação, puseram um sorriso na minha cara várias vezes.

Hüseyin, I would like to thank you for insights and thoughts that I would have never come up with on my own. The times that we were stuck in the process or restrained by our own tunnel vision, you were there for us to provide a piece of crucial information, thereby freeing us from the boulder blocking our academic path. Thank you for your supervision, even on a Friday afternoon when you're supposed to relax, celebrate the weekend or do some house chores.

Prof. Dr. Ir. Venner, Kees, ik heb het aan jou te danken dat ik de mogelijkheid heb een machtig mooi avontuur te beleven hier op een ander continent. Je diende al als voorzitter in mijn bachelorcommissie, en nu heb ik nogmaals het plezier je als voorzitter van mijn afstudeercommissie te hebben, deze keer als Braziliëtelg, aspirerend MSc. Dank voor deze kans!

Estendo minha gratidão ao Prof. André Fernando, para servir como presidente da banca e leitor da dissertação. Lembro-me de ter chegado no LASCA um dia, molhado até os ossos. Não esqueci do seu comentário de curtir o clima brasileiro 'úmido'.

Dr. Ir. Stevens, Richard, ook jij zat reeds in mijn afstudeercommissie drie jaar geleden. Dank voor het nogmaals zitting willen nemen in de commissie als extern lid. Ik kijk uit naar de grondige discussie die we zullen hebben over dit werk.

David, huisgenoot, kantoorgenoot en reisgenoot, dank voor alle wekdiensten, het goeie eten, de onvergetelijke tripjes naar de zee, de heuvels, steden en eilanden. Ik had niet echt verwachtingen toen we hier samen kwamen, maar als ik ze had gehad, had je ze wel allemaal overtroffen; ik had me geen beter maatje kunnen voorstellen met wie ik dit had kunnen delen.

Daniel, Eduardo, thanks for the times we laughed together. You and your pretty faces made getting up in the morning to drag my behind to the office slightly more bearable. Big hug for the Ana Paulas! When's the next rodízio?!

Heit, mem, Paul en Klaske, Douwe en Lotte, Tynke en Jules, Christien! Dankewol foar de stipe en it ivige geduld dat jimme hawwe moatte mei my as jongste neefke, broerke en bern. It war heech tiid dat de hikkeslúter ek einlings it hik mar ris slute soe. Kreas, no?

Ó amor, viu só? Está orgulhosa de mim? Espero que esteja. Foi com seu apoio que consegui. Você fez este ano maravilhoso para mim. Eu te agradeço por isso, querida. Espero que tenha sido igualmente maravilhoso para você. Comemoremos!

Abstract

In this study, laminar and turbulent boundary layers over curved surfaces and airfoils are solved while using simple modeling approaches coupled with spectral element methods.

A linear vortex panel method has been designed that can obtain reliable pressure distribution results in an inviscid setting. This pressure distribution is used as input for the integral boundary layer equations, which are spatially discretized using higher-order Galerkin method. A simple interaction law between inviscid region and boundary layer model is implemented as a semi-strong solution that predicts the changes that the displacement thickness exerts on the edge velocity. The semi-strong solution enables circumvention of Goldstein's singularity of separated flow.

The equations are solved using an implicit Euler based point-implicit scheme, and spectral vanishing viscosity is applied to stabilize the numerics. The resulting boundary layer parameters are used as input for the panel method, such that an iterative scheme is obtained that proceeds until convergence has been reached.

A wake is added to ensure smooth outflow at the trailing edge, such that the Kutta condition is fully satisfied. Validation of the boundary layer parameters as well as the pressure distributions is done using XFOIL and experimental results.

Contents

Chapter 1

Introduction

1.1 Boundary layer theory

For past decades, boundary layers have been the central motivation and topic of many articles, studies and additional research done over the entire world, and as such, it remains an important subject in the field of aeronautics and aerodynamics. Boundary layers, a concept illustrated in Figure 1.1, encapsulate complex physical processes such as turbulence, transition and heat transfer, resulting from the viscous effects in a fluid. Therefore, a good understanding of the physics is paramount for any practical application, which can range from wind turbines to hydrodynamics, as well as arterial blood flow, and many more. Boundary layer research has been conducted since the beginning of the previous century by Prandtl [1], who is credited as the discoverer of the boundary layer. Von Kármán built further on Prandtl's work by beginning with the boundary layer equations in integral form and arriving at the momentum integral equation [2], which bears Von Kármán's name. Important contributions in boundary layer theory have since been done by Schlichting [3], Cebeci [4], and Drela [5], to name a few. An extensive description of boundary layer research, methods and solutions can be found in Özdemir [6].

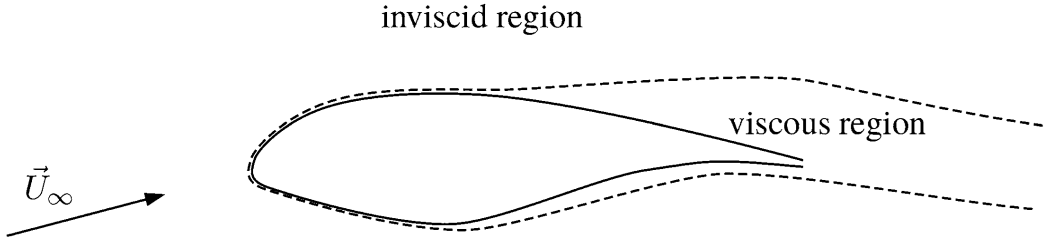

Figure 1.1: Boundary layer on an airfoil visualized, from Özdemir [6].

Another important contributor was Goldstein, who discovered a phenomenon currently known as the Goldstein singularity [7], where a simulated flow breaks down due to a singularity in the solution of the integral boundary layer equations. The physical explanation behind this simulation breakdown is that the flow becomes prone to separation, and the boundary layer calculation at that moment cannot be continued anymore. Some solutions have been proposed, such as the triple-layered boundary layer structure, found independently by Stewartson [8], Messiter [9], and Sychev [10], where the underlying assumption is that flow separation takes place in a boundary sublayer, and each so-called deck is solved with a different set of similarity equations. What is

found, is that a loss of hierarchy between the inviscid flow region and the boundary layer ensues, and thus the simulation breaks down. In other words, in a 'healthy' system, the inviscid region feeds a pressure gradient to the boundary layer. However, when a flow is liable to separate, a self-induced pressure gradient in the boundary layer is created, thus there is no more hierarchy between the two models at the point of separation and beyond. So begins the need for models that have a strong coupling between inviscid region and boundary layer solver, to bypass the singularity.

The simplest approach is the design of a panel method, that directly supplies its pressure distribution as input to the viscous model. The viscous model then calculates and outputs a displacement thickness, which is used to create a new effective aerodynamic body around the airfoil, for which subsequently, a new pressure distribution is calculated by the panel method. This method is applied iteratively and converges in this way, but as mentioned, the solution breaks down when separation occurs, due to Goldstein's singularity [7]. The most robust alternative is a strongly coupled system, where the pressure distribution is taken within the viscous model, and everything is solved at once. An example of such a strongly coupled (two-dimensional) system is *XFOIL*, by Mark Drela [11]. Starting development in the 80s, *XFOIL* has nowadays become a widely-used code to obtain reasonably accurate predictions of the aerodynamic properties of airfoils. Having been updated over the years, it has become a powerful tool to rapidly perform calculations on two-dimensional airfoils and geometries, with the possibility of inviscid as well as viscous calculations. For many applications, it serves as a useful means of validation for any generated aerodynamic result, and as such, it will be used in the present study.

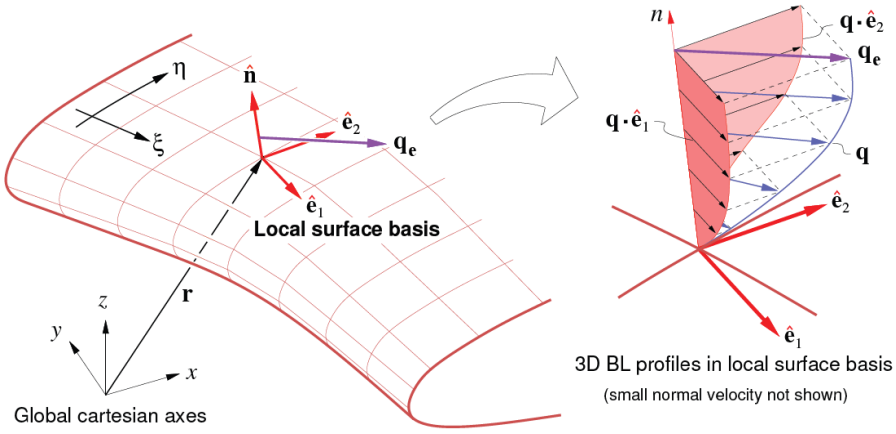

Figure 1.2: 3D wing geometry with local Cartesian vectors and three-dimensional boundary layer profile, from [12].

However, the strongly coupled method, albeit very robust, is significantly more complicated than the direct, iterative solver, and very unfeasible to extend to complex geometries in three-dimensional space. Drela published a study on a three-dimensional boundary layer model, [13], expanded as recently as 2022 by Zhang [12]. An example of the mesh needed for such a model can be seen in Figure 1.2. Including in this three-dimensional model the external pressure (or velocity) calculations would make the model even more complex, and extremely costly in computational terms. As these complete, strongly coupled models are unfeasible, computationally costly, and complicated to implement, and direct solvers break down in separated flows as a

result of Goldstein's singularity, there is a desire for a different approach that could be feasible in three dimensions as well. The idea of the interaction law is brought to life, in which a reaction from the boundary layer on the outer potential flow is incorporated in the viscous model. It can be seen as a semi-strong approach, as it couples the pressure distribution to the boundary layer in a weak form, and is not directly included in the system as opposed to in a strongly coupled method. Figure 1.3 shows in a schematic way how the direct method and the semi-strong scheme operate. In the left block, the models are fully separated. In the right, changes in the displacement thickness immediately induce a change in the external velocity, such that an 'updated' velocity is used in the boundary layer calculations.

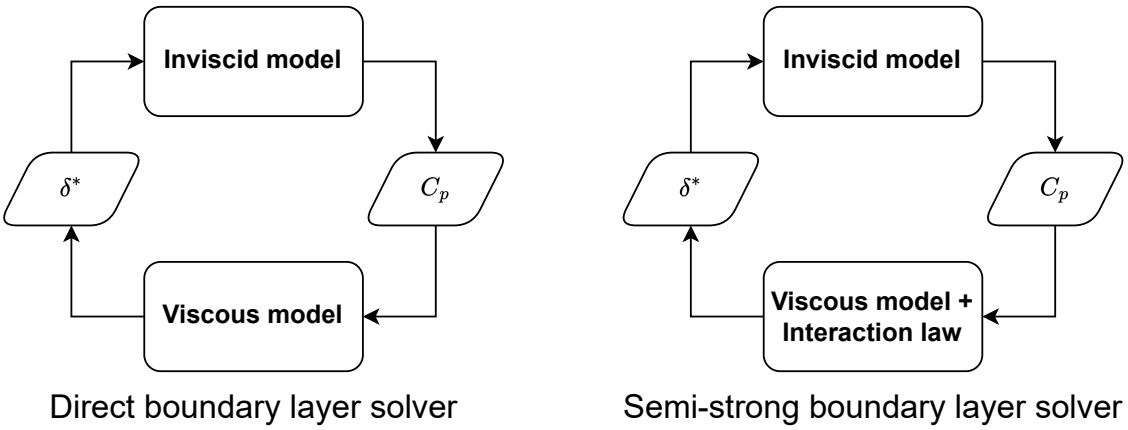

Figure 1.3: Block diagram showing different interaction methods.

In 1979, Veldman produces an early example of such a semi-strong scheme [14]. Over the years, Veldman develops new methods, including quasi-simultaneous schemes and transonic cases [15][16][17], until in his 2009 paper, he proposes a simple, semi-strong approach to solving the boundary layer. He names it a quasi-simultaneous viscous-inviscid interaction law [18]. The Von Kármán momentum integral equation is posed as the first in the system of equations. Head's entrainment method, which describes the mixing of fluids in a turbulent flow [19], is taken as the second equation, closing the system of equations with a so-called closure set. The external velocity is obtained from an inviscid panel method, and the change that the boundary layer induces on the external velocity is posed as an additional unknown in the system of equations, in the form of a simple interaction law. In this manner, the hierarchy between the models is kept even in separated flow. Veldman's article shows promising results in comparison to experiments, and in addition demonstrates the robustness of the semi-strong boundary layer solver.

1.2 Spectral element methods

Many different techniques have risen over the past decades to better approximate numerically the phenomena surrounding mankind. An example of such a technique, that is still widely used, is the finite difference method. Simple, robust and effective, it was already discovered and explained by Euler in his *Institutiones calculi differentialis* [20]. Depending on the mesh size, very reasonable results are obtained with a finite difference numerical scheme. Another technique is the finite element method, that relies on dividing a domain in a number of elements, usually two-

or three-dimensional. The convergence of the method depends on the set of basis functions used, often chosen triangular or quadrilateral. Then there are spectral methods, where the solution to a differential equation is globally approximated through the summation of a set of basis functions. Coefficients of the basis functions are found through weighted-residual projection to match the solution as well as possible. Spectral element methods combine the advantages of the latter two methods by dividing a domain into separate elements, where each element has its own high-order solution approximation. In numerical jargon, it takes the flexibility offered by h-type finite element methods to refine the mesh if one desires, and in addition, it allows the increase of the order of piece-wise polynomial approximation of a local element, or p-refinements. As such, spectral element methods are also called hp element methods.

The first use of a spectral element method is attributed to Patera [21], who in 1984 published a work, where he tested the method on an expanding pipe, or channel, to demonstrate the viability. The method proved itself promising, and was developed more in 1989, by Maday and Patera [22], who showed the spatial discretization of the incompressible Navier-Stokes equations. It then found its use in the field of seismology, where Chebyshev nodes instead of linearly distributed nodes were introduced by Faccioli et al. [23] to avoid Runge's phenomenon of oscillation occurring at the element's boundary through polynomial interpolation. Moreover, Legendre Gauss-Lobatto quadrature was used to approximate numerically the integrals when the governing partial differential equation is transformed to its weak formulation. A little while later, Lagrange polynomials were employed, once again in seismological context, as a set of basis functions by Komatitsch and Vilotte [24], resulting in a diagonal mass matrix and thus simplifying the subsequent matrix inversion. Karniadakis and Sherwin published a book detailing the spectral/hp element methods in the context of Computational Fluid Dynamics [25], explaining, among many things, the technique of collocated Gauss-Lobatto quadrature using a Lagrange basis defined at the Lobatto nodes. When using a collocated quadrature, the basis becomes approximately orthogonal, as it leads to a lumped diagonal mass matrix, and as such, the computational cost becomes significantly lower.

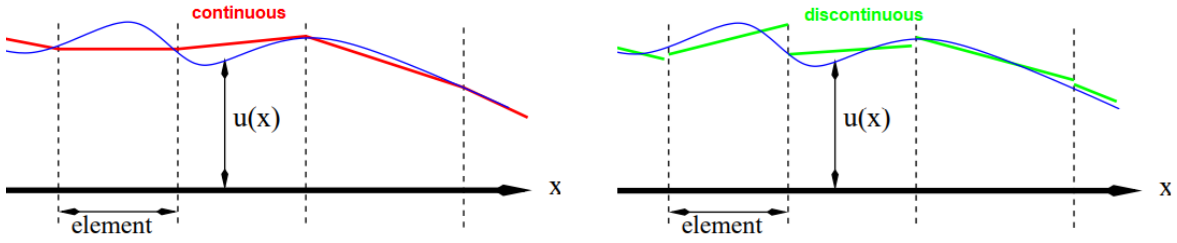

Figure 1.4: Visualization of CG and DG solutions obtained from a SEM solver, from Moura [26].

A principal distinction that can be made in spectral element methods is between continuous and discontinuous methods. An example of the former is Continuous Galerkin, or CG, whereas an example of the latter is Discontinuous Galerkin, or DG. In CG, continuity of the zeroth degree is enforced between elements, guaranteeing that the numerical solution (but not necessarily its derivative) be continuous on element interfaces. In DG, however, this condition is not imposed, allowing for small discontinuities across element interfaces and yielding a larger independence between the elements. This difference in continuity among elements can be seen in Figure 1.4. In addition, numerical dissipation is introduced through upwind fluxes on the interfaces. The

development of spectral element methods brought about some challenges, such as the choice between continuous or discontinuous methods, how to choose the configuration of mesh h and order p refinement for the most accurate result, or ensuring stability in the case of shocks where numerical dissipation does not suffice. In recent decades, all challenges and difficulties have been overcome, and currently, spectral element methods are successfully used in a variety of problems and applications [27]. In the present study, a higher-order Continuous Galerkin method is employed as discretization method, in a nodal-collocated formulation, which leads to a diagonal mass matrix.

1.3 SEM in boundary layer analysis

Some works from the past have already shown very reasonable results from a Discontinuous Galerkin scheme applied as spatial discretization to the unsteady version of the integral boundary layer equations, namely the master's theses of Van Es [28], Van den Boogaard [29] and Haciahmetoğlu [30]. In the quasi-simultaneous and fully simultaneous schemes, there are non-conservative effects present that change the location of the equilibria. Due to these changes, effectively the solution of the scheme is changed and a dependency on the numerical scheme used is created. Specialized non-conservative numerical schemes based on Discontinuous Galerkin have been applied to circumvent this dependence on numerical scheme in the master's theses by Seubers [31] and Passalacqua [32]. A joint effort to produce an unsteady two-dimensional interacting boundary layer method that also deals with the arising non-conservative mechanisms was performed by Özdemir et al. [33]. Another work that is worth mentioning, is a master's thesis by Ye [34], that also uses Discontinuous Galerkin as spatial discretization scheme and found improvements in the set of closure relations. Lastly, an article by Zhang et al. [35] shows a coupled solver based on Discontinuous Galerkin, designed for free transition flow.

A recent study from 2020 by Moura et al. [36] finds preliminary results with a Continuous Galerkin spatial discretization, applied to a non-trivial, complex boundary layer. The study finds reasonable results for a lower amount of degrees of freedom, demonstrating that Continuous Galerkin could have potential to reach accuracy at a lower computational cost. Aside from Continuous Galerkin having been used in high-fidelity simulations, its efficiency in simpler simulations, that do not necessarily require this high level of accuracy, e.g. the integral boundary layer equations, remains to be demonstrated. The present study serves as a first exploratory journey of the implementation of the Continuous Galerkin in such a manner. The aim is to discover the advantages and disadvantages that the higher-order Galerkin spectral element method offers when used as a spatial discretization scheme for the integral boundary layer equations.

1.4 The objective

Considering that spectral element methods are a relatively new technique of spatial discretization, it is interesting to see what possibilities it could offer when applied to a boundary layer. Results based on Discontinuous Galerkin have already been shown to be promising, but studies documenting on the performance of Continuous Galerkin are scarcer. This calls for research

that might uncover some of higher-order Galerkin spectral element methods' hidden qualities in facilitating an accurately discretized interactive boundary layer model.

The posed objective is to develop a semi-strong solver that comprises a viscous boundary layer modeled by a higher-order Continuous Galerkin approximation, combined with the simple interaction law to provide simultaneity with the external velocity model and circumvent Goldstein's singularity. The boundary layer model will receive the external inviscid velocity as input, obtained from a vortex panel method with linearly varying circulations across the panels. The viscous solver in turn will generate, from the same set of equations that Veldman uses [18], the boundary layer parameters that are used as new input in the panel method, until iteratively converging to the steady-state solution. Within the viscous solver, the interaction law is employed, such that the external velocity change as a result from the change in displacement thickness is predicted by the simple model, as mentioned in Veldman's article, yielding the semi-strong solution. Spectral vanishing viscosity is added as a means to stabilize the higher-order Galerkin scheme, which is otherwise free of numerical dissipation. The equations are solved using a point-implicit convergence scheme. NACA airfoils are taken as the boundary layer-generating surfaces, and can easily be compared to results obtained from *XFOIL*. Lastly, a simplified wake model is added to guarantee a smooth outflow at the trailing edge.

1.4.1 The outline

The present study will present the following topics:

- Chapter 2 will discuss the theory of potential flow and panel methods, that rely on this theory. The derivation of the potential function, which consists of linearly varying vortices superimposed on a uniform flow, is given, and the linear vortex panel method is presented. The Kutta condition is explained and results from NACA airfoils are compared with corresponding results from *XFOIL*. Moreover, a mesh refinement of the panel method is shown and the necessity of a wake is illustrated.

- Chapter 3 discusses boundary layer theory and shows the system of equations that will be solved in the present study, namely the Von Kármán momentum integral equation and Head's turbulent entrainment equation. For the laminar part, Thwaites' method is introduced and applied. Goldstein's singularity is demonstrated and the interaction law to circumvent it is explained. Lastly, a derivation of the analytical model is given, which represents the interaction law implemented in the boundary layer solver.

- Chapter 4 gives a basic explanation of Lagrange polynomials, the theory of Galerkin methods applied to the advection-diffusion equation, and shows a result for this equation as a test case. It proceeds to apply a higher-order Galerkin discretization on the governing set of equations, as well as on the interaction law. It discusses the point-implicit scheme that is used to converge the equations, and shows the theory behind spectral viscosity that is added to the system for stabilization measures. Lastly, an order refinement of the Galerkin scheme is shown.

- Chapter 5 shows and discusses the results from the boundary layer model applied to the NACA 0012 symmetrical airfoil and the NACA 2412 cambered airfoil, at various degrees

of angles of attack. Displacement thickness, momentum thickness and shape factor will be visualized, as well as the viscous pressure distributions. The aerodynamic lift and drag coefficients are calculated and shown. Comparisons to *XFOIL* and experimental results will be made in this chapter, and general comments on the results are given.

- Chapter 6 will draw conclusions on the presented work and give future recommendations.

Chapter 2

Potential Flow and Panel Methods

In this chapter, the panel method will be discussed. The theory of potential flow, on which the panel method relies, is treated briefly, and the potential function of an airfoil with linearly varying vortices in a uniform flow is shown and derived. It is explained how the Kutta condition is satisfied, and a mesh refinement of the panel method is performed by increasing the amount of panels. The manner in which the panel method is coupled with the boundary layer solver is explained, and the necessity of a wake is illustrated.

2.1 Potential theory

A flow surpassing an object can be separated into two different regions: one inviscid outer layer, and a boundary layer sufficiently close to the aforementioned object, where viscous effects need to be considered. For incompressible flow, it is established that the velocity is free of divergence,

$$\nabla \cdot \boldsymbol{V} = 0. \tag{2.1.1}$$

Potential theory relies heavily on the assumption that the velocity in the inviscid layer is irrational, and thus, it can be defined as having zero curl,

$$\nabla \times \boldsymbol{V} = 0. \tag{2.1.2}$$

From vector calculus, the velocity vector field can then conveniently be written as the gradient of a scalar potential,

$$\boldsymbol{V} = \nabla\phi \longrightarrow \nabla \times \nabla\phi = 0. \tag{2.1.3}$$

Now, substituting Equation 2.1.3 in Equation 2.1.1, an alternative formulation is obtained, namely the differential equation that forms the basis for potential flow:

$$\nabla \cdot \nabla\phi = \nabla^2\phi = 0. \tag{2.1.4}$$

The physical significance of writing the velocity as the gradient of a scalar potential, is the property of superposition, due to their linear nature: *let A and B be solutions of an arbitrary linear problem, then $A + B$ is also a solution of said problem.* What this means, in the case

of a fluid, is that a potential flow can be described as the summation of multiple (types of) elementary solutions.

The fundamental types of solutions are uniform flow, a source and sink, and a vortex. The latter is an indication of fluid particles moving in a circular trajectory, rather than revolving around their own centers, thereby maintaining the irrotational nature of the flow. The elementary solutions can be used to virtually approximate any shape or body, the illustration of which can be seen in Figure 2.1. A symmetric airfoil, also known as a teardrop profile, can be approximated well with sources and sinks, due to its symmetry. For a cambered airfoil, additionally, vortices are needed. The profile in (a) of Figure 2.1, can be formed by superposition of (b), the teardrop, and (c), the skeleton. This skeleton represents the camber line of the body being approximated, and contains a vortex distribution to form the characteristic curve of an airfoil profile. Skeleton theory will be discussed in more detail in section 3.6.

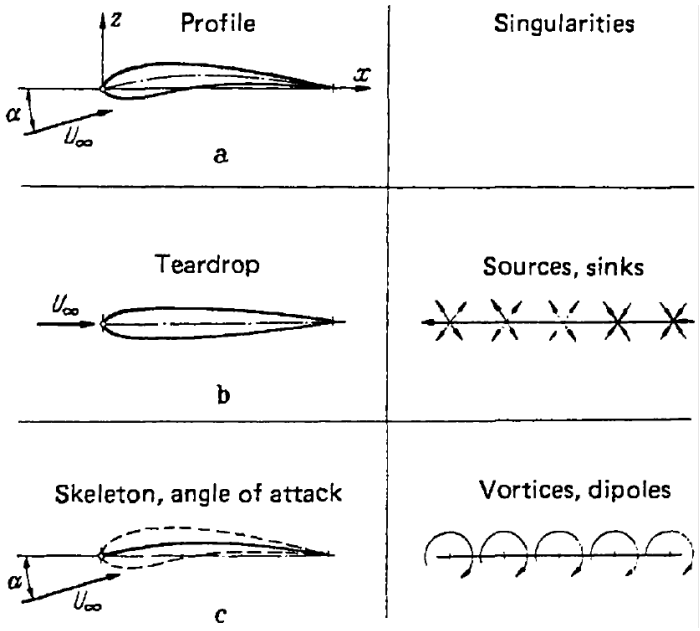

Figure 2.1: (a) Cambered airfoil at angle of alpha α, (b) Symmetric airfoil generated by sources and sinks, (c) Cambered airfoil at angle of attack α, generated by vortices, image taken from Schlichting [37].

It is decided to follow the well-established panel method described in the book by Kuethe and Chow [38], known as the linear vortex panel method, where a curved body, e.g. a cambered airfoil, is divided into m panels, with vortices of unknown circulation strength defined at every panel edge, varying linearly from endpoint to endpoint, such that $m+1$ circulations γ are introduced. Superimposing all these circulations and a uniform flow V_∞, the potential is then defined as

$$\phi(x,y) = V_\infty(x\cos\alpha + y\sin\alpha) - \sum_{j=1}^{m}\int_j \frac{\gamma(s_j)}{2\pi}\tan^{-1}\left(\frac{y-y_j}{x-x_j}\right)\mathrm{d}s_j, \tag{2.1.5}$$

where x and y can be taken as arbitrary control points, on or off the surface of the airfoil, and α is the angle of attack. The variable $\gamma(s_j)$ indicates the linearly varying circulation strength over panel j, which is written as

$$\gamma(s_j) = \gamma_j + (\gamma_{j+1} - \gamma_j)\frac{s_j}{S_j}, \tag{2.1.6}$$

where S_j is the total length of panel j, and s_j is the integration variable, shown in Equation 2.1.5 as well, increasing from 0 to total panel length S_j. In Equation 2.1.5, the varying circulations are integrated over each panel, and their summation is defined as the influence of the vortices on the velocity at point (x, y) in the system. Now, Equation 2.1.5 has to be solved for all γs to obtain the potential solution. This is done by applying a boundary condition; the velocity in the outward normal direction in the middle of the panel is zero. The implication is that there is no flow coming in, nor going out of the airfoil, which corresponds with the underlying physics of an impenetrable body. As a natural consequence, x and y in Equation 2.1.5 are taken as the midpoints (x_i, y_i) of each panel of the airfoil. As such, the potential solution will give the tangential velocity and consequently the pressure, embodied in the pressure coefficient C_p, at these midpoints, resulting in the familiar pressure distributions. The manner in which the system is made solvable is by applying the zero normal velocity as a Neumann boundary condition:

$$\frac{\partial}{\partial n}\phi(x_i, y_i) = \frac{\partial \phi}{\partial x_i} n_{i_x} + \frac{\partial \phi}{\partial y_i} n_{i_y} = 0, \qquad \text{for } i = 1, 2, ..., m. \tag{2.1.7}$$

There are now m equations, namely the midpoints from m panels, for $m + 1$ circulations. The equation that closes the system is the Kutta condition, and is given in section 2.3, such that the system is fully solvable.

2.2 Expanding the potential function

Taking Equation 2.1.5, the integral is somewhat simplified, introducing a natural coordinate ξ which will be substituted for the length of panel j, going from 0 to 1. Hence,

$$\xi = \frac{s_j}{S_j}, \qquad \mathrm{d}s_j = \frac{\mathrm{d}s_j}{\mathrm{d}\xi}\mathrm{d}\xi = S_j \mathrm{d}\xi. \tag{2.2.1}$$

As the integral in Equation 2.1.5 is carried out over each panel j, x_j, y_j and γ_j can be written as functions of this natural coordinate ξ as well,

$$\begin{aligned} x_j(\xi) &= x_j + (x_{j+1} - x_j)\xi, \\ y_j(\xi) &= y_j + (y_{j+1} - y_j)\xi, \\ \Gamma_j(\xi) &= \gamma_j + (\gamma_{j+1} - \gamma_j)\xi. \end{aligned} \tag{2.2.2}$$

Substituting Equation 2.2.2 into Equation 2.1.5 yields

$$\phi(x, y) = V_\infty(x \cos\alpha + y \sin\alpha) - \sum_{j=1}^{m} \int_0^1 \frac{\Gamma_j(\xi) S_j}{2\pi} \tan^{-1}\left(\frac{y - y_j(\xi)}{x - x_j(\xi)}\right) \mathrm{d}\xi, \tag{2.2.3}$$

where the sum of the integrals over all the panels is taken. A function $\beta_j(x, y, \xi) = \frac{y - y_j(\xi)}{x - x_j(\xi)}$ is substituted, that describes the angle between points (x, y) and (x_j, y_j), varying with ξ as $(x_j \to x_{j+1}, y_j \to y_{j+1})$. Rewriting the equation by taking out constants and substituting β

yields

$$\phi(x,y) = V_\infty(x\cos\alpha + y\sin\alpha) - \frac{1}{2\pi}\sum_{j=1}^{m} S_j \int_0^1 \Gamma_j(\xi)\tan^{-1}\beta_j(x,y,\xi)\mathrm{d}\xi. \tag{2.2.4}$$

Now, the formulation that allows the implementation of a Neumann boundary condition, as imposed in Equation 2.1.7, must be obtained. As such, the derivative of the potential function with respect to x and y is found,

$$\frac{\partial\phi}{\partial x} = V_\infty\cos\alpha - \frac{1}{2\pi}\sum_{j=1}^{m} S_j \int_0^1 \Gamma_j(\xi)\frac{1}{1+\beta_j^2}\frac{\partial\beta_j}{\partial x}\mathrm{d}\xi, \qquad \frac{\partial\beta_j}{\partial x} = \frac{-(y-y_j(\xi))}{(x-x_j(\xi))^2}, \tag{2.2.5}$$

$$\frac{\partial\phi}{\partial y} = V_\infty\sin\alpha - \frac{1}{2\pi}\sum_{j=1}^{m} S_j \int_0^1 \Gamma_j(\xi)\frac{1}{1+\beta_j^2}\frac{\partial\beta_j}{\partial y}\mathrm{d}\xi, \qquad \frac{\partial\beta_j}{\partial y} = \frac{1}{x-x_j(\xi)}. \tag{2.2.6}$$

Now, the function $\Gamma_j(\xi)$ will be rewritten, such that the individual circulations γ can be resolved explicitly,

$$\Gamma_j(\xi) = \gamma_j(1-\xi) + \gamma_{j+1}\xi. \tag{2.2.7}$$

Rewriting Equation 2.2.5 and Equation 2.2.6 yields

$$\begin{aligned}
\frac{\partial\phi}{\partial x} &= V_\infty\cos\alpha - \frac{1}{2\pi}\sum_{j=1}^{m} S_j\left(\gamma_j\int_0^1 \frac{\partial\beta_j/\partial x}{1+\beta_j^2}(1-\xi)\mathrm{d}\xi + \gamma_{j+1}\int_0^1 \frac{\partial\beta_j/\partial x}{1+\beta_j^2}\xi\mathrm{d}\xi\right),\\
\frac{\partial\phi}{\partial y} &= V_\infty\sin\alpha - \frac{1}{2\pi}\sum_{j=1}^{m} S_j\left(\gamma_j\int_0^1 \frac{\partial\beta_j/\partial y}{1+\beta_j^2}(1-\xi)\mathrm{d}\xi + \gamma_{j+1}\int_0^1 \frac{\partial\beta_j/\partial y}{1+\beta_j^2}\xi\mathrm{d}\xi\right).
\end{aligned} \tag{2.2.8}$$

Now the derivatives with respect to x and y are found. The identities in Equation 2.2.8 are substituted in Equation 2.1.7. It is emphasized that the x and y coordinates are not general anymore, but rather the m panel midpoints (x_i, y_i), with corresponding unit normal vectors (n_{i_x}, n_{i_y}), are substituted in place of (x, y). As such, the function β_j obtains an extra index i to make clear that the midpoint coordinates (x_i, y_i) and endpoints (x_j, y_j) of the panels are meant; the term becomes β_{ij}. Rearranging the terms, one obtains the specific form of the governing vortex panel method equation,

$$\begin{aligned}
V_\infty(n_{i_x}\cos\alpha + n_{i_y}\sin\alpha) - \frac{1}{2\pi}\sum_{j=1}^{m} S_j\left(\gamma_j\int_0^1\left[n_{i_x}\frac{\partial\beta_{ij}}{\partial x_i} + n_{i_y}\frac{\partial\beta_{ij}}{\partial y_i}\right]\frac{1-\xi}{1+\beta_{ij}^2}\mathrm{d}\xi + \ldots\right.\\
\left.\ldots + \gamma_{j+1}\int_0^1\left[n_{i_x}\frac{\partial\beta_{ij}}{\partial x_i} + n_{i_y}\frac{\partial\beta_{ij}}{\partial y_i}\right]\frac{\xi}{1+\beta_{ij}^2}\mathrm{d}\xi\right) = 0, \qquad \text{for } i = 1,2,\ldots,m.
\end{aligned} \tag{2.2.9}$$

Equation 2.2.9 has $m+1$ γs, for the endpoints of the panels. As mentioned in section 2.1, there are $i = 1, 2, \ldots, m$ equations, for m midpoints. Hence, there is one unknown too many. To close the system, the Kutta condition is introduced.

2.3 The Kutta condition

The Kutta condition states that the flow leaving a sharp object, such as the trailing edge of an airfoil, must be smooth. In other words, there is no circulation around the sharp corner of the trailing edge of an airfoil. Anderson [39] explains the Kutta condition in more detail. In mathematical terms, this means that the first and last circulation γ_1 and γ_{m+1} must be equal in magnitude, and negative each other, as to cancel out their effects and ensure smooth outflow:

$$\gamma_1 + \gamma_{m+1} = 0. \tag{2.3.1}$$

This is the closing relation that provides a full set of equations, in combination with Equation 2.2.9. Firstly rearranging Equation 2.2.9,

$$\sum_{j=1}^{m} \frac{S_j}{2\pi V_\infty} \left(\gamma_j \int_0^1 \left[n_{i_x} \frac{\partial \beta_{ij}}{\partial x_i} + n_{i_y} \frac{\partial \beta_{ij}}{\partial y_i} \right] \frac{1-\xi}{1+\beta_{ij}^2} \mathrm{d}\xi \; + ... \right. \tag{2.3.2}$$

$$\left. ... + \gamma_{j+1} \int_0^1 \left[n_{i_x} \frac{\partial \beta_{ij}}{\partial x_i} + n_{i_y} \frac{\partial \beta_{ij}}{\partial y_i} \right] \frac{\xi}{1+\beta_{ij}^2} \mathrm{d}\xi \right) = n_{i_x} \cos\alpha + n_{i_y} \sin\alpha, \qquad \text{for } i = 1, 2, ..., m,$$

and substituting matrix identities to simplify,

$$M_{ij+1} = \frac{S_j}{2\pi V_\infty} \int_0^1 \left[n_{i_x} \frac{\partial \beta_{ij}}{\partial x} + n_{i_y} \frac{\partial \beta_{ij}}{\partial y} \right] \frac{\xi}{1+\beta_{ij}^2} \mathrm{d}\xi,$$

$$N_{ij} = \frac{S_j}{2\pi V_\infty} \int_0^1 \left[n_{i_x} \frac{\partial \beta_{ij}}{\partial x} + n_{i_y} \frac{\partial \beta_{ij}}{\partial y} \right] \frac{1-\xi}{1+\beta_{ij}^2} \mathrm{d}\xi,$$

where M_{ij+1} is the matrix that linearly builds the influence of the next circulations $\gamma+1$, namely ξ increasing from 0 to 1, and N_{ij} is the matrix that linearly decreases the influence from previous ones γ, as $1-\xi$ tends to 0, starting from 1. It is emphasized that either matrix has dimension $m \times m$, but for matrix N, both indices i and j range from 1 to m, whereas for matrix M, index i ranges from 1 to m and index j ranges from 2 to $m+1$. Equation 2.3.2 now becomes

$$\sum_{j=1}^{m} (N_{ij}\gamma_j + M_{ij+1}\gamma_{j+1}) = n_{i_x} \cos\alpha + n_{i_y} \sin\alpha, \qquad \text{for } i = 1, 2, ..., m. \tag{2.3.3}$$

Now, adding the extra equation Equation 2.3.1 that will ensure the Kutta condition is satisfied, and lastly collecting all the terms, one arrives at the system

$$\begin{bmatrix} N_{11} & N_{12}+M_{12} & \cdots & \cdots & M_{1m+1} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ N_{m1} & \cdots & \cdots & N_{mm}+M_{mm} & M_{mm+1} \\ 1 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{Bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_m \\ \gamma_{m+1} \end{Bmatrix} = 2\pi V_\infty \begin{Bmatrix} (n_1)_x \cos\alpha + (n_1)_y \sin\alpha \\ \vdots \\ (n_m)_x \cos\alpha + (n_m)_y \sin\alpha \\ 0 \end{Bmatrix}, \tag{2.3.4}$$

which has $m+1$ equations for $m+1$ unknowns, and can be solved as a system of linear equations.

2.4 Results and mesh refinement

A panel method is implemented as an external inviscid layer solver, and can provide accurate and rapid solutions in cases where viscous effects are negligible. An example of a widely used panel method solver is *XFOIL* [11], hence a comparison between the two models can be made.

To demonstrate the functionality of the panel method and the convergence, a mesh refinement is done on the same test cases, where the number of x and y coordinates of an airfoil is increased such that there are more panels to form a smoother curvature. The amount of panels used is gradually going from 12 to 300. The default configuration of *XFOIL* is 160 panels, with which the results are compared. From Equation 2.3.4, having solved for the circulations γ, the velocity components u and v for each panel are obtained (from $\frac{\partial \phi}{\partial x}$ and $\frac{\partial \phi}{\partial y}$). However, typically the result from a panel method is given in the form of the pressure coefficient,

$$C_p = 1 - \left(\frac{V}{V_\infty}\right)^2. \tag{2.4.1}$$

The derivation of this equation can be found in Appendix A. The results of the implemented panel method with mesh refinement can be seen in Figure 2.2 and Figure 2.4.

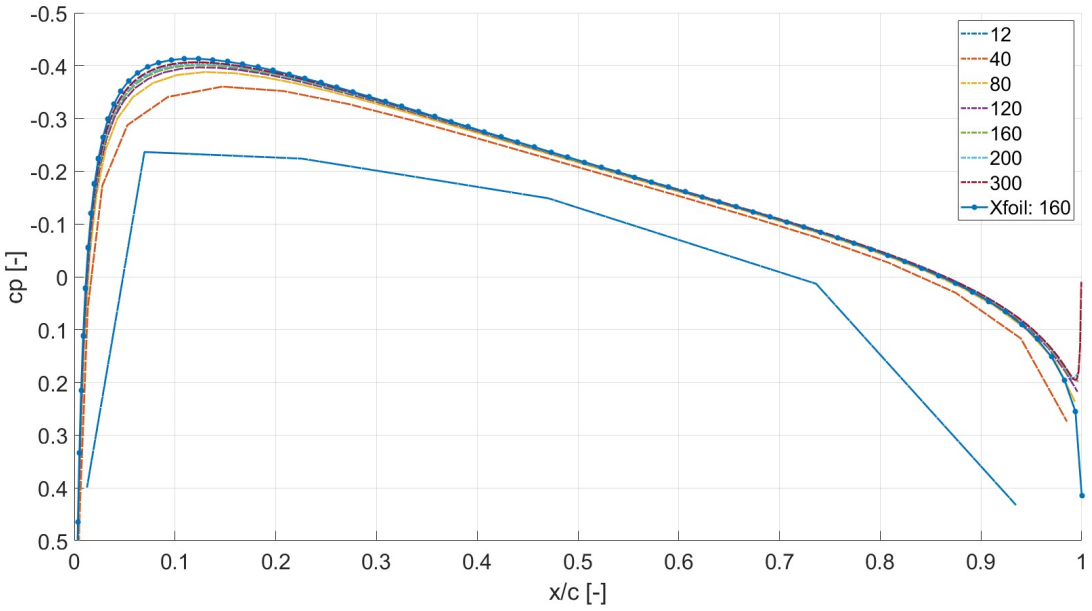

Figure 2.2: A mesh refinement of the panel method for NACA 0012 at an angle of attack of 0°.

The graph in Figure 2.2 is cut off for legibility purposes, but it is implied that the graph descends to a pressure coefficient of 1, namely the leading edge stagnation point. The symmetry condition of a symmetrical airfoil at 0° is satisfied, as the lines for pressure and suction side are on top of each other. For the symmetrical NACA 0012 airfoil, the correspondence gradually becomes better with increasing number of panels at the leading edge. At the trailing edge, however, an upward tail starts to grow for a high amount of panels, where it should actually descend towards the value of 1, namely the trailing edge stagnation point in the inviscid case. This might be attributed to the fact that the airfoil is not actually a closed body, but rather,

the first and last coordinate of the airfoil are set such that they reach sufficiently close to zero. Figure 2.3 shows the pressure distribution, zoomed in at the trailing edge for legibility purposes. The pressure distribution over the entire domain can be seen in the lower left corner of Figure 2.3. Moreover, the open and closed geometry at the trailing edge can be seen in the upper right corner. Figure 2.3 shows the results for the open and (manually) closed trailing edge, however this manual close causes a sudden kink in the curvature of the airfoil trailing edge, shown in the upper right graph, which still results in a spike in the pressure coefficient, albeit smaller than the result of the open trailing edge. This spike is physically interpreted as a sudden acceleration of the flow, caused by the abrupt, sharp angle between two panels. In section 2.6, it will be demonstrated that the wake resolves this problem.

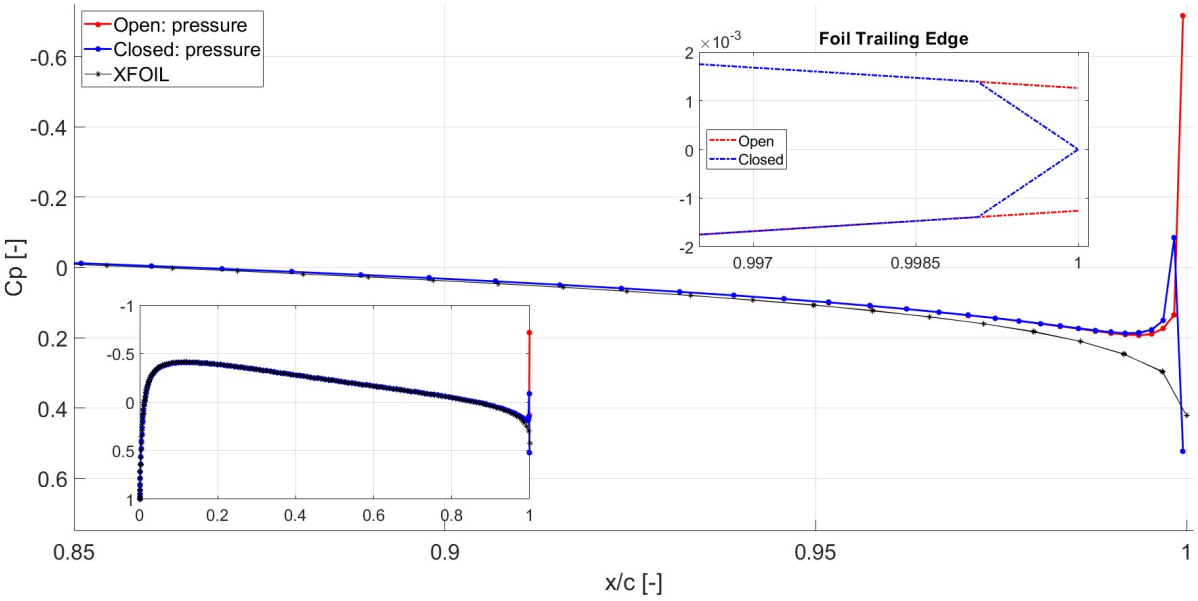

Figure 2.3: The pressure distribution for an open and fully closed trailing edge of a NACA 0012 airfoil at 0°, modeled using 300 panels, and the XFOIL result for reference.

A similar mesh refinement is done for a cambered airfoil at an angle, to show that the panel method can also accurately calculate pressure distributions for different circumstances. The NACA 2412 airfoil is chosen, at an angle of attack of 10°. The result can be seen in Figure 2.4. The gradual approach to the *XFOIL* suction peak can be seen with the increase of the amount of panels. For 300 panels, a similar tail can be seen at the leading edge as in Figure 2.2, but it is much smaller in magnitude, and is resolved by the wake as well. From the mesh refinement, it is concluded that a number of panels between 160 and 200 is adequately accurate to perform simulations with, as the results for this amount of panels is sufficiently close to *XFOIL*'s result, but not overly high to unnecessarily increase computational cost.

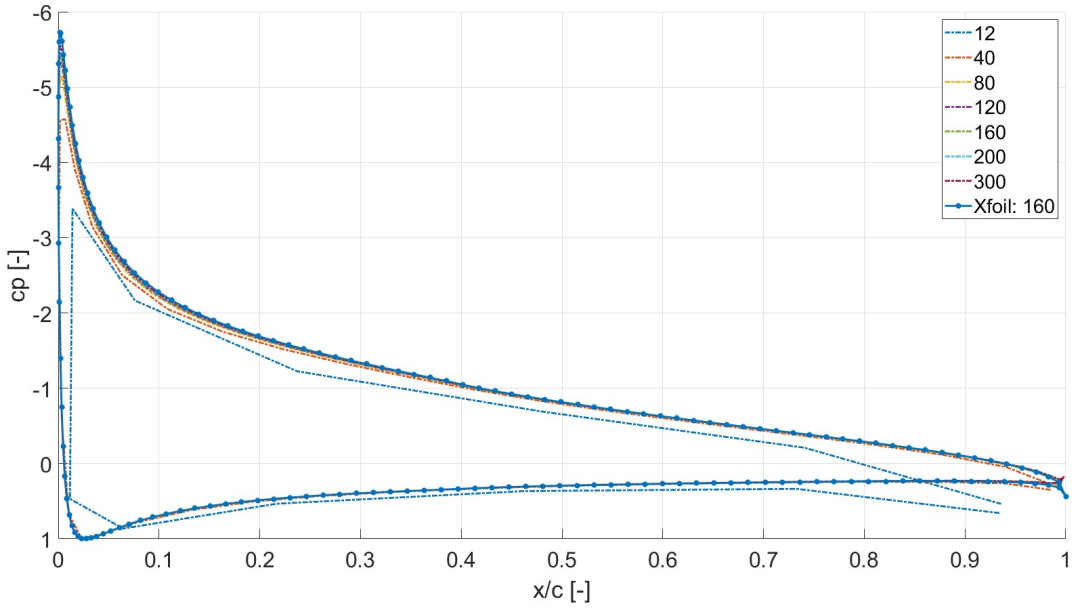

Figure 2.4: A mesh refinement of the panel method for NACA 2412 at an angle of attack of 10°.

2.5 Coupling with boundary layer solver

A different application of panel methods is to couple it with a boundary layer solver. Iteratively, the two models work together to arrive at a converged solution. However, the displacement thickness calculated in the boundary layer model works in supplying a different, somewhat larger, effective body. The area between the actual airfoil and this effective body is considered the boundary layer, whereas everything outside this effective body is considered inviscid. Cebeci [4] and Veldman [18] describe how the boundary layer solver and inviscid model are coupled.

Firstly, a zero normal velocity was imposed in Equation 2.1.7. In coupling with an inviscid solver, a nonzero 'blowing' positive normal velocity is assumed and applied, instead of the zero Neumann boundary condition, and derived from

$$v = \frac{\mathrm{d}}{\mathrm{d}x}(u_e \delta^*), \quad \text{such that} \quad \frac{\partial \phi}{\partial n} = v, \tag{2.5.1}$$

also known as the surface transpiration concept [40]. The idea behind this concept is the forming of the effective body around the airfoil and 'pushing' the outer inviscid layer away. Since the inviscid layer is pushed away and a different effective aerodynamic body is created, Equation 2.3.1, which made the flow satisfy the Kutta condition, is not valid on the airfoil surface anymore. Hence, the condition should be applied at the edge of the effective body instead of on the trailing edge itself.

When viscous effects are taken into consideration, the Kutta condition is satisfied if the velocities above and below the trailing edge are equal, i.e. equal pressure, and have parallel outflow angles. However, as mentioned, in a viscous situation the condition is not imposed on the airfoil trailing edge itself, but rather one δ^* away from the trailing edge, such that the difference between the velocities at the edge of the effective aerodynamic body is zero, as

explained by Cebeci [4]. It will be seen that a wake is necessary to accurately model smooth outflow on the trailing edge. When imposing the Kutta condition in a viscous case without a wake, similar to how it is done in Equation 2.3.1 in the inviscid case, but now one δ^* away from the trailing edge, one cannot guarantee that the upper and lower outflow have the same velocity as well as parallel outflow angles. One can set a difference in circulation in the Kutta condition around the trailing edge to try to resolve the matter of differing angles. Instead of demanding in Equation 2.3.1 that the first circulation be minus the last circulation, a difference between the two circulations is now imposed to try to resolve the issue of trailing edge outflow of different angles. Let the Kutta condition be rewritten,

$$\gamma_1 + \gamma_{m+1} = \Delta\Gamma, \tag{2.5.2}$$

where $\Delta\Gamma$ is a user-specified value. This $\Delta\Gamma$ is manipulated to see the effect that it has on the velocity and the outflow angles, one displacement thickness δ^* above and below the trailing edge. In Figure 2.5, the effects of this imposed circulation difference can be seen. For the symmetrical case of NACA 0012 airfoil at 0°, the two outflow angles one δ^* above and below the trailing edge are shown. Also shown is the difference in velocity Δv between the two flows that leave the trailing edge. It can be seen that, for zero circulation difference, i.e. the original Equation 2.3.1 to satisfy the Kutta condition, there is inherently a difference between the outflow angles. In imposing parallel outflow angles by setting a difference in circulation $\Delta\Gamma$ of 7 (or -7), there is a difference in velocities leaving the trailing edge of around 0.3 (or -0.3).

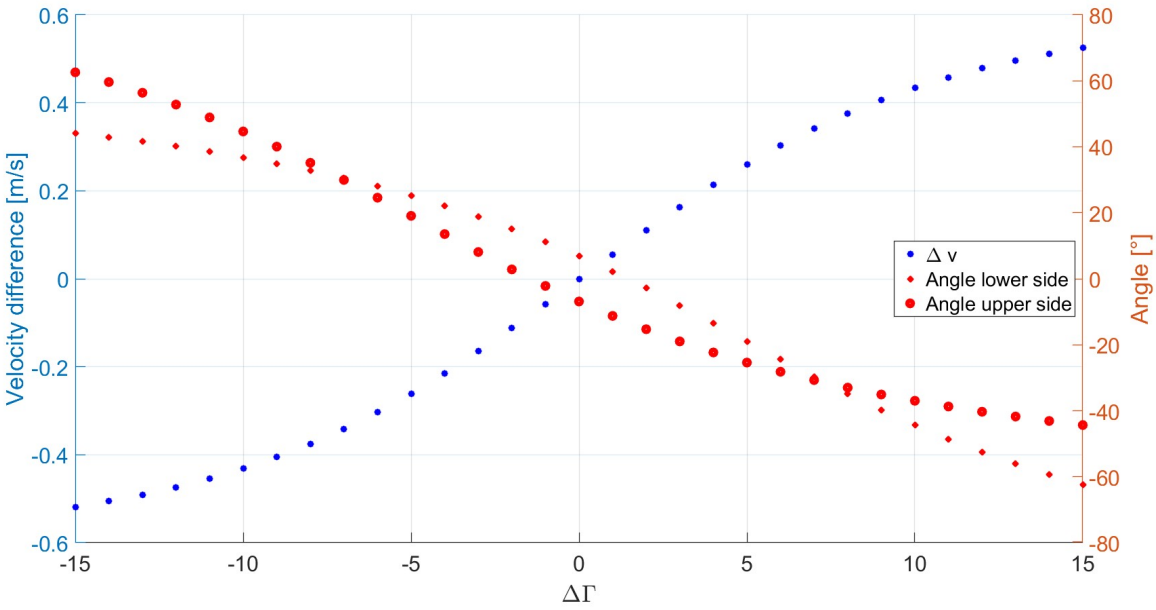

Figure 2.5: Difference in circulation around the trailing edge of a NACA 0012 airfoil at 0°, including velocity difference and outflow angle on lower and upper airfoil side, one displacement thickness δ^* away from the trailing edge.

Figure 2.5 explains the necessity of a wake to ensure smooth outflow of the trailing edge and satisfy the Kutta condition, as there cannot be equal velocities and parallel outflow of the airfoil without a wake. To guarantee both, the work has to be 'divided'. The wake ensures that the

flows leaving the trailing edge are directed outwards with parallel angles, while the last row in Equation 2.3.4 serves to equate the trailing edge outflow velocities.

2.6 The wake

When the circulations γ have been calculated, the velocity at any point can be given by Equation 2.2.8. Considering that the gradient of the potential is the velocity, and is dependent on the x and y position where it is evaluated, an ordinary differential equation can be set up,

$$\begin{aligned} u(x,y) &= \frac{\partial x}{\partial t}(x,y) = \frac{\partial \phi}{\partial x}(x,y), \\ v(x,y) &= \frac{\partial y}{\partial t}(x,y) = \frac{\partial \phi}{\partial y}(x,y), \end{aligned} \tag{2.6.1}$$

where the definitions of the potential's gradient can be substituted from Equation 2.2.8 to obtain an explicit expression from which the velocity in x and y direction can readily be obtained. Initial conditions are chosen sufficiently close to the trailing edge, on the upper and lower side, and used as input by an explicit ODE solver, available in MATLAB, to calculate the trajectory of a fluid particle with an initial position slightly above or below the airfoil's trailing edge. Coordinates along this trajectory are chosen according to a cosine distribution, thus obtaining two wake lines with a higher density of points closer to the trailing edge. Using this newly-defined geometry as input for the panel method, the pressure distribution is again calculated. The result is presented in Figure 2.6, as well as the airfoil and wake geometry, and optimal agreement with *XFOIL* is found. For the sake of completeness, the NACA 2412 including wake is presented in Figure 2.7, showing that the spike at the trailing edge has disappeared, and that the result is indeed reliable.

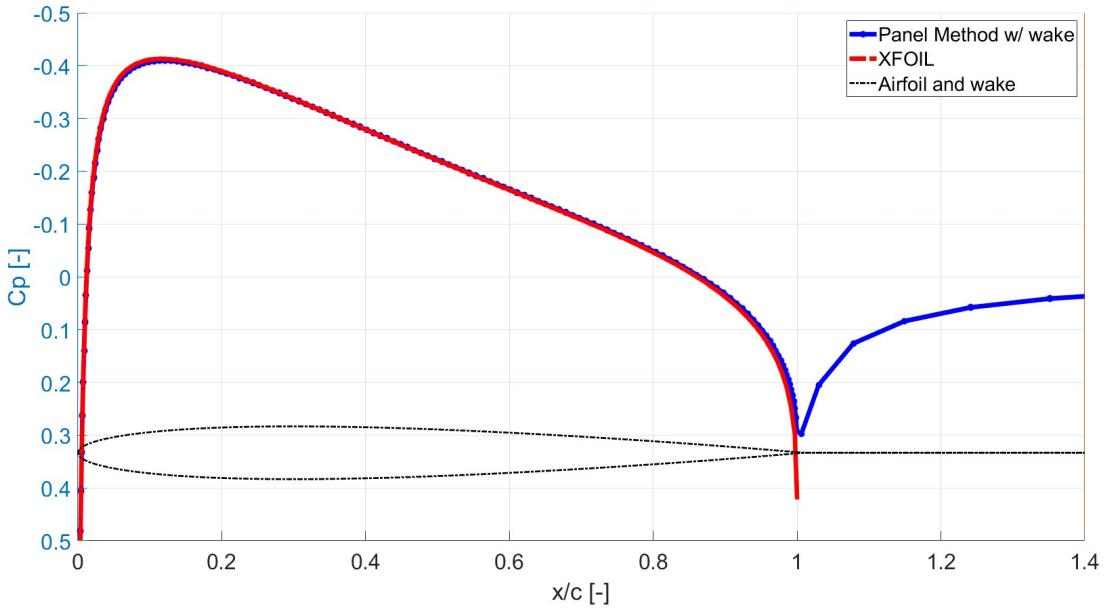

Figure 2.6: A wake generated for NACA 0012 at an angle of attack of 0°, compared with XFOIL.

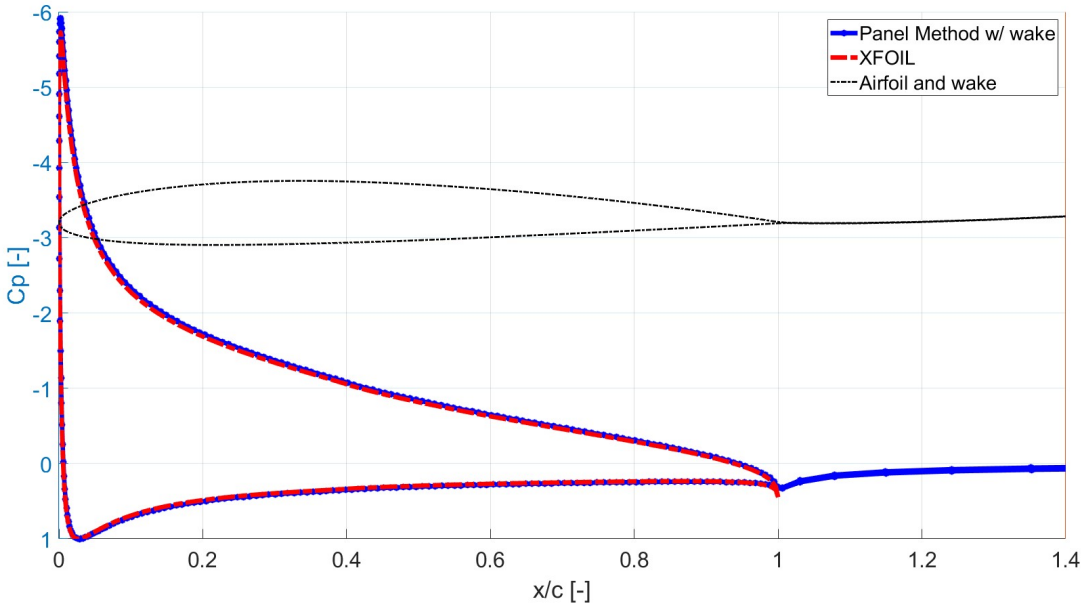

Figure 2.7: A wake generated for NACA 2412 at an angle of attack of 10°, compared with XFOIL.

Chapter 3

Integral Boundary Layer Equations

This chapter considers the system of equations employed by Veldman in his article, namely the Von Kármán momentum integral equation and Head's entrainment equation. Thwaites' method is discussed as the method to calculate the laminar region of the boundary layer. The closing relations of the model are shown, and the complete system of boundary layer equations is presented. Goldstein's singularity is demonstrated, and emphasis is put on the semi-strong coupling, for which the analytical model is derived.

3.1 The system of equations

The integral boundary layer equation, derived by Theodore von Kármán in 1950, is taken as a first equation. The derivation from the boundary layer equations to the Von Kármán equation can be followed step by step in Drela, [5], chapter 4. The equation is as follows,

$$\frac{1}{u_e^2}\frac{\partial}{\partial t}(u_e\theta H) + \frac{\partial\theta}{\partial x} + \frac{\theta}{u_e}(2+H)\frac{\partial u_e}{\partial x} = \frac{1}{2}c_f, \tag{3.1.1}$$

where the variables u_e, θ, H and c_f are the edge velocity, the momentum thickness, the shape factor δ^*/θ and the skin friction coefficient, respectively. Formulas for these variables, also known as the closure relations, will be given in section 3.3. In the present study, only steady situations are considered, and thus the time-dependent term drops, simplifying the equation to

$$\frac{\mathrm{d}\theta}{\mathrm{d}x} + \frac{\theta}{u_e}(2+H)\frac{\mathrm{d}u_e}{\mathrm{d}x} = \frac{1}{2}c_f. \tag{3.1.2}$$

Consistently throughout the present study, the boundary layer is calculated in two regions. The leading edge stagnation point on an airfoil is taken as the start, after which the upper or lower side of the foil is calculated. Firstly, the laminar region is calculated using Thwaites' method, which will be introduced in section 3.2. Laminar flow is assumed until transition to turbulent flow sets in at a user-specified location (for example $s = 0.1$), at which point a switch is made from Thwaites' equations to Equation 3.1.2. To form a closed system of equations with the Von Kármán equation, Head's Equation 3.3.2 is added as the closure set, to be introduced in section 3.3. This system of equations is initially solved using an explicit ODE solver, available in MATLAB. However, this ODE solver does not provide accurate solutions after Goldstein's

singularity, as will be shown in section 3.4, and is only used to obtain an initial condition for the higher-order Galerkin solution of the boundary layer, that already approximates the actual solution before separation. The higher-order Galerkin method will be introduced in chapter 4.

3.2 Thwaites' method

In 1960, Bryan Thwaites derived a method to analytically integrate the momentum thickness from the edge velocity [41], which upholds reasonably well for laminar boundary layers. Following Drela's derivation in [5], the von Kármán equation is manipulated by multiplying it with two times the local momentum thickness Reynolds number $Re_\theta = \frac{u_e\theta}{\nu}$. Thwaites arrived at an identity

$$\frac{u_e}{\nu}\frac{\mathrm{d}(\theta^2)}{\mathrm{d}s} = 2[\mathcal{T} - (H+2)\lambda] \equiv \mathcal{F}_\theta, \tag{3.2.1}$$

where a transition to streamwise coordinate s has been made, and with the substituted identities

$$\mathcal{T} \equiv Re_\theta \frac{c_f}{2}, \qquad\qquad \lambda \equiv \frac{\theta^2}{\nu}\frac{\mathrm{d}u_e}{\mathrm{d}s}. \tag{3.2.2}$$

In Equation 3.2.2, the first term $\mathcal{T}$ represents a non-dimensional wall shear, whereas the second term λ stands for a streamwise, non-dimensional, external velocity gradient. Thwaites empirically found the relation for the right-hand side $\mathcal{F}_\theta = 0.45 - 6\frac{\theta^2}{\nu}\frac{\mathrm{d}u_e}{\mathrm{d}s} = 0.45 - 6\lambda$ to be the best fit. Substituting this identity in Equation 3.2.1 enabled the possibility of direct explicit integration by means of an inverted product rule. Multiplying Equation 3.2.1 with νu_e^5, he found

$$\left[\frac{u_e}{\nu}\frac{\mathrm{d}(\theta^2)}{\mathrm{d}s} = 0.45 - 6\frac{\theta^2}{\nu}\frac{\mathrm{d}u_e}{\mathrm{d}s}\right]\nu u_e^5 \quad \longrightarrow \quad \frac{\mathrm{d}(u_e^6\theta^2)}{\mathrm{d}s} = 0.45\nu u_e^5. \tag{3.2.3}$$

If one knows the edge velocity distribution and the kinematic viscosity, Equation 3.2.3 can be directly integrated, such that

$$\theta^2(s) = \frac{1}{u_e^6(s)}\left[u_e^6(s_0)\theta(s_0) + 0.45\nu\int_{s_0}^{s} u_e^5(s')\,\mathrm{d}s'\right], \tag{3.2.4}$$

where the $'$ simply denotes the integration variable. No initial data is required, as the simulation is run from a stagnation point, where $s = 0$ and $u_e = 0$, and thus, $\theta(0)$ is immaterial. Such is Thwaites' method, which will be used to obtain results for the momentum thickness θ in the laminar region of the boundary layer. The point of transition to turbulent flow is user-specified, and in the present study, it is set at 0.1 length in the streamwise coordinate. Having obtained the momentum thickness, one can calculate the normalized velocity gradient λ from Equation 3.2.2, and with that, the shape factor is calculated by means of Drela's [5] curve fits:

$$H = 2.61 - 4.1\lambda + 14\lambda^3 + \frac{0.56\lambda^2}{(\lambda + 0.18)^2}. \tag{3.2.5}$$

The normalized wall shear $\mathcal{T}$, from which the skin friction coefficient c_f is calculated with Equation 3.2.2, is given by the curve fit

$$\mathcal{T} = 0.220 + 1.52\lambda - 5\lambda^3 - \frac{0.072\lambda^2}{(\lambda + 0.18)^2}. \tag{3.2.6}$$

The skin friction coefficients (as well as the pressure coefficients) will be used to obtain the lift and drag coefficients, the calculation of which is given in Appendix A.

3.3 Head's entrainment equation

The unknowns for which the Von Kármán equation will be solved are θ and H. Hence, there is need for another equation to close the system. This equation is found in the method of entrainment, or carrying something along by a fluid, by Head [19]. Head proposes an altered shape factor,

$$H_1 = \frac{\delta - \delta^*}{\theta},$$

where δ stands for the boundary layer thickness. Head's entrainment equation is given by

$$\frac{1}{u_e}\frac{\partial}{\partial t}(\theta(H_1 + H)) + \frac{1}{u_e}\frac{\partial}{\partial x}(u_e \theta H_1) = C_E, \tag{3.3.1}$$

where C_E denotes the entrainment coefficient, i.e. the coefficient that denotes the 'carrying' of a fluid by a turbulent flux [42]. The steady version is considered, and the brackets are expanded, yielding

$$H_1\frac{\mathrm{d}\theta}{\mathrm{d}x} + \theta\frac{\mathrm{d}H_1}{\mathrm{d}x} + \frac{H_1\theta}{u_e}\frac{\mathrm{d}u_e}{\mathrm{d}x} = C_E. \tag{3.3.2}$$

Closure relations for $H_1 = H_1(H)$ and $C_E = C_E(H_1)$ will be provided below. Putting both Equation 3.1.2 and Equation 3.3.2 together in a matrix system of equations, one arrives at

$$\begin{bmatrix} 1 & 0 \\ H_1 & \theta\frac{\mathrm{d}H_1}{\mathrm{d}H} \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}x}\begin{Bmatrix} \theta \\ H \end{Bmatrix} = \begin{pmatrix} \frac{1}{2}c_f - \frac{\theta}{u_e}(2+H)\frac{\mathrm{d}u_e}{\mathrm{d}x} \\ C_E - \frac{H_1\theta}{u_e}\frac{\mathrm{d}u_e}{\mathrm{d}x} \end{pmatrix}. \tag{3.3.3}$$

Moreover, closure relations for the additional variables in the equations are needed. Assuming that the edge velocity is known from the panel method, such relations are needed for c_f, H_1 and C_E, and are obtained from Veldman [18]. Some of these closure relations were found to contain some typing errors and were corrected; the corrected closure relations are given.

$$Re_\theta = \frac{u_e\theta}{\nu}, \tag{3.3.4}$$

$$c_{f_0} = \frac{0.01013}{log(Re_\theta) - 1.02}, \tag{3.3.5}$$

$$c_f = c_{f_0}\left[\frac{0.9}{\left(6.55\left(1 - \sqrt{\frac{1}{2}c_{f_0}}\right) - 0.4\right)H} - 0.5\right], \tag{3.3.6}$$

$$h_t = min(H, 0.5 * (H + 2.732)), \tag{3.3.7}$$

$$H_1 = \begin{cases} 1.9710 + \frac{3.2h_t}{h_t+2.4689}, & \text{if } H > 4, \\ h_t \frac{0.5h_t+1}{h_t-1}, & \text{if } H \leq 4, \end{cases} \tag{3.3.8}$$

$$C_E = 0.0306(H_1 - 3)^{-0.6169}. \tag{3.3.9}$$

Using Equation 3.3.3, and all of the above closure relations, a fully closed system of (ordinary differential) equations is obtained, that can be integrated explicitly when the edge velocity distribution is known.

3.4 Explicit integration and Goldstein's singularity

Taking a closer look at Equation 3.3.3, more specifically the system matrix

$$\begin{bmatrix} 1 & 0 \\ H_1 & \theta \frac{\mathrm{d}H_1}{\mathrm{d}H} \end{bmatrix},$$

one concludes that, for a value of $\frac{dH_1}{dH} = 0$, a column of zeroes exists, rendering the matrix singular, and thus the simulation breaks down. This is better seen in Figure 3.1, where some relations between Head's entrainment coefficient and the shape factor are displayed.

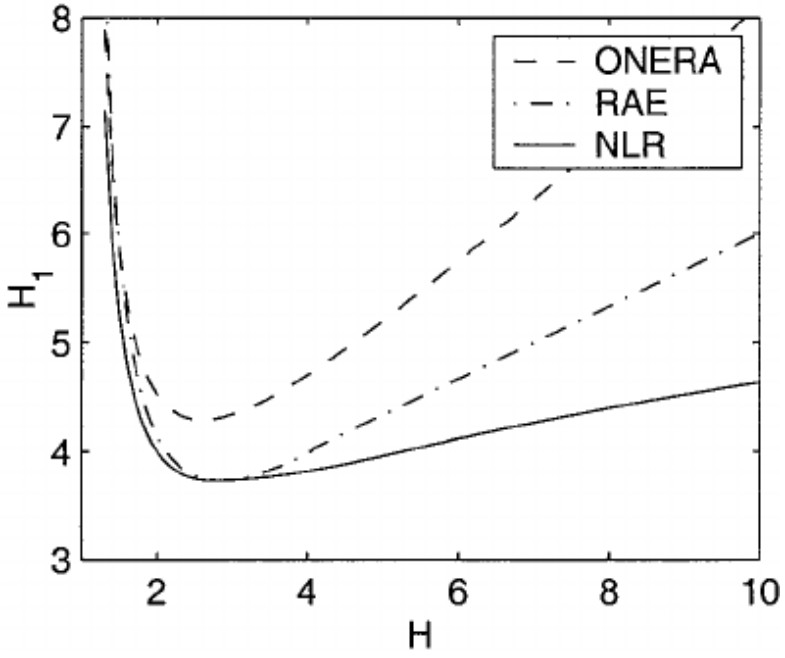

Figure 3.1: Different H1-H relations from ONERA [43], RAE [44] and NLR [45], graph taken from Veldman [46].

All relations show a local minimum at $H \approx 2.7$, where accordingly $\frac{dH_1}{dH} = 0$. Generally, the shape factor of boundary layers lies around 2.6 for laminar flow, and 1.4 for turbulent [5], which in Figure 3.1 lies to the left of this minimum. However, when a flow nears separation, the shape factor rises from its turbulent value of 1.4 upward, until it reaches the local minimum shown in Figure 3.1, and as such, $\frac{dH_1}{dH}$ indeed becomes 0, and a column of zeroes forms in the

system matrix. The matrix becoming singular is Goldstein's singularity at work, and causes the system to break down when a flow is nearing separation. Beyond this point of separation, explicit integration is not possible anymore, as the inverted system matrix diverges to infinity.

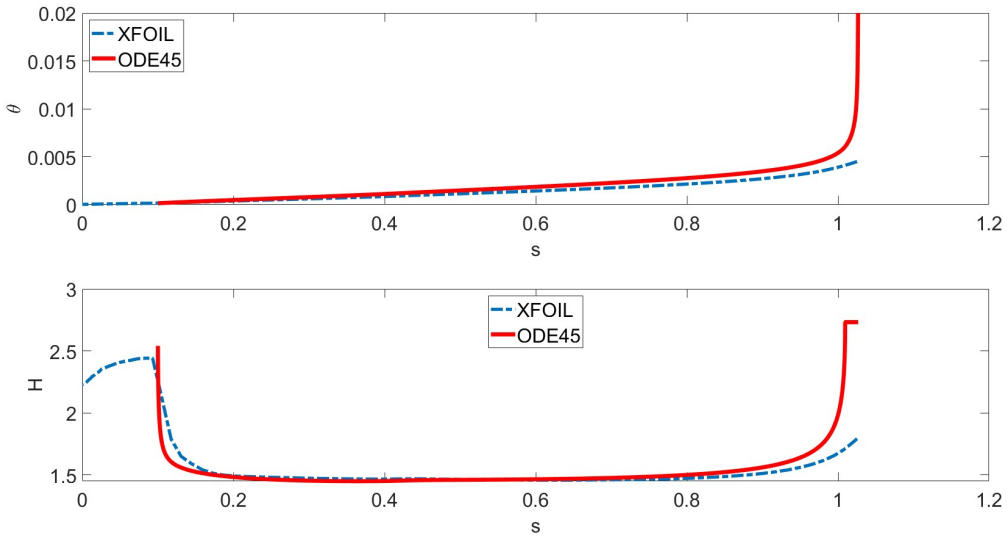

Figure 3.2: Momentum thickness and shape factor on suction side of a NACA 2412 at 0°, Reynolds $1 * 10^6$.

An example of a simulation that breaks down, can be seen in Figure 3.2. The suction side of a NACA 2412 airfoil is modeled at 0° angle of attack, and a Reynolds number of $1 * 10^6$, with a forced transition at the streamwise coordinate of $s = 0.1$. The result from the explicit ODE solver that is available in MATLAB is shown, as well as *XFOIL*'s result for reference. A study by Kwon et al. [47] states that before a turbulent flow starts to separate, the shape factor rises to between 2 and 2.5. In the lower graph in Figure 3.2, the *XFOIL* result has not yet reached this value, whereas the shape factor from the explicit solver is already increasing when nearing the trailing edge, after which there is exponential growth in both momentum thickness and shape factor. From Figure 3.2, it is clear that the results from the ODE solver diverge to a high value, and the simulations would break down were it not for manual clipping of the results. This manual clipping is set at the shape factor separation value of 2.7, which is clearly visible in the lower graph in Figure 3.2, and enables the solver to still produce results, but one can conclude that these results are not reliable. At this point and beyond, explicit integration of the differential equations is no longer possible.

Figure 3.2 shows the numerical consequence of Goldstein's singularity, and highlights the importance and the necessity of a semi-strong solution employed in a boundary layer model if it is to produce sensible results in a situation where separation might occur. Hence, the semi-strong interaction between inviscid velocity and boundary layer is introduced as a solution to bypass Goldstein's singularity of separation and produce reliable results.

3.5 Veldman's quasi-simultaneous interaction scheme

Veldman [18] proposes a scheme that involves using the change in displacement thickness directly in a submodel to update the external velocity in the boundary layer simulations. The principal

idea behind this scheme is to circumvent Goldstein's singularity of separation. Where there would previously be a 'hierarchical' relation between the external panel method solver and the internal boundary layer model, as the latter is dictated by the former, now they are merged in a quasi-simultaneous structure, creating the semi-strong solution. The overall strategy is to use a simple analytical model to improve the convergence of the weak coupling interactions between the boundary layer solver and the panel method. This specific analytical model will be derived in section 3.6. Firstly, a description is given for when the semi-strongly coupled model has converged. The input variables displacement thickness δ^* and external velocity u are defined. Let the boundary layer solver be denoted by the operator S (as in solver), such that $\delta^* = Su$, or equivalently, $u = S^{-1}\delta^*$, and the panel method (with prescribed blowing) by the operator P, such that $u = P\delta^*$. Now, a representation of the functional model is given in the form of two equations and two variables. The system will have converged when there is no more (or very little) change between the variables between iterations. The solution of these two equations is the desired converged state. For example, the converged δ^* distribution is such that $(P - S^{-1})\delta^* = 0$, or alternatively, the converged velocity distribution is such that $(S - P^{-1})u = 0$. The interaction proposed by Veldman is obtained by inserting the analytical model in such a way as to have the following combined interactive system,

$$\delta^{*(n)} = Su^{(n)} \tag{3.5.1}$$

$$u^{(n)} = P\delta^{*(n-1)} + A\left(\delta^{*(n)} - \delta^{*(n-1)}\right) \tag{3.5.2}$$

where A denotes the analytical model. A unified equation can be derived for the (modified) boundary layer solver, namely

$$S^{-1}\delta^{*(n)} = P\delta^{*(n-1)} + A\left(\delta^{*(n)} - \delta^{*(n-1)}\right) \ , \tag{3.5.3}$$

whose alternative form,

$$\left(S^{-1} - A\right)\delta^{*(n)} = \left(P - A\right)\delta^{*(n-1)}, \tag{3.5.4}$$

makes clear that, upon convergence, as $\delta^{*(n-1)} \longrightarrow \delta^{*(n)}$, the obtained solution matches that of the original problem, namely $(P - S^{-1})\,\delta^* = 0$. A visualization of the scheme, and the difference with the direct, weak solver, is shown in Figure 3.3, where $\delta^{*(n)} - \delta^{*(n-1)}$ is denoted as $\Delta\delta^*$.

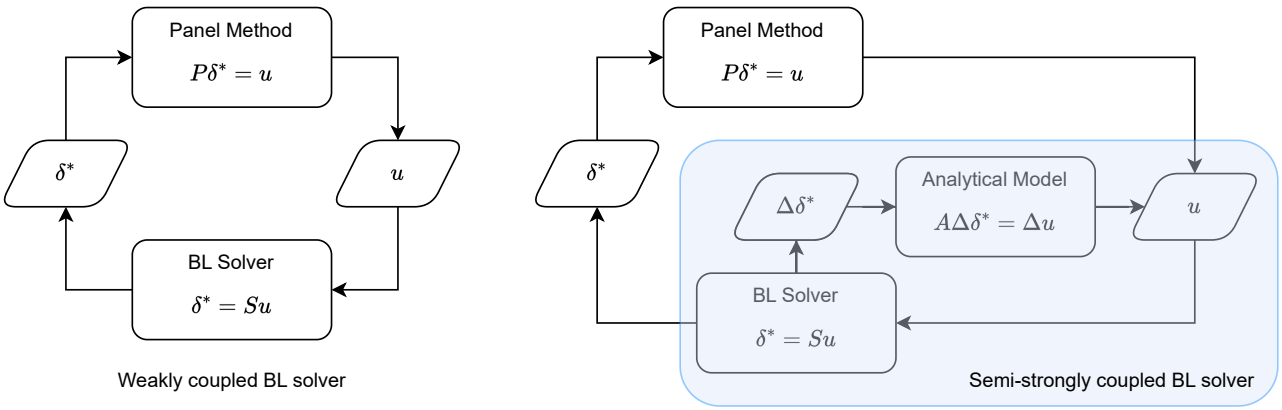

Figure 3.3: Block diagram of the weakly and semi-strongly coupled solvers.

3.6 The analytical model

The analytical model is introduced by taking the thin-airfoil approximation from Schlichting [37], also known as the skeleton theory. Schlichting explains the theory after describing the influence of the elementary solutions (given in section 2.1 as well) on the body contour formed in potential theory; the skeleton theory serves to give a good prediction of the lift generated, and relies on the assumption that the body is thin and just slightly cambered. As such, the circulations can be placed on the chord line instead of on the skeleton line, the process of which is seen in Figure 3.4, going from (a) to (b). In (c) of Figure 3.4, the circulation distribution is shown along the chord line, and the variable $k(x)$ is introduced, representing the vortex strength per unit length. Defining a strip on the chord line of width $\mathrm{d}x$, the circulation of this strip can be calculated,

$$\mathrm{d}\Gamma = k(x)\mathrm{d}x. \tag{3.6.1}$$

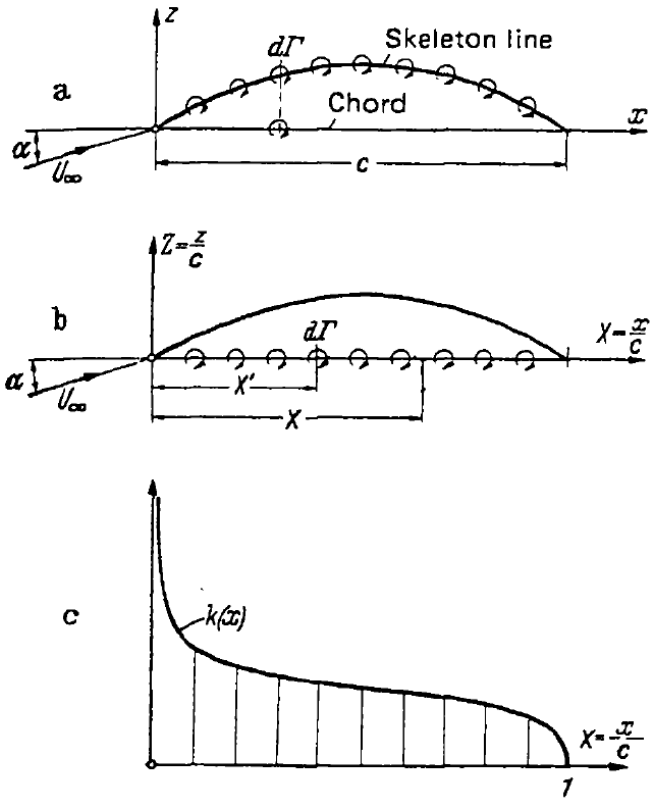

Figure 3.4: (a) Vortex distribution on skeleton line, (b) Vortex distribution on chord, (c) Circulation distribution along chord, image taken from Schlichting [37].

The vortex density $k(x)$ induces changes Δ in the velocity components u and v, which Schlichting quantifies in two equations using the law of Biot-Savart,

$$\Delta u(x,y) = \frac{1}{2\pi}\int_0^c k(\xi)\frac{y}{(x-\xi)^2+y^2}\mathrm{d}\xi, \tag{3.6.2}$$

$$\Delta v(x,y) = -\frac{1}{2\pi}\int_0^c k(x')\frac{x-\xi}{(x-\xi)^2+y^2}\mathrm{d}\xi. \tag{3.6.3}$$

The at (x,y) induced velocities are given by Equation 3.6.2 and Equation 3.6.3, where c is the

chord length and ξ is the integration variable for the vortex density. Note that in Equation 3.6.2 and Equation 3.6.3, there is a singularity for $\xi = x$. This singularity will be dealt with in subsection 4.2.1. As the profile is assumed to be slightly cambered, the circulations are placed on the chord line, and thus, y becomes 0. The equations are taken with limit $y \to 0$, becoming

$$\Delta u(x) = \pm\frac{1}{2}k(x), \tag{3.6.4}$$

$$\Delta v(x) = -\frac{1}{2\pi}\int_0^c k(\xi)\frac{\mathrm{d}\xi}{x-\xi}. \tag{3.6.5}$$

In Equation 3.6.4, the integral with the singularity has disappeared due to the limit of $y \to 0$, therefore the vorticity $k(x)$ has become a direct function of the singularity coordinate x. Equation 3.6.5 can be rewritten with an inverse Cauchy relation [48], such that

$$k(x) = \frac{2}{\pi}\int_0^c \Delta v(\xi)\frac{\mathrm{d}\xi}{x-\xi}. \tag{3.6.6}$$

It should be noted that, in applying the inverse Cauchy relation, the variables x and ξ were also switched. Evaluating the integral in Equation 3.6.6, $v(\xi)$ becomes $v(x)$, a function of the singularity position x, similar to $k(x)$ in Equation 3.6.4. Now, Equation 3.6.6 can be substituted in Equation 3.6.4, and one arrives at what Cebeci et al. [49] called the perturbation velocity to account for the viscous effects,

$$\Delta u(x) = \frac{1}{\pi}\int_0^c \Delta v(\xi)\frac{\mathrm{d}\xi}{x-\xi}. \tag{3.6.7}$$

Equation 3.6.7 describes the changes induced on the velocity by changes in a velocity Δv, as a result of a vortex distribution on the chord line. The inviscid velocity from the panel method can be added to form a combined model for the total velocity $u(x)$, such that

$$u(x) = u_0(x) + \frac{1}{\pi}\int_0^c v(\xi)\frac{\mathrm{d}\xi}{x-\xi}. \tag{3.6.8}$$

Lastly, Veldman [18] gives the final relation that prescribes wall transpiration as the gradient of velocity and displacement thickness,

$$v(\xi) = \frac{\mathrm{d}(u\delta^*)}{\mathrm{d}\xi}, \tag{3.6.9}$$

with which the model is complete,

$$u(x) = u_0(x) + \frac{1}{\pi}\int_0^c \frac{\mathrm{d}(u\delta^*)}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{x-\xi}. \tag{3.6.10}$$

The idea behind Equation 3.6.10 is to update the external velocity distribution simultaneously with the changing of the displacement thickness, thus creating the semi-strong solution in the boundary layer model. The model is further expanded to fit into the higher-order Galerkin scheme in subsection 4.2.1.

Chapter 4

Higher-Order Galerkin Applied to the Integral Boundary Layer Equations

This chapter will discuss the theory of higher-order Galerkin and Lagrange polynomials, and show a one-element example discretization of the advection-diffusion equation. It will proceed to show the spatial discretization of the governing equations employed in the boundary layer model and explain collocation of the Lagrange nodes to lump the mass matrix, as well as how the interaction law is incorporated in the spatial discretization. The point-implicit iterative scheme is completely derived, and spectral vanishing viscosity is explained and introduced to stabilize the higher-order Galerkin scheme. Finally, an order refinement of an arbitrary solution is presented to show convergence.

4.1 Theory of higher-order Galerkin

Galerkin methods, named after the Russian mathematician Boris Galerkin, are an intricate spatial discretization method in Computational Fluid Dynamics. Its governing principles rely on discretizing a function or a partial differential equation as a polynomial that is constructed through a summation of projection constants multiplied with a set of basis functions. The elegance behind the method is that it allows for mesh refinement as well as order increase to augment accuracy of the numerical solution. A distinction that can be made between Galerkin methods is continuous [25], and discontinuous [50].

The principal difference between Continuous and Discontinuous Galerkin, as mentioned in section 1.2, is that in the former, continuity of the zeroth degree $\mathcal{C}^0$ over interfaces is guaranteed, whereas the latter allows for discontinuities on interfaces, hence the name difference between the two. The advantage that these discontinuities offer is the numerical dissipation via the fluxes in the scheme. However, in the present study, to not overcomplicate the implementation, only one element will be used in the turbulent domain, and as such, only two interfaces will be present, namely the edges of the single element. With only two interfaces, a scheme based on Discontinuous Galerkin would have little stabilization due to the fluxes. To stabilize the numerics, additionally spectral vanishing viscosity [22] would have to be applied, which has been developed more for Continuous than for Discontinuous Galerkin. It is noted that, in the case of implementation of multiple elements, Discontinuous Galerkin could be considered

worthwhile as discretization scheme. In any case, a Dirichlet boundary condition on the first interface is supplied by Thwaites' method for the laminar region. On the second interface, a zero Neumann boundary condition is applied, which in Continuous Galerkin is simply implemented by not actively applying a boundary condition, i.e. 'doing nothing'. In Discontinuous Galerkin, the implementation would be somewhat more involved. Keeping the above in mind, the choice is made to spatially discretize the turbulent boundary layer according to the Continuous Galerkin scheme. However, as only one element is considered and as such, there are no interfaces between elements to force continuity on, a more appropriate nomenclature is higher-order Galerkin, which will be used henceforth.

The spatial discretization of a one-dimensional variable, in this case velocity, of arbitrary order P, is written as the summation of a set of solution coefficients, denoted by $\hat{}$, multiplied with the corresponding function from the set of basis functions,

$$u(x) \approx \sum_{j=0}^{P} \hat{u}_j \phi_j(\xi), \tag{4.1.1}$$

where $\hat{u}_j$ indicates the jth solution coefficient, and ϕ_j the corresponding basis function in the standard domain Ω_{st} that ranges from $[-1, 1]$, denoted by the coordinate ξ. This will be explained in more detail further in this section.

Examples of basis functions that are commonly used in higher-order Galerkin methods, are Lagrange polynomials [51] and Legendre polynomials [52]. The principle difference between these two polynomials is that the former is nodal, whereas the latter is modal. This will be explained in further detail in section 4.2. In the present study, for each instance, Lagrange polynomials are used as the set of basis functions, as the polynomials as well as their derivatives are relatively simple to implement, and easy to understand. Lagrange polynomials are defined for a set of user-specified nodes on the standard domain Ω_{st}, and have the property of the Kronecker delta, namely that they are 1 in their corresponding node, and zero in the other nodes,

$$\delta_{i,j} = \begin{cases} 0 \text{ if } i \neq j, \\ 1 \text{ if } i = j. \end{cases} \tag{4.1.2}$$

The mathematical formulation [26][51] for the ith Lagrange polynomial is given by

$$\ell_i(\xi) = \left[\prod_{j=0,\dots,P}^{j \neq i} (\xi_i - \xi_j) \right]^{-1} \prod_{j=0,\dots,P}^{j \neq i} (\xi - \xi_j), \quad i = 0, \dots, P. \tag{4.1.3}$$

Note that in Equation 4.1.3, the rightmost term indicates the polynomial term with standard variable ξ, whereas the first term is a normalizing factor to ensure unit value at the specific node for which the Lagrange polynomial was designed, $\ell_i(\xi_i) = 1$.

A visual example of Lagrange polynomials is shown in Figure 4.1. Three nodes are defined,

namely $[-1, 0, 1]$. Using these nodes in Equation 4.1.3, the following basis is obtained,

$$\ell_i(\xi) = \left\{ \begin{array}{l} \frac{1}{(-1-0)(-1-1)}(\xi - 0)(\xi - 1) \\ \frac{1}{(0--1)(0-1)}(\xi - -1)(\xi - 1) \\ \frac{1}{(1-0)(1--1)}(\xi - -1)(\xi - 0) \end{array} \right\} = \left\{ \begin{array}{l} \frac{1}{2}(\xi^2 - \xi) \\ -(\xi^2 - 1) \\ \frac{1}{2}(\xi^2 + \xi) \end{array} \right\}, \quad i = 1, 2, 3. \tag{4.1.4}$$

The order of the polynomial is by definition always equal to the amount of nodes minus 1, hence in this specific case, the Lagrange polynomials are quadratic, as can be seen in Equation 4.1.4. From Figure 4.1, the Kronecker delta property can be seen clearly; three polynomials are defined that have unit value at their corresponding node, and are zero at the other nodes. This is also confirmed by plugging the nodes $[-1, 0, 1]$ in the functions in Equation 4.1.4. The nodes were linearly spaced as a means of clear illustration, but it is emphasized that the nodes can be specified simply by the user.

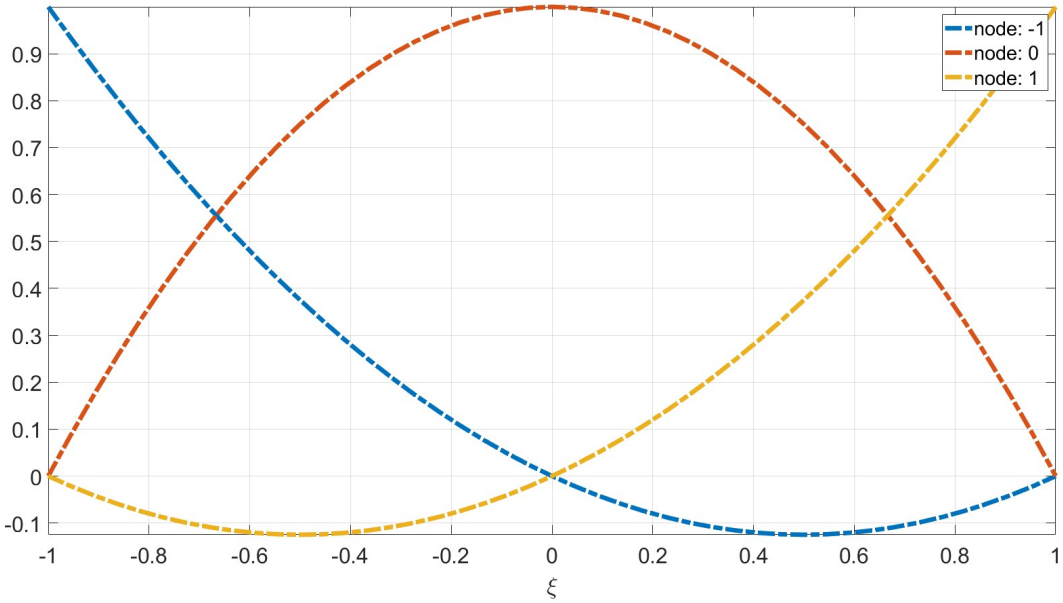

Figure 4.1: Three quadratic Lagrange polynomials.

Referring back to Galerkin methods, the manner in which the Galerkin projection is done, is similar to finite element formulations, namely by taking the PDE in its strong form, for example the linear one-dimensional advection-diffusion equation,

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \mu\frac{\partial^2 u}{\partial x^2}, \tag{4.1.5}$$

and taking it in its weak form on a domain Ω, using the same set of basis functions ϕ_i (the Lagrange polynomials) that will be used to spatially discretize the solution,

$$\int_\Omega \phi_i \frac{\partial u}{\partial t}\mathrm{d}x + \int_\Omega \phi_i\, a\frac{\partial u}{\partial x}\mathrm{d}x = \int_\Omega \phi_i\, \mu\frac{\partial^2 u}{\partial x^2}\mathrm{d}x, \qquad \text{for } i = 0, 1, ..., P. \tag{4.1.6}$$

Analytical solutions to a particular PDE have to satisfy certain smoothness conditions, whereas these conditions are somewhat loosened in the weak formulation, as it satisfies the PDE in

a 'weaker' way. Now, the second-order derivative can be split through integration by parts, reducing the degree of smoothness that the candidate solution has to satisfy,

$$\int_\Omega \phi_i \frac{\partial^2 u}{\partial x^2}\,\mathrm{d}x = \left[\phi_i \frac{\partial u}{\partial x}\right]_\ominus^\oplus - \int_\Omega \frac{\partial \phi_i}{\partial x}\frac{\partial u}{\partial x}\,\mathrm{d}x, \tag{4.1.7}$$

where $\ominus$ and $\oplus$ denote the values at the borders of the domain. The Lagrange polynomial is used as the set of basis functions, and therefore, due to its Kronecker delta property, the boundary term, i.e. the first term on the right-hand side of Equation 4.1.7 simplifies considerably, as only ϕ_0 and ϕ_P are non-zero at the element boundaries. These are thus dictated by either Dirichlet or Neumann boundary conditions and are obtained as input values, known beforehand, for the right-hand side of Equation 4.1.7. The domain Ω is taken as one element in the Galerkin formulation, and the projection of the velocity u, Equation 4.1.1, is substituted, yielding the following discretized set of equations,

$$\begin{aligned} &\int_\Omega \phi_i \left(\sum_{j=0}^{P} \frac{\mathrm{d}\hat{u}_j}{\mathrm{d}t}\phi_j\right)\mathrm{d}x + a\int_\Omega \phi_i\left(\sum_{j=0}^{P}\frac{\mathrm{d}\phi_j}{\mathrm{d}x}\hat{u}_j\right)\mathrm{d}x = ... \\ ... \;&\mu\left(\left[\phi_i\frac{\partial u}{\partial x}\right]_\ominus^\oplus - \int_\Omega \frac{\partial\phi_i}{\partial x}\left(\sum_{j=0}^{P}\frac{\partial\phi_j}{\partial x}\hat{u}_j\right)\mathrm{d}x\right), \qquad \text{for } i = 0, 1, ..., P, \end{aligned} \tag{4.1.8}$$

where the projection constants $\hat{u}_j$ are a function of time only, $\hat{u}_j(t)$, and the boundary terms $\left[\phi_i\frac{\partial u}{\partial x}\right]_\ominus^\oplus$ disappear through Dirichlet boundary conditions, or are known from Neumann boundary conditions. In the case of a zero Neumann boundary condition, the first term on the right-hand side of Equation 4.1.8 disappears altogether.

The integrals are evaluated using the commonly used Gauss-type quadratures [53], a way of approximating integrals by summing weight values multiplied with the function to be integrated, probed at specific quadrature nodes. These nodes are specified as the roots of Jacobi polynomials [53]. Quadrature will be explained in more detail in section 4.2. As these quadratures are applied in the standard coordinates in Galerkin methods, ranging from $[-1, 1]$, the differentials and integrals must be transformed using the Jacobian factor of transformation,

$$J = \frac{\mathrm{d}x}{\mathrm{d}\xi} = \frac{x^\oplus - x^\ominus}{2}, \tag{4.1.9}$$

where ξ denotes the 'standard' coordinate, and $[x^\ominus, x^\oplus]$ denote the boundaries of the domain. Equation 4.1.8 is transformed to standard coordinates, giving

$$\begin{aligned} &\int_{-1}^{1} \phi_i \left(\sum_{j=0}^{P} \frac{\mathrm{d}\hat{u}_j}{\mathrm{d}t}\phi_j\right)\frac{\mathrm{d}x}{\mathrm{d}\xi}\mathrm{d}\xi + a\int_{-1}^{1} \phi_i\left(\sum_{j=0}^{P}\frac{\mathrm{d}\phi_j}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x}\hat{u}_j\right)\frac{\mathrm{d}x}{\mathrm{d}\xi}\mathrm{d}\xi = ... \\ ... \;&\mu\left[\phi_i\frac{\partial u}{\partial x}\right]_\ominus^\oplus - \mu\int_{-1}^{1}\frac{\mathrm{d}\phi_i}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x}\left(\sum_{j=0}^{P}\frac{\mathrm{d}\phi_j}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{\mathrm{d}x}\hat{u}_j\right)\frac{\mathrm{d}x}{\mathrm{d}\xi}\mathrm{d}\xi, \qquad \text{for } i = 0, 1, ..., P. \end{aligned} \tag{4.1.10}$$

Substituting the Jacobian and matrix identities for the mass matrix, advection matrix, and

diffusion matrix, a system of equations in ODE can be written in matrix form,

$$J\sum_{j=0}^{P}\mathbf{M}_{ij}\frac{\mathrm{d}\hat{u}_j}{\mathrm{d}t}+a\sum_{j=0}^{P}\mathbf{A}_{ij}\hat{u}_j=\mu\left[\phi_i\frac{\partial u}{\partial x}\right]_{\ominus}^{\oplus}-\frac{\mu}{J}\sum_{j=0}^{P}\mathbf{D}_{ij}\hat{u}_j,\qquad\text{for } i=0,1,...,P, \tag{4.1.11}$$

with

$$\mathbf{M}_{ij}=\int_{-1}^{1}\phi_i\phi_j\,\mathrm{d}\xi,\qquad\mathbf{A}_{ij}=\int_{-1}^{1}\phi_i\phi_j'\,\mathrm{d}\xi,\qquad\mathbf{D}_{ij}=\int_{-1}^{1}\phi_i'\phi_j'\,\mathrm{d}\xi. \tag{4.1.12}$$

Lastly, vectors $\hat{\mathbf{u}}$ and $\boldsymbol{\phi}$ for $\hat{u}_j$ and ϕ_i are substituted to complete the formulation,

$$J\,\mathbf{M}\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{u}}+a\,\mathbf{A}\hat{\mathbf{u}}=\mu\left[\boldsymbol{\phi}\frac{\partial u}{\partial x}\right]_{\ominus}^{\oplus}-\frac{\mu}{J}\mathbf{D}\hat{\mathbf{u}}, \tag{4.1.13}$$

with

$$\hat{\mathbf{u}}=\{u_0,...,u_P\}^T,\qquad\boldsymbol{\phi}=\{\phi_0,...,\phi_P\}^T. \tag{4.1.14}$$

Terms that are known from either Dirichlet or Neumann boundary conditions are moved to the right-hand side. A spatially discretized system of equations has now been created, that contains only a time-dependent term. Hence, one is left with a set of ordinary differential equations, such that an ODE solver, for example ODE45 from MATLAB, is able to march the system forward in time. In this manner, a PDE with first- and second-order spatial derivatives can be transformed to an ODE that is time-dependent only, and can produce results within desired accuracy by increasing order P.

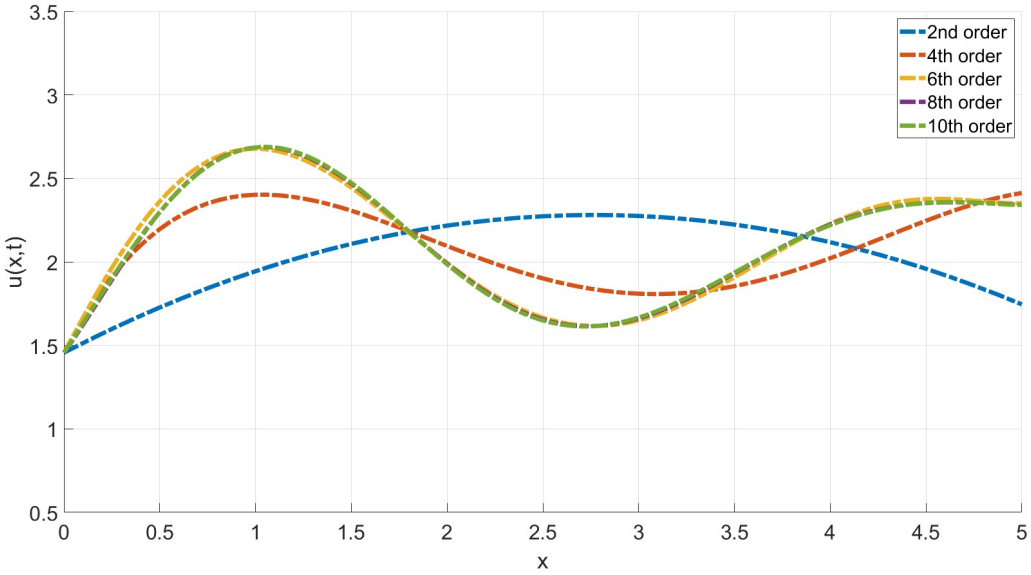

Figure 4.2: Solutions for the advection-diffusion equation, modeled at different orders of accuracy.

To show the functionality of the higher-order Galerkin scheme, a test case of the linear, unsteady, one-dimensional advection-diffusion equation, as given in Equation 4.1.13, is run, with

the Lagrange polynomials being the set of basis functions. For an advection speed of $a = 1$, and a viscosity of $\mu = 0.1$, a one-element solution of various orders of accuracy is seen in Figure 4.2. A sinusoidal function is used as Dirichlet boundary condition on the left border of the domain, and on the right border, by 'doing nothing', a Neumann boundary condition is applied of $\frac{\partial u}{\partial x} = 0$. Note that in Figure 4.2, the solutions of order 8 and 10 are on top of each other, indicating that the solution has already converged.

It is further emphasized that the solutions in Figure 4.2 consist of the summations of coefficients multiplied with their respective Lagrange polynomial. As such, the solution in its entirety is in fact a polynomial of order P. Knowing the set of solution coefficients $\hat{u}_j$, one can obtain the solution at any given x coordinate. Firstly, the coordinate for which one wants to know the solution needs to be mapped linearly to the standard domain through the following relation [26],

$$\xi = \frac{2x - x^{\oplus} - x^{\ominus}}{x^{\oplus} - x^{\ominus}}, \quad x \in \Omega = [x^{\oplus}, x^{\ominus}]. \tag{4.1.15}$$

Inversely, one can calculate a corresponding x coordinate for any given ξ through

$$x = \left(\frac{1-\xi}{2}\right) x^{\ominus} + \left(\frac{1+\xi}{2}\right) x^{\oplus}, \quad \xi \in \Omega_{st} = [-1, 1]. \tag{4.1.16}$$

Now, let the coordinate at which the solution is to be obtained, be denoted by $x_?$. The solution is now found by reconstructing the location of $x_?$ through linear mapping to ξ, into the standard domain $[-1, 1]$, using Equation 4.1.15. Once $\xi_?$ is known, the solution is simply found through interpolation of the summated Lagrange polynomial,

$$u(x_?) \approx \sum_{j=0}^{P} \hat{u}_j \ell_j(\xi_?). \tag{4.1.17}$$

Lastly, a solution of order P produced by higher-order Galerkin has an order of accuracy of $P + 1$ [25]. In reality, Galerkin spectral element methods have two manners in which they can converge. One can increase the amount of elements, thereby decreasing the mesh size h, or one can increase the polynomial order of approximation P. As such, an error convergence (and thus implicitly the order of accuracy) can be established as

$$\epsilon \propto h^{P+1}. \tag{4.1.18}$$

This can best be visualized in the residual of the solutions. A solution of order 50 is generated as a reference solution, with which the solutions from Figure 4.2 are compared, according to the L2 norm,

$$\text{Residual} = \sqrt{\sum (u_{\text{sol}} - u_{\text{num}})^2}, \tag{4.1.19}$$

where u_{sol} is the reference solution of order 50, and u_{num} denotes the numerical solutions of lower order in Figure 4.2. The produced result can be seen in Figure 4.3. Note that Figure 4.3 shows a logarithmic y axis and a linear x axis, i.e., 'exponential' convergence with linearly increasing polynomial order; by slightly increasing the number of nodes (or degrees of freedom), and with that, the polynomial order of the approximating solution, one can obtain an exponential

convergence to a solution, showing the potential that higher-order Galerkin methods have in numerical analysis.

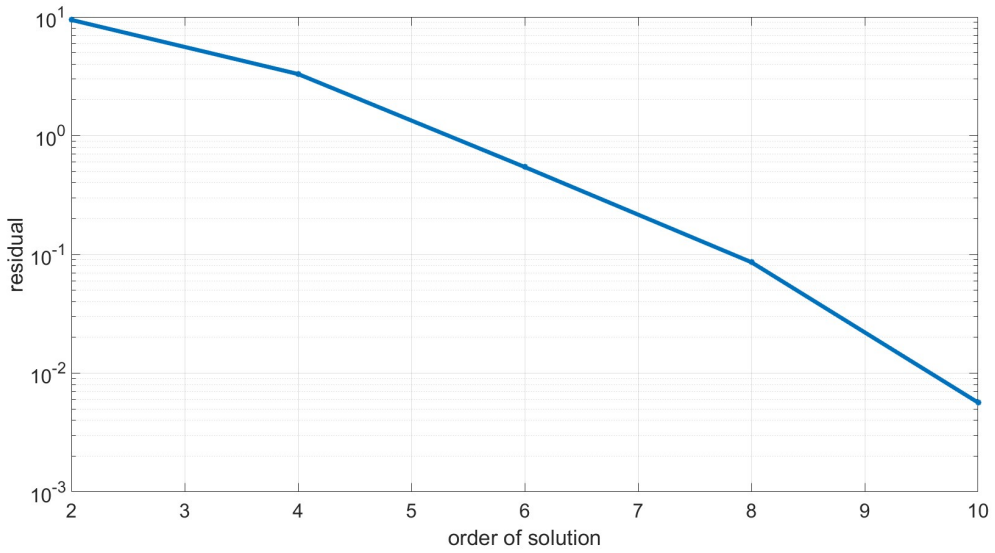

Figure 4.3: Residual for linear advection-diffusion equation.

Having explained the theory of higher-order Galerkin methods, it will now be applied to the governing system of integral boundary layer equations from section 3.3, in the next section, section 4.2.

4.2 Nodal Galerkin on the system of equations

Depending on the basis function, a Galerkin method is applied in a nodal or modal manner [25]. Referring back to section 4.1, Lagrange and Legendre polynomials are mentioned. The former is nodal [51], where a set of nodes defines the Lagrange polynomials that serve as basis. All polynomials are of the same degree P. The latter, the Legendre basis, is modal [52], where the set of basis functions is hierarchical, i.e., lower order sets are contained by higher-order sets, such that the order of the set is 'increasing', as it were, instead of all the functions having the same order such as in nodal formulations. As mentioned before, a nodal set of Lagrange polynomials is employed as basis in the present study, and as such, a nodal Galerkin scheme is applied to the system of equations.

Starting from the governing matrix system of Equation 3.3.3, the weak form is taken. For the sake of completeness, the system of equations is repeated here,

$$\begin{bmatrix} 1 & 0 \\ H_1 & \theta\frac{\mathrm{d}H_1}{\mathrm{d}H} \end{bmatrix} \frac{\mathrm{d}}{\mathrm{d}x} \begin{Bmatrix} \theta \\ H \end{Bmatrix} = \begin{pmatrix} \frac{1}{2}c_f - \frac{\theta}{u_e}(2+H)\frac{\mathrm{d}u_e}{\mathrm{d}x} \\ C_E - \frac{H_1\theta}{u_e}\frac{\mathrm{d}u_e}{\mathrm{d}x} \end{pmatrix}. \tag{4.2.1}$$

As mentioned in section 4.1, Lagrange polynomials are chosen as an appropriate set of nodal basis functions ϕ. They will be denoted as ℓ_i for $i = 0, 1, ..., P$ for the weak form of the equations, and ℓ_j for $j = 0, 1, ..., P$ for the expanded projections, respectively. As such, the solution for the variables θ and H will be expanded in this polynomial basis, denoting the variables as solution

coefficients $\hat{\theta}_j$ and $\hat{H}_j$, for $j = 0, 1, ..., P$, similar to the discretization of the advection-diffusion equation in section 4.1. The Galerkin projections will take on the form

$$\theta(x) \approx \sum_{j=0}^{P} \hat{\theta}_j \ell_j(\xi) \quad \text{and} \quad H(x) \approx \sum_{j=0}^{P} \hat{H}_j \ell_j(\xi). \tag{4.2.2}$$

Taking Equation 4.2.1, the term $\frac{1}{u_e}\frac{\mathrm{d}u_e}{\mathrm{d}x}$ can be merged together using the chain rule.

$$\begin{aligned} \text{Let } g(x) &= u_e(x), \\ \text{and } f(g(x)) &= \ln u_e, \\ \text{such that } \frac{\mathrm{d}f(g(x))}{\mathrm{d}x} &= \frac{\mathrm{d}f}{\mathrm{d}g}\frac{\mathrm{d}g}{\mathrm{d}x} \quad \rightarrow \quad \frac{\mathrm{d}\ln u_e}{\mathrm{d}x} = \frac{1}{u_e}\frac{\mathrm{d}u_e}{\mathrm{d}x}, \end{aligned}$$

and thus, the term $\frac{1}{u_e}\frac{\mathrm{d}u_e}{\mathrm{d}x}$ becomes $\frac{\mathrm{d}}{\mathrm{d}x}\ln u_e$. The weak forms on domain Ω are, respectively,

$$\int_\Omega \frac{\mathrm{d}\theta}{\mathrm{d}x}\ell_i \,\mathrm{d}x = \int_\Omega \frac{1}{2}c_f \,\ell_i \,\mathrm{d}x - \int_\Omega \theta(2+H)\frac{\mathrm{d}\ln u_e}{\mathrm{d}x}\ell_i \,\mathrm{d}x, \tag{4.2.3}$$

$$\int_\Omega H_1\frac{\mathrm{d}\theta}{\mathrm{d}x}\ell_i \,\mathrm{dx} + \int_\Omega \theta\frac{\mathrm{d}H_1}{\mathrm{d}H}\frac{\mathrm{d}H}{\mathrm{d}x}\ell_i \,\mathrm{dx} = \int_\Omega C_E \,\ell_i \,\mathrm{d}x - \int_\Omega H_1\theta\frac{\mathrm{d}\ln u_e}{\mathrm{d}x}\ell_i \,\mathrm{d}x. \tag{4.2.4}$$

Individual terms are each mapped linearly to the standard domain Ω_{st}, using coordinate ξ. Furthermore, the Galerkin projections $\theta \approx \sum_j \hat{\theta}_j\ell_j$ and $H \approx \sum_j \hat{H}_j\ell_j$ are substituted, where the integrals are evaluated using collocation of the Gauss-Lobatto nodes, stemming from the rule of Gauss-Lobatto quadrature [54], which concentrates nodes near the edges and includes the borders of the domain itself. An alternative commonly used Gaussian quadrature (more information on quadrature can be found in [55]) is Gauss-Legendre [56], but this technique does not include the borders. As a Dirichlet condition will be applied on the left border of the domain, provided by Thwaites' calculation of the laminar boundary layer until transition, the borders need to be explicitly included within the quadrature. As such, following [25] and [54], Gauss-Lobatto quadrature approximates an integral in the following manner,

$$\int_{-1}^{1} u(\xi)\,\mathrm{d}\xi \approx \sum_{k=1}^{Q} w_k u(\xi_k), \tag{4.2.5}$$

where Q is the amount of quadrature nodes where the function is to be probed, ξ_k is the node to probe the function on, and w_k is the weight with which this value contributes to the approximation of the integral. The coordinates ξ_k are obtained by the roots of the Jacobi [53] polynomial $\mathcal{J}^{1,1}_{Q-2}(\xi)$ of degree $Q-2$, whereas the weights are given by $w_k = \frac{2}{Q(Q-1)}[\mathcal{J}^{0,0}_{Q-1}(\xi_k)]^{-2}$, following the derivation of [25]. In any case, the nodes of the Lagrange polynomials are chosen by design to correspond to the Lobatto nodes at which the quadrature is performed, such that contributions from other nodes are anulled due to the Kronecker delta property of Lagrange polynomials.

Implementing the quadrature in the first integral term of Equation 4.2.3 gives

$$\int \frac{\mathrm{d}\theta}{\mathrm{d}x}\ell_i \,\mathrm{d}x \approx \int \left(\sum_{j=0}^{P} \hat{\theta}_j \ell_j'\right) \ell_i \,\mathrm{d}\xi \approx \sum_{k=0}^{P} w_k \sum_j \hat{\theta}_j \ell_j'(\xi_k)\ell_i(\xi_k),$$
$$= \sum_{k=0}^{P} w_k \sum_j \hat{\theta}_j \ell_j'(\xi_k)\delta_{i,k} = w_i \sum_{j=0}^{P} \hat{\theta}_j \ell_j'(\xi_i).$$

Due to the collocated nature of the quadratures and the Kronecker delta property of Lagrangian polynomials, many weight terms drop, simplifying the equations and accelerating the process. The term w_i represents the weight associated with the quadrature node. The other terms are obtained similarly,

$$\int \frac{1}{2} c_f \ell_i \,\mathrm{dx} \approx \frac{1}{2} J w_i c_f(\hat{\theta}_i, \hat{H}_i),$$
$$\int \theta(2+H)\frac{\mathrm{d}\ln u_e}{\mathrm{d}x}\,\mathrm{d}x \approx w_i(2+\hat{H}_i)\hat{\theta}_i \left.\frac{\mathrm{d}\ln u_e}{\mathrm{d}\xi}\right|_{\xi_i},$$
$$\int H_1 \frac{\mathrm{d}\theta}{\mathrm{d}x}\ell_i \,\mathrm{d}x \approx w_i H_1(\hat{H}_i)\sum_{j=0}^{P}\hat{\theta}_j\ell_j'(\xi_i),$$
$$\int \theta \frac{\mathrm{d}H_1}{\mathrm{d}H}\frac{\mathrm{d}H}{\mathrm{d}x}\ell_i \,\mathrm{dx} \approx w_i\hat{\theta}_i H_1'(\hat{H}_i)\sum_{j=0}^{P}\hat{H}_j\ell_j'(\xi_i),$$
$$\int C_E\,\ell_i \,\mathrm{dx} \approx J w_i C_E(\hat{H}_i),$$
$$\int H_1\theta\frac{\mathrm{d}\ln u_e}{\mathrm{d}x}\ell_i \,\mathrm{d}x \approx w_i H_1(\hat{H}_i)\hat{\theta}_i \left.\frac{\mathrm{d}\ln u_e}{\mathrm{d}\xi}\right|_{\xi_i},$$

where the Jacobian J appears for terms that do not possess a spatial derivative. A close observation allows discarding the weights for each term, as these weights appear on both sides of Equation 4.2.3 and Equation 4.2.4. Lastly, the first equation $i = 0$ in the system can be discarded, as the first coefficients $\hat{\theta}_0$ and $\hat{H}_0$ are already known variables from Thwaites' method of analytically obtaining the boundary layer parameters in the laminar region; these values are used directly as a Dirichlet boundary condition for the first of the Gauss-Lobatto nodes, to guarantee continuity between the laminar and turbulent domains. On the right side of the border, by 'doing nothing', a zero Neumann boundary condition is applied. Hence, the system is diminished by one equation, yielding the following set,

$$\left.\begin{aligned}
\sum_{j=0}^{P}\hat{\theta}_j\ell_j'(\xi_i) &= \frac{1}{2}Jc_f(\hat{\theta}_i,\hat{H}_i) - (2+\hat{H}_i)\hat{\theta}_i\left.\frac{\mathrm{d}\ln u_e}{\mathrm{d}\xi}\right|_{\xi_i} \\
H_1(\hat{H}_i)\sum_{j=0}^{P}\hat{\theta}_j\ell_j'(\xi_i) + \hat{\theta}_i H_1'(\hat{H}_i)\sum_{j=0}^{P}\hat{H}_j\ell_j'(\xi_i) &= \ldots \\
\ldots = JC_E(\hat{H}_i) - H_1(\hat{H}_i)\hat{\theta}_i - \left.\frac{\mathrm{d}\ln u_e}{\mathrm{d}\xi}\right|_{\xi_i}
\end{aligned}\right\} \quad \text{for } i = 1, 2, \ldots, P, \qquad (4.2.6)$$

where $\hat{\theta}_0$ and $\hat{H}_0$ are thus known terms from Thwaites' method. Before this equation is solved numerically, the analytical model for the external velocity, described in section 3.6, should also be incorporated in the discretization.

4.2.1 Implementation of the external velocity model in the Galerkin scheme

Equation 3.6.10 is taken and repeated for the sake of completeness,

$$u(x) = u_0(x) + \frac{1}{\pi}\int_0^c \frac{\mathrm{d}(u\delta^*)}{\mathrm{d}\xi}\frac{\mathrm{d}\xi}{x-\xi}. \tag{4.2.7}$$

The integration variable ξ in Equation 4.2.7 is now rewritten to ζ to avoid ambiguity with the standard coordinate ξ in the higher-order Galerkin scheme. Moreover, Equation 4.2.7 is transformed to streamwise coordinates, with the domain ranging from the point where transition sets in (which will be the user-specified $s = 0.1$) until the trailing edge. Another change, as accuracy of the analytical model is not of first concern, is that the model can be simplified even further by replacing u in the integrand with V_∞, and taking it out of the integrand. Lastly, $u(s_i)$ is written as $u_e(s_i)$ to be consistent with Equation 4.2.6, giving

$$u_e(s_i) = u_0(s_i) + \frac{V_\infty}{\pi}\int_{0.1}^{TE} \frac{\mathrm{d}\delta^*}{\mathrm{d}\zeta}\frac{\mathrm{d}\zeta}{s_i-\zeta}, \tag{4.2.8}$$

where ζ is the continuous streamwise coordinate, s_i denotes the discretized streamwise coordinate on point i of the grid in the turbulent region, and u_0 is the external velocity supplied by the inviscid panel method solver. For the sake of consistency with the article of Veldman [18], a discrete form of the integral in Equation 4.2.8 is adopted, on a uniform grid of 120 points. The integral in Equation 4.2.8 is then split into two parts to work around the singularity in the denominator in the integrand, where the displacement thickness is linearly interpolated on the midpoints of the grid and summed. At the neighboring points of the singularity, a quadratic interpolation is performed to avoid the singularity itself, yielding

$$u_e(s_i) \approx u_0(s_i) - \frac{2hV_\infty}{\pi}\left.\frac{\mathrm{d}^2\delta^*}{\mathrm{d}\zeta^2}\right|_i + \frac{hV_\infty}{\pi}\sum_{j\neq i-i,i}\left.\frac{\mathrm{d}\delta^*}{\mathrm{d}\zeta}\right|_{j+\frac{1}{2}}\ln\left|\frac{i-j}{i-j-1}\right|, \tag{4.2.9}$$

where h denotes the mesh size of the uniform grid. Note that a first and second derivative of the displacement thickness with respect to the streamwise coordinate s is required (written as ζ in Equation 4.2.9). From the higher-order Galerkin model, an approximation of the momentum thickness θ and the shape factor H, both of order P are obtained. As the displacement thickness δ^* is given by the product of these two solutions, it can be expressed as a polynomial of order $2P$. To derive the displacement thickness δ^* from these variables, one simply multiplies,

$$\theta H = \delta^*, \tag{4.2.10}$$

such that an approximation of order $2P$ for the displacement thickness is found. The solution coefficients $\hat{\theta}_j$ and $\hat{H}_j$ are specifically known at the $P+1$ Lobatto nodes, and are used to

approximate the solution over the entire domain of the turbulent flow with a polynomial of order P,

$$\theta(s) \approx \sum_{j=0}^{P} \hat{\theta}_j \ell_j, \qquad H(s) \approx \sum_{j=0}^{P} \hat{H}_j \ell_j. \tag{4.2.11}$$

Through polynomial interpolation, the set of solution coefficients is found at $2P+1$ Lobatto nodes, needed to obtain the polynomial approximation of order $2P$ for δ^*,

$$\hat{\theta}_k = \sum_{j=0}^{P} \hat{\theta}_j \ell_j(\xi_k), \qquad \hat{H}_k = \sum_{j=0}^{P} \hat{H}_j \ell_j(\xi_k), \qquad \text{for } k = 1, 2, ..., 2P+1. \tag{4.2.12}$$

The set of $P+1$ solution coefficients has now gone to a set of $2P+1$ solution coefficients through polynomial interpolation. As the product of these sets is equal to the set of solution coefficients for the displacement thickness δ^*, the sets are multiplied,

$$\hat{\delta}^*_k = \hat{\theta}_k \hat{H}_k = \left(\sum_{j=0}^{P} \hat{\theta}_j \ell_j(\xi_k) \right) \left(\sum_{j=0}^{P} \hat{H}_j \ell_j(\xi_k) \right), \quad \text{for } k = 1, 2, ..., 2P+1. \tag{4.2.13}$$

Having obtained the full set of $2P+1$ solution coefficients for the displacement thickness δ^*, the solution is represented as the sum of the solution coefficients multiplied with their basis function, yielding a solution of polynomial order $2P$,

$$\delta^*(s) \approx \sum_{k=0}^{2P} \hat{\delta}^*_k \ell_k. \tag{4.2.14}$$

Summarizing the process, from the Galerkin projected definitions of the momentum thickness θ and the shape factor H, the displacement thickness δ^* was obtained,

$$\theta H \approx \left(\sum_{j=0}^{P} \hat{\theta}_j \ell_j \right) \left(\sum_{j=0}^{P} \hat{H}_j \ell_j \right) \rightarrow \sum_{k=0}^{2P} \hat{\delta}^*_k \ell_k \approx \delta^*(s). \tag{4.2.15}$$

From the definition Equation 4.2.14 for the displacement thickness, the derivatives needed for Equation 4.2.9 are obtained by taking the derivatives with respect to ξ denoted by the $'$,

$$\delta^{*\prime} \approx \sum_{k=0}^{2P} \hat{\delta}^*_k \ell'_k = \sum_{k=0}^{2P} \hat{\delta}^*_k \frac{\mathrm{d}\ell_k}{\mathrm{d}\xi}, \qquad \delta^{*\prime\prime} \approx \sum_{k=0}^{2P} \hat{\delta}^*_k \ell''_k = \sum_{k=0}^{2P} \hat{\delta}^*_k \frac{\mathrm{d}^2\ell_k}{\mathrm{d}\xi^2}. \tag{4.2.16}$$

Note, however, that the derivatives in Equation 4.2.16 are taken with respect to the standard coordinate ξ, whereas for Equation 4.2.9, they are needed with respect to the streamwise coordinate s.

A division by the Jacobian J solves this problem, such that

$$\begin{aligned}\frac{\mathrm{d}\delta^*}{\mathrm{d}s} &\approx \left(\sum_{k=0}^{2P} \hat{\delta}_k^* \ell_k'\right) \frac{\mathrm{d}\xi}{\mathrm{d}s} = \frac{1}{J}\left(\sum_{k=0}^{2P} \hat{\delta}_k^* \ell_k'\right), \\ \frac{\mathrm{d}^2\delta^*}{\mathrm{d}s^2} &\approx \left(\sum_{k=0}^{2P} \hat{\delta}_k^* \ell_k''\right) \frac{\mathrm{d}^2\xi}{\mathrm{d}s^2} = \frac{1}{J^2}\left(\sum_{k=0}^{2P} \hat{\delta}_k^* \ell_k''\right).\end{aligned} \tag{4.2.17}$$

Nodal polynomial formulations have been obtained in Equation 4.2.17, which will be used in the model described in Equation 4.2.9. To obtain the values of the first and second derivative of δ^* with respect to s on the coordinates of the uniform grid specifically, Equation 4.2.17 will be evaluated on the coordinates of the uniform grid, linearly mapped to ξ. Combining Equation 4.2.9, Equation 4.2.15 and Equation 4.2.17, the external velocity, discretized on a streamwise uniform grid of 120 points, is written as a function of the solution coefficients $\hat{\theta}_j$ and $\hat{H}_j$, namely

$$u_e(s_i, \delta^*) = u_e(s_i, \hat{\theta}_j, \hat{H}_j). \tag{4.2.18}$$

Note that with Equation 4.2.18, the value of the external velocity has been obtained on the uniform streamwise grid. One final step should be performed, such that the system of equations is complete; for Equation 4.2.6, the value $\frac{\mathrm{d}\ln u_e}{\mathrm{d}\xi}$ is needed, evaluated at ξ_i, namely $P+1$ Lobatto nodes. Having the solution of the external velocity at 120 points, they are linearly interpolated to the values at the original Gauss-Lobatto quadrature nodes to obtain a set of solution coefficients $\hat{u}_j$. It should be noted that this is a simple, first-order linear interpolation, and as such, the uniform mesh on which the velocity and displacement thickness are discretized cannot be too coarse. In the case of a polynomial solution of very high order, this mesh would need to be refined as well, to minimize the interpolation error when approximating the solution coefficients $\hat{u}_j$ of the external velocity.

In any case, the set of solution coefficients $\hat{u}_j$ is obtained, with which the entire solution is found through Lagrange interpolation, $u_e(s) \approx \sum_j \hat{u}_j \ell_j$, but it is not yet in the form in which it is needed in Equation 4.2.6. An intermediate variable v is introduced, taken as the natural logarithm of the set of solution coefficients $\hat{u}_j$, such that

$$\hat{v}_j = \ln \hat{u}_j \quad \rightarrow \quad \ln u_e(s) \approx \sum_j \hat{v}_j \ell_j. \tag{4.2.19}$$

Now, one last operation is performed for the term to be ready for Equation 4.2.6, which is to obtain the derivative probed at the nodes ξ_i,

$$\left.\frac{\mathrm{d}\ln u_e}{\mathrm{d}\xi}\right|_{\xi_i} = \frac{\mathrm{d}}{\mathrm{d}\xi}\sum \hat{v}_j \ell_j(\xi_i) = \sum \hat{v}_j \ell_j'(\xi_i), \quad \text{for } i = 1, 2, ..., P. \tag{4.2.20}$$

The last identity has now been derived, with which Equation 4.2.6 can be completely closed. The challenge that now remains, is how to solve this system of equations. In the next section, section 4.3, the numerical technique to solve the system in Equation 4.2.6 will be discussed.

4.3 Point-implicit iteration scheme

A point-implicit (simplified implicit Euler) method is employed [57][58] to find the solution of the nonlinear system of governing equations for the turbulent part of the boundary layer. Hence, the point-implicit method is employed only as a means to accelerate convergence to the steady-state solution of the governing equations within the boundary layer solver. The manner in which it works, is that it sweeps back and forth across the spatial nodes to find updates values, before proceeding to the next time step and repeating the spatial sweeps again. A visualization of the scheme is seen in Figure 4.4. It should be noted that there is a point 0, but this is the Dirichlet boundary condition, and hence, it is not included in the iteration scheme, as its value is fixed. The scheme commences at time step n at node 1, and makes a first prediction of a $\Delta\theta_1$ and ΔH_1, going to the green dot (θ_1^*, H_1^*). It starts to do spatial sweeps, thereby calculating and updating the changes $\Delta\theta$ and ΔH that will denote the difference between time step n and time step $n+1$. The changes are temporarily added to the existing values of θ and H, denoting the most recent value with an asterisk, (θ^*, H^*), indicated by the blue dots in Figure 4.4. In calculating the change of the next point in the forward sweep, the most recent values of the previous points are used. Arriving at node p, the system does a backward sweep, again always using the most recent point of each node, until it arrives at the first node again. Now $\Delta\theta$s and ΔHs are obtained for all points $1, 2, ..., p$, each of them updated twice (forward and then backward), using the most recent values (θ^*, H^*) in the scheme. It is noted that the spatial sweeps use these most recent values only to obtain updated $\Delta\theta_{1,2,...,p}$ and $\Delta H_{1,2,...,p}$; the sweeps are meant to converge between time step n and $n+1$.

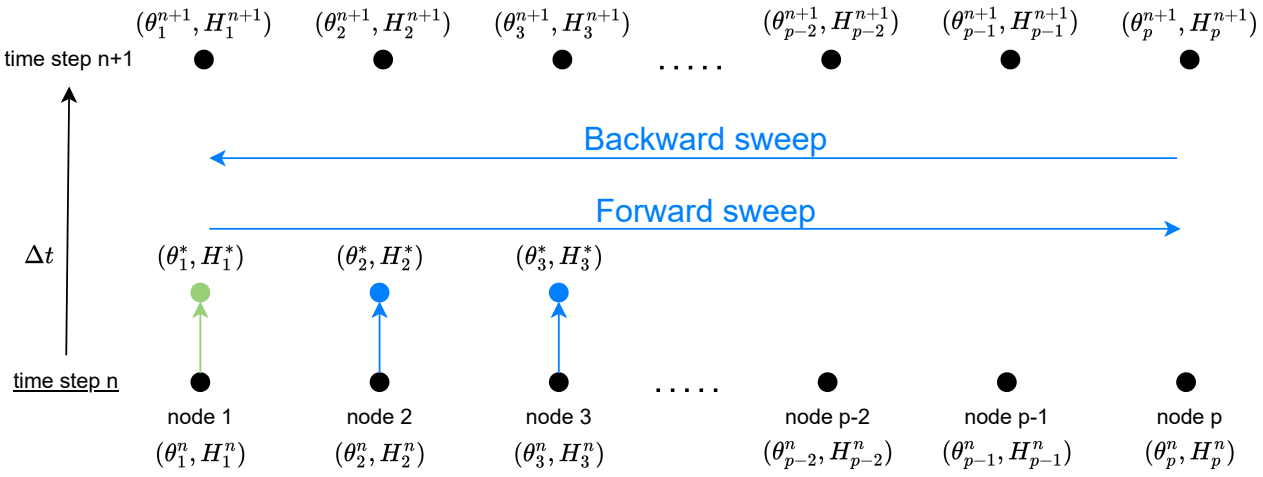

Figure 4.4: Point-implicit scheme at work.

The idea behind the point-implicit scheme is to introduce a time-dependency in the equations as a residue that tends to zero while iterating further, to retrieve the steady-state solution. Therefore, the time dependent terms on the left-hand side of the governing equations are not taken away, as they are needed for the implicit Euler scheme. The system of equations, Equations (6-7) from Veldman [18], is reintroduced,

$$\frac{H}{u_e}\frac{\partial\theta}{\partial t} + \frac{\theta}{u_e}\frac{\partial H}{\partial t} = -\frac{\partial\theta}{\partial x} + \frac{c_f}{2} - (2+H)\,\theta\,\frac{\mathrm{d}\ln u_e}{\mathrm{d}x} \equiv R_1(\theta, H), \tag{4.3.1}$$

$$\frac{H_1 + H}{u_e}\frac{\partial \theta}{\partial t} + (H_1' + 1)\frac{\theta}{u_e}\frac{\partial H}{\partial t} = -H_1\frac{\partial \theta}{\partial x} - H_1'\,\theta\frac{\partial H}{\partial x} + C_E - H_1\,\theta\frac{\mathrm{d}\ln u_e}{\mathrm{d}x} \equiv R_2(\theta, H). \quad (4.3.2)$$

The time dependent velocity term $\partial u/\partial t$ however, is not included in the calculation. A physical velocity changing with time is not considered, but rather, the time derivatives of θ and H are only introduced to march forward in a pseudo-time and do not necessarily have a physical meaning; they help converge to the steady-state of θ and H, for which $R_1(\theta, H) = 0$ and $R_2(\theta, H) = 0$. Hence, the time derivative of u is not considered a variable to be solved, and as such, it is not treated in the equations. In the spirit of pseudo-time derivatives, the left-hand side can actually be simplified by replacing the values of θ, H and u before the time derivatives, by constant values to once again aid convergence, such that

$$\frac{H^\star}{u^\star}\frac{\partial \theta}{\partial t} + \frac{\theta^\star}{u^\star}\frac{\partial H}{\partial t} = R_1(\theta, H), \quad (4.3.3)$$

$$\frac{H_1^\star + H^\star}{u^\star}\frac{\partial \theta}{\partial t} + (H_1'^\star + 1)\frac{\theta^\star}{u^\star}\frac{\partial H}{\partial t} = R_2(\theta, H)\ , \quad (4.3.4)$$

where the superscript * denotes the constants that help converge to the steady-state solutions. Given that the solution for the whole turbulent part of the boundary layer is expressed through a single polynomial based on Lagrange basis functions built upon Gauss-Lobatto nodes, namely,

$$\theta(x) = \sum_{j=0}^{P} \hat{\theta}_j\,\ell_j(\xi(x)) \quad \text{and} \quad H(x) = \sum_{j=0}^{P} \hat{H}_j\,\ell_j(\xi(x)), \quad (4.3.5)$$

the higher-order Galerkin method applied to Equation 4.3.3 and Equation 4.3.4 leads to

$$\frac{\hat{H}_i^\star}{\hat{u}_i^\star}\frac{\partial \hat{\theta}_i}{\partial t} + \frac{\hat{\theta}_i^\star}{\hat{u}_i^\star}\frac{\partial \hat{H}_i}{\partial t} = R_1^i(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{H}}), \quad (4.3.6)$$

$$\frac{H_1(\hat{H}_i^\star) + \hat{H}_i^\star}{\hat{u}_i^\star}\frac{\partial \hat{\theta}_i}{\partial t} + [H_1'(\hat{H}_i^\star) + 1]\frac{\hat{\theta}_i^\star}{\hat{u}_i^\star}\frac{\partial \hat{H}_i}{\partial t} = R_2^i(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{H}}). \quad (4.3.7)$$

In addition, $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{H}}$ denote the whole set of solution coefficients (or nodal solution values), expressed as column vectors, and

$$R_{1,2}^i(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{H}}) = \frac{1}{Jw_i}\int R_{1,2}(\theta, H)\,\ell_i(\xi)\,\mathrm{d}x, \quad (4.3.8)$$

or the weak formulation of the residual, for which collocated quadrature is assumed as per the strategy of higher-order Galerkin Spectral Element Methods, explained in section 4.2. In addition, w_i is the quadrature weight associated to node ξ_i of Gauss-Lobatto's rule and J is the mapping Jacobian factor (i.e. half of the turbulent section's length), as discussed in section 4.2.

Let the solution variation across a time step be denoted as $\Delta\theta = \theta^{n+1} - \theta^n$ and $\Delta H = H^{n+1} - H^n$, with n referring to the current time step level, then implicit Euler's method prescribes that

$$\frac{\hat{H}_i^\star}{\hat{u}_i^\star}\frac{\Delta \hat{\theta}_i}{\Delta t} + \frac{\hat{\theta}_i^\star}{\hat{u}_i^\star}\frac{\Delta \hat{H}_i}{\Delta t} = R_1^i(\hat{\boldsymbol{\theta}}^{n+1}, \hat{\boldsymbol{H}}^{n+1}), \quad (4.3.9)$$

$$\frac{H_1(\hat{H}_i^\star)+\hat{H}_i^\star}{\hat{u}_i^\star}\frac{\Delta\hat{\theta}_i}{\Delta t}+[H_1^{'}(\hat{H}_i^\star)+1]\frac{\hat{\theta}_i^\star}{\hat{u}_i^\star}\frac{\Delta\hat{H}_i}{\Delta t} \;=\; R_2^i(\hat{\boldsymbol{\theta}}^{n+1},\hat{\boldsymbol{H}}^{n+1}), \tag{4.3.10}$$

with the following approximation both for R_1^i and R_2^i,

$$R^i(\hat{\boldsymbol{\theta}}^{n+1},\hat{\boldsymbol{H}}^{n+1}) \;\approx\; R^i(\hat{\boldsymbol{\theta}}^{n},\hat{\boldsymbol{H}}^{n})+\frac{\partial R^i}{\partial\hat{\theta}_i}\Delta\hat{\theta}_i+\frac{\partial R^i}{\partial\hat{H}_i}\Delta\hat{H}_i+\sum_{j\neq i}\left(\frac{\partial R^i}{\partial\hat{\theta}_j}\Delta\hat{\theta}_j+\frac{\partial R^i}{\partial\hat{H}_j}\Delta\hat{H}_j\right), \tag{4.3.11}$$

where all partial derivatives are to be evaluated at time level n.

In the point-implicit method, iterative sweeps shall be used across solution points to update solution values until eventually reaching the solution at the next time level. Now, recognizing that

$$R^i(\hat{\boldsymbol{\theta}}^{n},\hat{\boldsymbol{H}}^{n})+\sum_{j\neq i}\left(\frac{\partial R^i}{\partial\hat{\theta}_j}\Delta\hat{\theta}_j+\frac{\partial R^i}{\partial\hat{H}_j}\Delta\hat{H}_j\right) \;\approx\; R^i(\hat{\boldsymbol{\theta}}^{\star},\hat{\boldsymbol{H}}^{\star}), \tag{4.3.12}$$

where the superscript * now denotes the most recent updated solution values (approximating values at time level $n+1$) for all points neighboring point i, the following iterative scheme is obtained,

$$\left(\frac{\hat{H}_i^\star}{\hat{u}_i^\star\,\Delta t}-\frac{\partial R_1^i}{\partial\hat{\theta}_i}\right)\Delta\hat{\theta}_i+\left(\frac{\hat{\theta}_i^\star}{\hat{u}_i^\star\,\Delta t}-\frac{\partial R_1^i}{\partial\hat{H}_i}\right)\Delta\hat{H}_i \;=\; R_1^i(\hat{\boldsymbol{\theta}}^{\star},\hat{\boldsymbol{H}}^{\star}), \tag{4.3.13}$$

$$\left(\frac{[H_1(\hat{H}_i^\star)+\hat{H}_i^\star]}{\hat{u}_i^\star\,\Delta t}-\frac{\partial R_2^i}{\partial\hat{\theta}_i}\right)\Delta\hat{\theta}_i+\left(\frac{[H_1'(\hat{H}_i^\star)+1]\,\hat{\theta}_i^\star}{\hat{u}_i^\star\,\Delta t}-\frac{\partial R_2^i}{\partial H_i}\right)\Delta\hat{H}_i \;=\; R_2^i(\hat{\boldsymbol{\theta}}^{\star},\hat{\boldsymbol{H}}^{\star}). \tag{4.3.14}$$

The above is a 2x2 linear system whose solution is trivial and yields $\Delta\theta_i = \theta_i^{n+1}-\theta_i^n$ and $\Delta H_i = H_i^{n+1}-H_i^n$ at any single solution point i.

The point-implicit scheme prescribes that one should update the preliminary solution of level $n+1$, namely the solution vector $(\hat{\boldsymbol{\theta}}^{\star},\hat{\boldsymbol{H}}^{\star})$, immediately as the variations $\Delta\theta_i$ and ΔH_i become available for each single point. This way, one can perform forward and backward sweeps across the points always using the most recent solution $(\hat{\boldsymbol{\theta}}^{\star},\hat{\boldsymbol{H}}^{\star})$ in the evaluation of the right-hand side of the above equations along the process – so that the estimated variation at any given point can benefit from the most recent updated solution from its neighbors.

Generally, the amount of spatial sweeps needed to converge all $\Delta\theta_{1,\dots,p}$ and $\Delta H_{1,\dots,p}$ to proceed from time step n to time step $n+1$ lies around two times forward and backward. After these four sweeps, $\Delta\theta$ and ΔH do not change significantly anymore, and additional sweeps are trivial. As such, one can then proceed to time step $n+1$ to repeat the cycle.

The numerical scheme behind solving the equations was explained in this section. In the next section, section 4.4, it will be discussed how one goes about stabilizing the numerics.

4.4 Spectral vanishing viscosity

Without element interfaces, both Discontinuous and Continuous Galerkin would have no, or very little diffusive properties. As such, it can be beneficial to introduce an artificial viscosity that can stabilize the scheme by diffusing energy from a selection of modes in the candidate solution. A technique that has recently taken a growth in Large Eddy Simulations is spectral

vanishing viscosity, or SVV. Originally introduced as a pseudo-viscosity term in one-dimensional hyperbolic problems [59], it has now been developed to serve for additional robustness in LES without deteriorating the accuracy of a numerical solution [60][61].

A higher-order Galerkin spatially discretized pure diffusion equation acts on the entire approximated solution and all its modes equally, as seen in Equation 4.1.13 and Equation 4.1.12, whereas spectral viscosity can be 'activated' from a specific, user-specified mode. The idea behind SVV is to ensure diffusivity that acts only on the higher modes of the solution, as these are the ones prone to cause instabilities and diverging simulations, but maintain the level of accuracy that higher-order Galerkin provides. Instead of the regular diffusive term

$$\mu \frac{\partial^2 u}{\partial x^2}, \quad \text{in the Galerkin scheme} \quad \frac{\mu}{J} \sum_j \mathbf{D}_{ij} \hat{u}_j,$$

a term is written with a type of stabilizing filter that acts on modes of higher order,

$$\mu_{svv} \frac{\partial}{\partial x} \left(\mathcal{Q} \star \frac{\partial u}{\partial x} \right). \tag{4.4.1}$$

In Equation 4.4.1, $\mathcal{Q}$ takes on the role of this filter, the so-called SVV kernel, and μ_{svv} is the artificial viscosity coefficient that dictates the overall magnitude of SVV. Now, following a similar derivation as the one in Equation 4.1.7,

$$\int_\Omega \phi_i \frac{\partial}{\partial x} \left(\mathcal{Q} \star \frac{\partial u}{\partial x} \right) \mathrm{d}x = \left[\phi_i \mathcal{Q} \star \frac{\partial u}{\partial x} \right]_\ominus^\oplus - \int_\Omega \frac{\partial \phi_i}{\partial x} \left(\mathcal{Q} \star \frac{\partial u}{\partial x} \right) \mathrm{d}x. \tag{4.4.2}$$

Generally, the boundary term $\left[\phi_i \mathcal{Q} \star \frac{\partial u}{\partial x}\right]_\ominus^\oplus$ is disregarded, and SVV is assumed to be active only within internal elements. Hence, in the latter term in Equation 4.4.2, the discrete projection of the velocity $u \approx \sum_j u_j \phi_j$ is substituted, giving

$$\int_\Omega \frac{\partial \phi_i}{\partial x} \left(\mathcal{Q} \star \frac{\partial u}{\partial x} \right) \mathrm{d}x = J^{-1} \int_{-1}^{1} \phi_i' \mathcal{Q} \star \left(\sum_j u_j \phi_j' \right) \mathrm{d}\xi. \tag{4.4.3}$$

Now, as the right-hand side of Equation 4.4.3 contains a derivative term in the projection, a rotation should be performed to find a set of coefficients that corresponds to the element-wise coefficients of $\frac{\partial u}{\partial \xi}$. This set $\hat{g}_j$ is found through a projection,

$$\int (\sum_{j=0}^{P} \hat{u}_j \phi_j') \phi_i \, \mathrm{d}\xi = \int (\sum_{j=0}^{P} \hat{g}_j \phi_j) \phi_i \, \mathrm{d}\xi, \qquad \text{for } i = 0, 1, ..., P, \tag{4.4.4}$$

and thus, looking back at Equation 4.1.12,

$$\mathbf{A}\hat{\mathbf{u}} = \mathbf{M}\hat{\mathbf{g}}, \quad \text{or} \quad \hat{\mathbf{g}} = \mathbf{M}^{-1}\mathbf{A}\hat{\mathbf{u}}. \tag{4.4.5}$$

Now, the coefficients $\hat{g}_j$ of the polynomial $\sum_j \hat{u}_j \phi_j'$ are known. Before the filter is applied, however, an additional change to a modal basis should be performed. The filtering function of

kernel $\mathcal{Q}$ is more naturally applied to a polynomial expansion that is written in a modal, or hierarchical basis, such that a clearer distinction between modes can be made. It is remembered that the basis used in the present study, is the nodal Lagrange basis, and as such, applying viscosity directly on this basis would cause interference in all modes. Hence, an additional transformation to a modal basis is required; specifically one with orthonormal properties, such that the SVV kernel $\mathcal{Q}$ is more easily specified as modal filter. As mentioned in section 4.2, a common modal set of basis functions is the Legendre polynomials [52], such that the nodal Lagrange basis is rotated to the modal Legendre basis with an additional Galerkin projection,

$$\int \left(\sum_{j=0}^{P} \hat{o}_j \varphi_j \right) \varphi_i \, \mathrm{d}\xi = \int \left(\sum_{j=0}^{P} \hat{g}_j \phi_j \right) \varphi_i \, \mathrm{d}\xi, \qquad \text{for } i = 0, 1, ..., P, \tag{4.4.6}$$

such that

$$\mathbf{I}\hat{\mathbf{o}} = \mathbf{R}\hat{\mathbf{g}}, \quad \text{with} \quad \mathbf{R}_{ij} = \int_{-1}^{1} \varphi_i \phi_j \, \mathrm{d}\xi. \tag{4.4.7}$$

In Equation 4.4.7, φ_i denotes the set of orthonormal Legendre polynomials. Due to the orthonormal properties, $\mathbf{I}$ denotes the identity matrix, and can be omitted for simplicity. As there is now an orthogonal set of coefficients $\hat{o}_j$, the SVV kernel $\mathcal{Q}$ can be defined through a diagonal matrix $\mathbf{Q}$ multiplying $\hat{o}_j$. Usually, a monotonically increasing kernel function is taken; a well-known example that reaches unit value as its last entry, is the exponential kernel, originally proposed in [59],

$$\mathbf{Q}_{ii} = \exp\left[-\frac{(i-P)^2}{(i-P_{svv})^2} \right], \quad \text{for } i > P_{svv}, \tag{4.4.8}$$

where P_{svv} is the order at which SVV is 'activated', and can be specified by the user. An adequate value of P_{svv} has been found to be $P/2$. Now, SVV is activated through the kernel in Equation 4.4.8, and should be transformed back into the original basis ϕ_i through $\mathbf{R}^{-1}$, obtaining a set of coefficients $\hat{\mathbf{q}}$ that has already been filtered,

$$\hat{\mathbf{q}} = \mathbf{R}^{-1}\mathbf{Q}\hat{\mathbf{o}} = \mathbf{R}^{-1}\mathbf{Q}\mathbf{R}\hat{\mathbf{g}} = \mathbf{R}^{-1}\mathbf{Q}\mathbf{R}\mathbf{M}^{-1}\mathbf{A}\hat{\mathbf{u}}. \tag{4.4.9}$$

All matrices have been defined such that $\mathcal{Q} \star (\sum_j \hat{u}_j \phi'_j) = \sum_j \hat{q}_j \phi_j$. Now, Equation 4.4.3 is rewritten,

$$J^{-1} \int_{-1}^{1} \phi'_i \mathcal{Q} \star \left(\sum_{j=0}^{P} u_j \phi'_j \right) \mathrm{d}\xi = J^{-1} \int_{-1}^{1} \phi'_i (\sum_{j=0}^{P} \hat{q}_j \phi_j) = \sum_{j=0}^{P} \hat{q}_j \mathbf{A}_{ji} = \sum_{j=0}^{P} \mathbf{A}^T_{ij} \hat{q}_j, \tag{4.4.10}$$

such that, combining Equation 4.4.1 and Equation 4.4.10,

$$\frac{\mu_{svv}}{J}\mathbf{A}^T\hat{\mathbf{q}} = \frac{\mu_{svv}}{J}\mathbf{A}^T\mathbf{R}^{-1}\mathbf{Q}\mathbf{R}\mathbf{M}^{-1}\mathbf{A}\hat{\mathbf{u}} = \frac{\mu_{svv}}{J}\mathbf{H}\hat{\mathbf{u}}. \tag{4.4.11}$$

In Equation 4.4.11, $\mathbf{H} = \mathbf{A}^T\mathbf{R}^{-1}\mathbf{Q}\mathbf{R}\mathbf{M}^{-1}\mathbf{A}$, denoting the artificial diffusion term that, in the pure case, is denoted by $\mathbf{D}$. Moreover, if the diagonal matrix $\mathbf{Q}$ becomes the identity matrix, the pure diffusion matrix $\mathbf{D}$ is actually obtained for $\mathbf{H}$, showing the similarity to standard diffusion.

4.4.1 SVV in the system of equations

The pure diffusion equation in its simplest form takes on the following shape,

$$\frac{\partial u}{\partial t} = \mu \frac{\partial^2 u}{\partial x^2}. \tag{4.4.12}$$

It should be noted that this is only one equation to solve one variable. Preferably, the SVV terms would be implemented in its simplest form, being

$$\begin{aligned} \frac{\partial \theta}{\partial t} &= ... + SVV_\theta, \\ \frac{\partial H}{\partial t} &= ... + SVV_H, \end{aligned} \tag{4.4.13}$$

however, referring back to section 4.3, more specifically Equation 4.3.1 and Equation 4.3.2, the system of equations contains coupled time-derivatives of both variables that appear in both equations. Hence, the SVV terms should be introduced accordingly, to obey the diffusion mechanism in a mathematical manner.

Let R_1 and R_2 once again be defined,

$$\begin{aligned} R_1(\theta, H) &\equiv -\frac{\partial \theta}{\partial x} + \frac{c_f}{2} - (2 + H)\,\theta\,\frac{\mathrm{d} \ln u_e}{\mathrm{d}x}, \\ R_2(\theta, H) &\equiv -H_1 \frac{\partial \theta}{\partial x} - H_1^{'}\,\theta\,\frac{\partial H}{\partial x} + C_E - H_1\,\theta\,\frac{\mathrm{d} \ln u_e}{\mathrm{d}x}, \end{aligned} \tag{4.4.14}$$

such that

$$\begin{aligned} \frac{H}{u_e}\frac{\partial \theta}{\partial t} + \frac{\theta}{u_e}\frac{\partial H}{\partial t} &= R_1(\theta, H), \\ \frac{H_1 + H}{u_e}\frac{\partial \theta}{\partial t} + (H_1' + 1)\frac{\theta}{u_e}\frac{\partial H}{\partial t} &= R_2(\theta, H). \end{aligned} \tag{4.4.15}$$

Now, Equation 4.4.13 shows in essence how the system operates with a single viscosity term. However, given that the system is coupled, the same coupling should be applied on the single viscosity terms to satisfy the diffusion mechanics. Let Equation 4.4.15 be rewritten in matrix form,

$$\begin{bmatrix} \frac{H}{u_e} & \frac{\theta}{u_e} \\ \frac{H_1+H}{u_e} & (H_1' + 1)\frac{\theta}{u_e} \end{bmatrix} \begin{Bmatrix} \frac{\partial \theta}{\partial t} \\ \frac{\partial H}{\partial t} \end{Bmatrix} = \begin{Bmatrix} R_1(\theta, H) \\ R_2(\theta, H) \end{Bmatrix}, \tag{4.4.16}$$

such that the matrix is obtained that prescribes the coupling of the time-derivatives in the system of equations. This system matrix is multiplied with Equation 4.4.13 to obtain a correct implementation of the SVV terms, yielding

$$\begin{bmatrix} \frac{H}{u_e} & \frac{\theta}{u_e} \\ \frac{H_1+H}{u_e} & (H_1' + 1)\frac{\theta}{u_e} \end{bmatrix} \begin{Bmatrix} SVV_\theta \\ SVV_H \end{Bmatrix} = \begin{Bmatrix} \frac{H}{u_e} SVV_\theta + \frac{\theta}{u_e} SVV_H \\ \frac{H_1+H}{u_e} SVV_\theta + (H_1' + 1)\frac{\theta}{u_e} SVV_H. \end{Bmatrix}. \tag{4.4.17}$$

Hence, the total system of equations can be set up,

$$\begin{aligned} \frac{H}{u_e}\frac{\partial\theta}{\partial t} + \frac{\theta}{u_e}\frac{\partial H}{\partial t} &= R_1(\theta,H) + \frac{H}{u_e}SVV_\theta + \frac{\theta}{u_e}SVV_H, \\ \frac{H_1+H}{u_e}\frac{\partial\theta}{\partial t} + (H_1'+1)\frac{\theta}{u_e}\frac{\partial H}{\partial t} &= R_2(\theta,H) + \frac{H_1+H}{u_e}SVV_\theta + (H_1'+1)\frac{\theta}{u_e}SVV_H. \end{aligned} \tag{4.4.18}$$

Now let SVV_θ and SVV_H be defined, following section 4.4's derivation,

$$\begin{aligned} SVV_\theta &= \frac{\mu_\theta}{J}\mathbf{A}^T\mathbf{R}^{-1}\mathbf{Q}\mathbf{R}\mathbf{M}^{-1}\mathbf{A}\hat{\boldsymbol{\theta}} = \frac{\mu_\theta}{J}\mathbf{F}\hat{\boldsymbol{\theta}}, \\ SVV_H &= \frac{\mu_H}{J}\mathbf{A}^T\mathbf{R}^{-1}\mathbf{Q}\mathbf{R}\mathbf{M}^{-1}\mathbf{A}\hat{\boldsymbol{H}} = \frac{\mu_H}{J}\mathbf{F}\hat{\boldsymbol{H}}, \end{aligned} \tag{4.4.19}$$

where μ_θ and μ_H are separate artificial viscosity coefficients that can be defined individually. Moreover, let the matrices $\mathbf{A}$, $\mathbf{M}$, $\mathbf{R}$ and $\mathbf{Q}$ from Equation 4.4.19 be taken from section 4.4. To avoid confusion with the shape factor variable $\hat{\boldsymbol{H}}$, matrix $\mathbf{H}$ from section 4.4 is substituted with $\mathbf{F}$. Matrices $\mathbf{A}$ and $\mathbf{M}$ are calculated using the Lagrange polynomials and its derivatives only, whereas $\mathbf{R}$ is calculated using Lagrange and Legendre polynomials. Finally, $\mathbf{Q}$ is taken as diagonal matrix where the values start monotonically increasing to unit value, starting from P_{svv}, where the latter is taken to be $P_{svv} = round(\frac{1}{2}P)$.

Lastly, let the spectral vanishing viscosity terms be included (with a minus sign coming from the integration by parts in Equation 4.4.2) in the complete discretized set of equations that is iteratively solved using the point-implicit scheme, defined as residuals

$$\begin{aligned} R_1\left(\hat{\boldsymbol{\theta}}^*, \hat{\boldsymbol{H}}^*\right) \equiv\ & -\sum_{j=0}^{P}\hat{\theta}_j\ell_j'(\xi_i) + \frac{1}{2}Jc_f(\hat{\theta}_i,\hat{H}_i) - (2+\hat{H}_i)\hat{\theta}_i\left.\frac{\mathrm{d}\ln u_e}{\mathrm{d}\xi}\right|_{\xi_i} - \ldots \\ & \ldots - \frac{\hat{H}_i}{u_e|_{\xi_i}}\frac{\mu_\theta}{J}\sum_{j=0}^{P}\mathbf{F}_{ij}\hat{\theta}_j - \frac{\hat{\theta}_i}{u_e|_{\xi_i}}\frac{\mu_H}{J}\sum_{j=0}^{P}\mathbf{F}_{ij}\hat{H}_j \qquad \text{for } i=1,2,\ldots,P, \\ R_2\left(\hat{\boldsymbol{\theta}}^*, \hat{\boldsymbol{H}}^*\right) \equiv\ & -H_1(\hat{H}_i)\sum_{j=0}^{P}\hat{\theta}_j\ell_j'(\xi_i) - \hat{\theta}_iH_1'(\hat{H}_i)\sum_{j=0}^{P}\hat{H}_j\ell_j'(\xi_i) + JC_E(\hat{H}_i) - \ldots \\ & \ldots - H_1(\hat{H}_i)\hat{\theta}_i\left.\frac{\mathrm{d}\ln u_e}{\mathrm{d}\xi}\right|_{\xi_i} - \frac{H_1(\hat{H}_i)+\hat{H}_i}{u_e|_{\xi_i}}\frac{\mu_\theta}{J}\sum_{j=0}^{P}\mathbf{F}_{ij}\hat{\theta}_j - \ldots \\ & \ldots - (H_1'(\hat{H}_i)+1)\frac{\hat{\theta}_i}{u_e|_{\xi_i}}\frac{\mu_H}{J}\sum_{j=0}^{P}\mathbf{F}_{ij}\hat{H}_j \qquad \text{for } i=1,2,\ldots,P, \end{aligned} \tag{4.4.20, 4.4.21}$$

which finally can be fed back as input for the point-implicit time-iterative scheme, Equation 4.3.13 and Equation 4.3.14.

In this section and section 4.3, it was explained in which numerical manner the equations would be solved, and how the numerical process is stabilized. In the next section, section 4.5, the appropriate polynomial order for which to run the simulations will be decided.

4.5 Polynomial order refinement

Whereas it is common to perform a mesh refinement to obtain an idea of the convergence of a scheme, in the present study only one element is used, and as such, the mesh is not refined. What will be refined, however, is the polynomial order; such a refinement is performed on the scheme by increasing the amount of nodes, such that the polynomial expansion of the solution has a higher order. In section 4.1, it was already shown that the polynomial order of the Lagrange basis function equals the amount of nodes minus 1. Hence, increasing the amount of nodes will increase the order of the approximating polynomials, allowing an order refinement to be performed. The objective of this order refinement is to establish an adequate number of nodes, i.e., degrees of freedom, for the simulations.

The system is tested on the symmetrical case of the NACA 0012 airfoil at 0° angle of attack. A simulation of polynomial order 60 is run, of which the solution is taken as the reference solution, to determine which order is reasonably accurate. All simulations are obtained with the same Dirichlet boundary condition coming from Thwaites' method, which runs until the streamwise coordinate $s = 0.1$, after which the switch to the higher-order Galerkin scheme is made. The results are shown in Figure 4.5.

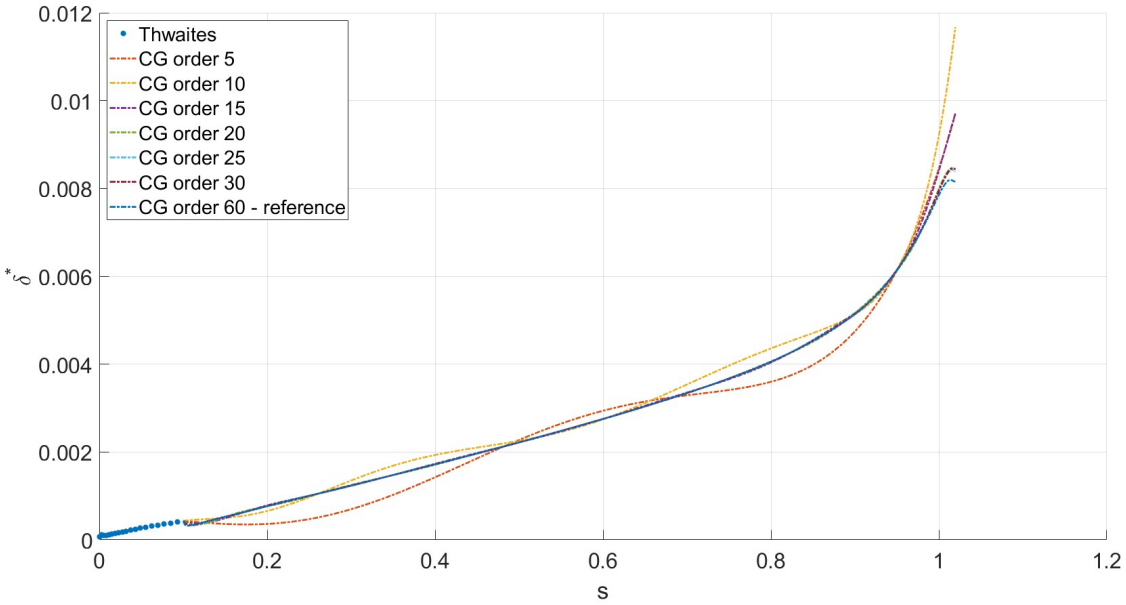

Figure 4.5: Order refinement of solution for NACA 0012 airfoil at 0°.

In Figure 4.5, the displacement thickness of the different solutions is shown, including that of the reference solution. Especially the solutions of the lower orders diverge significantly from the reference, whereas from order 20, the solutions seem to stabilize.

Reconstructing the solution at any given streamwise coordinate s within the domain of the turbulent boundary layer, is done through polynomial interpolation. As mentioned in section 4.1, a higher-order Galerkin solution consists of the summation of the products of the solution coefficients with their respective polynomial from the set of basis functions. Summing all these

intermediate solutions gives the total solution for the system, being

$$\theta(s) \approx \sum_{j=0}^{P} \hat{\theta}_j \ell_j, \qquad H(s) \approx \sum_{j=0}^{P} \hat{H}_j \ell_j. \tag{4.5.1}$$

In subsection 4.2.1, the derivation on how to obtain the solution for the displacement thickness δ^* was shown, which is approximated by a polynomial of order $2P$, namely Equation 4.2.14, repeated for completeness,

$$\hat{\delta}^*_k(s) \approx \sum_{k=0}^{2P} \hat{\delta}^*_k \ell_k. \tag{4.5.2}$$

Finding the solution for any arbitrary $s_?$ would simply entail finding the corresponding linearly mapped standard coordinate $\xi_?$, and the solution is found by using this coordinate as the input in the Lagrange polynomials,

$$\hat{\delta}^*_k(s_?) \approx \sum_{k=0}^{2P} \hat{\delta}^*_k \ell_k(\xi_?). \tag{4.5.3}$$

All that is left, is to decide on the appropriate number of nodes (or polynomial order) for which to run the simulations. A visualization of the residuals of the solutions in Figure 4.5 helps decide this amount of nodes. The residuals are calculated, once again using the L2 norm Equation 4.1.19, with u_{sol} being the reference solution of order 60, and u_{num} the numerical solutions of lower order. The calculated residuals for all orders can be seen in Figure 4.6.

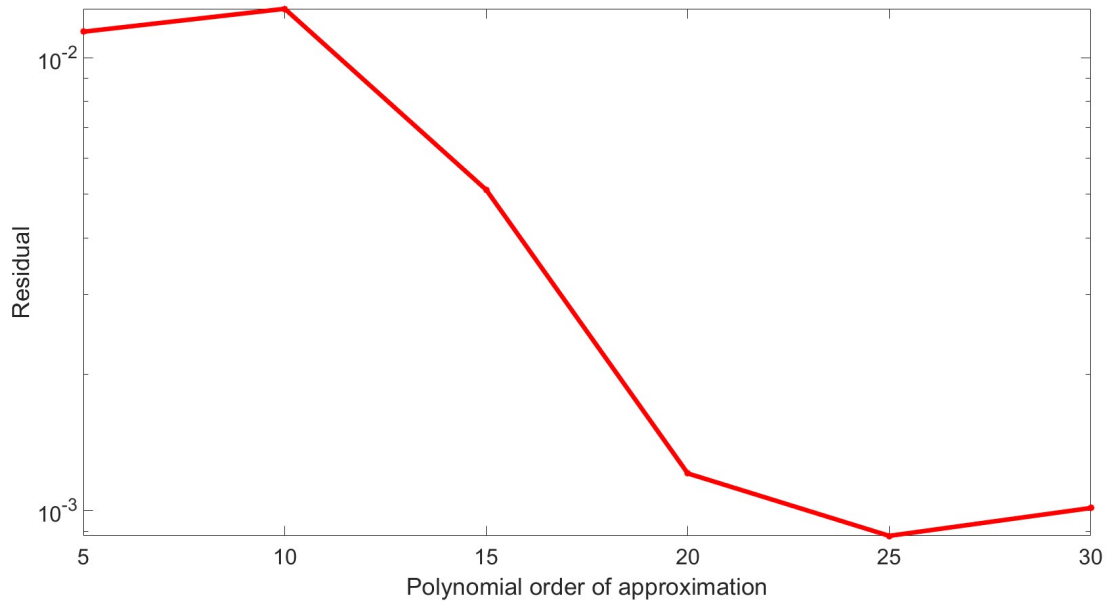

Figure 4.6: Residuals for different polynomial orders, solution of order 60 as reference.

Observing Figure 4.6, the residual drops an order of magnitude from order 10 to order 20. After order 20, the solution seems to stabilize, indicating that solutions from order 20 and higher are sufficiently close to the reference solution. As such, polynomial approximations of 20th order are chosen to represent the solutions from the boundary layer solver to obtain sufficiently accurate results, but not to unnecessarily increase the computational cost of the simulations.

It is interesting to note that Figure 4.6 shows a different trend than Figure 4.3. The error convergence of the two solutions do not agree with each other, as there is not a clear linear trend distinguishable in Figure 4.6, and as such, the solution does not necessarily converge exponentially. An explanation for this loss of exponential convergence could be the linear interpolation to the set of solution coefficients $\hat{u}_j$. An alternative, more likely explanation for this loss is the first-order nature of the closing relations mentioned in section 3.3. In addition, these relations, more specifically Equation 3.3.7 and Equation 3.3.8, contain a non-smooth derivative, which affects the convergence of higher-order Galerkin, losing its exponential character [25]. As such, the error convergence $\epsilon \propto h^{P+1}$ would not be valid anymore. To verify this suspicion, a simulation could be run with fully smooth closing relations. This assessment, however, lies outside the scope of the current research.

The polynomial order refinement concludes the chapter on higher-order Galerkin. The performance of the solver will be demonstrated and discussed in the next chapter, chapter 5.

Chapter 5

Results

This chapter will present the results obtained from the boundary layer solver. For various airfoils at various angles, the boundary layer parameters and resulting viscous pressure distribution will be shown and compared to *XFOIL* and experimental results. Moreover, the aerodynamic parameters lift and drag coefficient are discussed. The lift and drag coefficients are obtained using the pressure and skin friction distributions. The derivation on how they are obtained is given in Appendix A.

5.1 The symmetrical case

The most straightforward result obtained from the boundary layer solver is the momentum thickness θ and the shape factor H, as these are the variables for which the adopted set of integral boundary layer equations are set up and solved. As the simulations are run from the leading edge stagnation point until the trailing edge, two sets of results are obtained, namely one for the suction, or upper side, and one for the pressure, or lower side. The symmetrical case of the NACA 0012 airfoil at 0° angle of attack will be presented first, as a means to verify whether the outcome is symmetric, as it should be. Pressure coefficient values are available from experiments performed by Ladson [62]. The experiments were conducted at an angle of attack of 0.021°, for a chord-based Reynolds number of $9.0 * 10^6$, with tripping devices at 5% chord length to guarantee fixed transition to turbulent flow. These circumstances have been recreated in *XFOIL* and in the present boundary layer solver to resemble the circumstances in which the experiments were performed. For the sake of completeness, a simulation at exactly 0° angle of attack and chord-based Reynolds number of $1.0 * 10^6$ has also been performed to demonstrate ideal symmetry, and can be seen in Appendix B.

Observing the results from Figure 5.1, there is an overprediction of the momentum thickness when turbulence sets in. The laminar part that is calculated by Thwaites' method seems to align well, however the curve produced by the adopted integral boundary layer equations is significantly different in the turbulent region. The 'tripping' device at 5% chord length, or streamwise coordinate $s = 0.065$, can very clearly be seen in the shape factor graph, as the shape factor reaches the appropriate value of around 2.6 for laminar flow and, when turbulence sets in, drops to around 1.4, with some fluctuations for turbulent flow, likely due to underresolution. Referring back to Kwon et al. [47], the shape factor of a turbulent flow that begins to separate

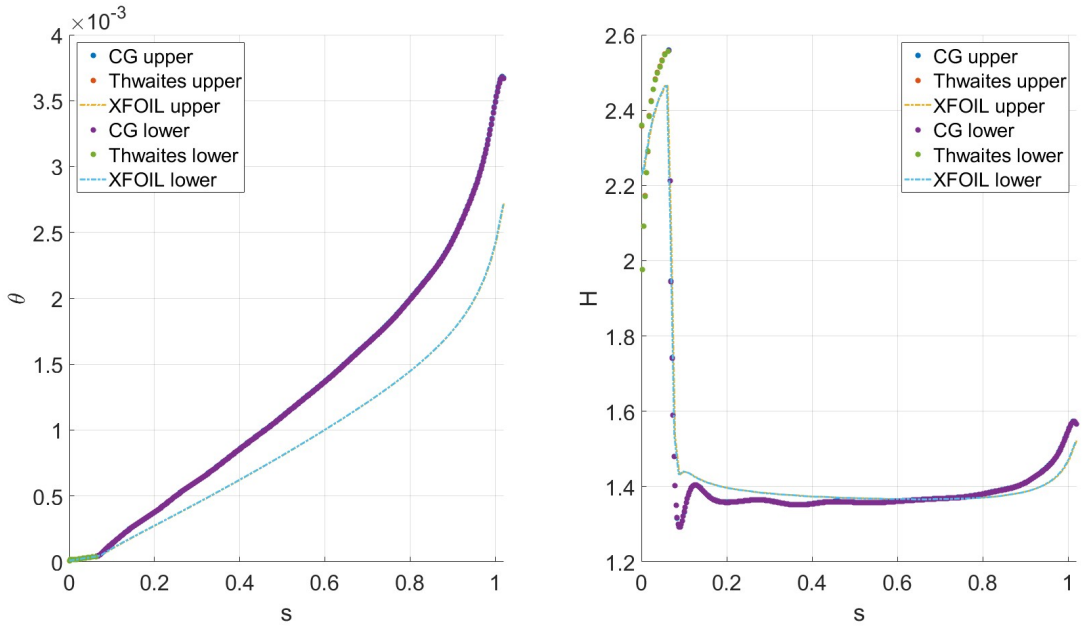

Figure 5.1: The momentum thickness and shape factor results for NACA 0012 at 0.021°, approximated with a Lagrange polynomial of order 20 at Reynolds of $9 * 10^6$*, compared with XFOIL.*

rises to between 2 and 2.5. These values are not reached yet in Figure 5.1, however nearing the trailing edge of the foil, the shape factor is already starting to gradually increase. From the shape factor plot, it can be seen that the increase in shape factor sets in slightly earlier in the boundary layer solver than in *XFOIL*, at around 80% chord length.

It should be noted that the results for the shape factor in the laminar boundary layer, produced by *XFOIL*, stem from a set of laminar boundary layer equations [11], and hence, it avoids any approximations and simplifications that are inherent to the assumptions made to apply Thwaites' method. As such, *XFOIL* is likely to produce more accurate results than the results for the shape factor in the laminar region from the present boundary layer solver. Moreover, *XFOIL* uses a different, less simplified model for the turbulent boundary layer as well, namely Green's shear stress lag-entrainment equation [63]. However, it is stressed that the objective of the present study is not to potentially produce boundary layer parameter results of improved accuracy, but rather to test the viability of the semi-strong interaction combined with a higher-order Galerkin discretization for the turbulent boundary layer.

A curiosity in section 5.1 is the fluctuations, just after transition to turbulent flow has set in. At first glance, an explanation might be Runge's phenomenon of oscillation at an element's edges. However, the nodes in the higher-order Galerkin were collocated following the Gauss-Lobatto nodes of quadrature, thereby lumping the mass matrix but also avoiding Runge's phenomenon as Gauss-Lobatto nodes are more densely distributed closer to the borders of the domain [64]. A probable explanation is an underresolution in the spatial scheme, indicating that a higher number of nodes would solve the problem. Increasing the amount of nodes increases the polynomial order, and more importantly, the computational cost. Another more preferable solution could be to increase the spectral viscosity to dampen out these fluctuations.

Multiplying the momentum thickness and the shape factor, one obtains the displacement thickness. This result, including comparison, can be seen in Figure 5.2.

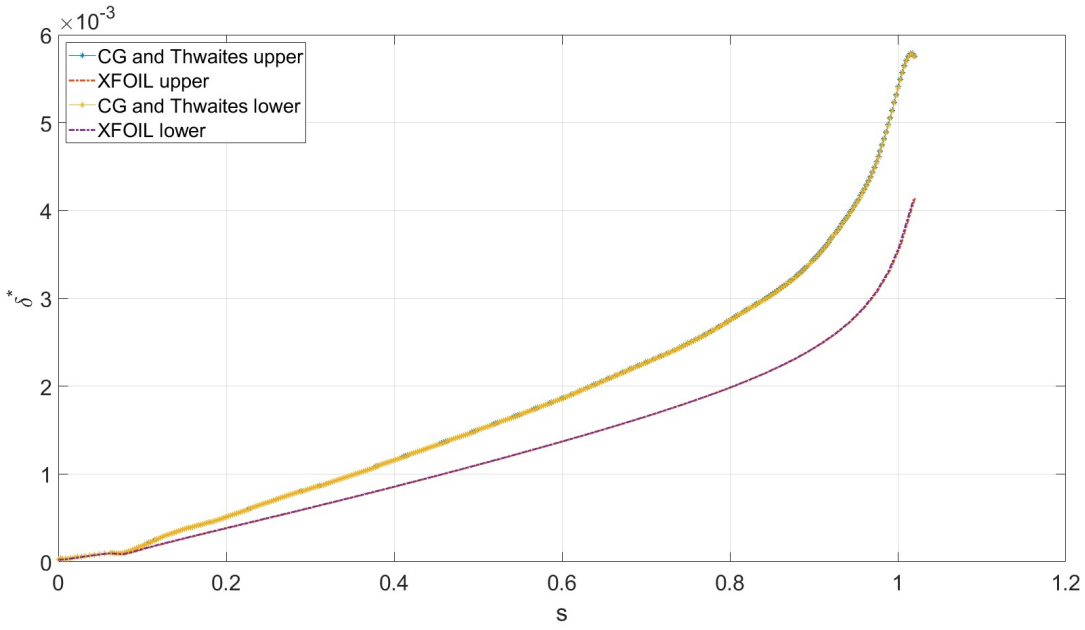

Figure 5.2: The displacement thickness for NACA 0012 at 0.021°, approximated with a Lagrange polynomial of order 20 at Reynolds of $9 * 10^6$*, compared with XFOIL.*

Here as well, the curves are aligned on top of each other, emphasizing the symmetrical nature of the problem. The overestimation is visible in the turbulent part, whereas the laminar part before tripping follows *XFOIL*'s curve closely. Once more, the trip at 5% chord length, $s = 0.065$, can be seen clearly, indicating transition to turbulence has set in. Moreover, the effect of the fluctuations in the shape factor just after transition is seen in this dip downward.

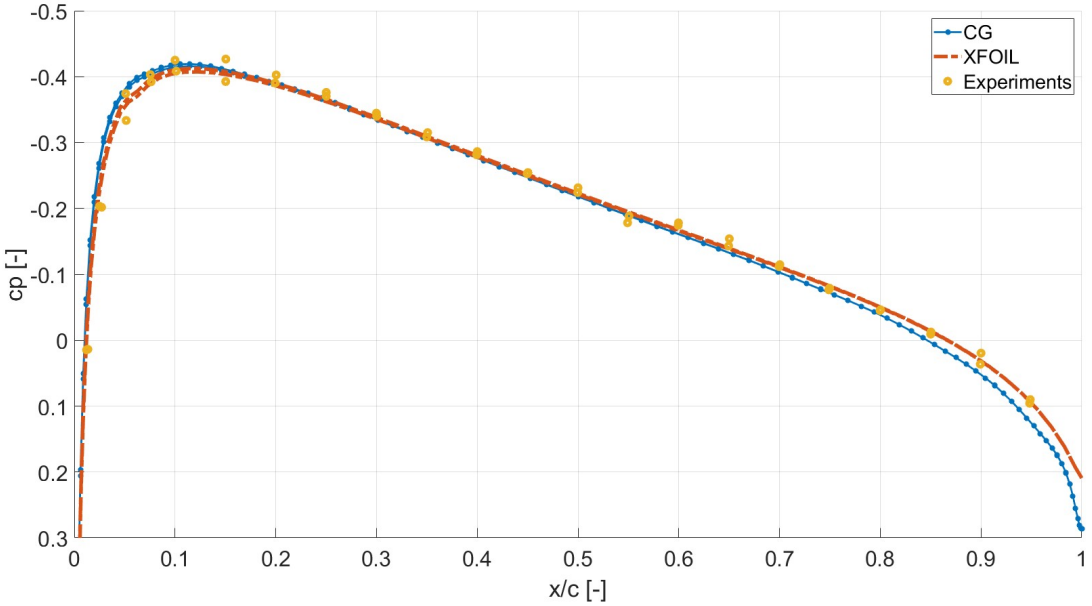

Figure 5.3: The viscous pressure distribution for NACA 0012 at 0.021° at Reynolds of $9 * 10^6$*, compared with XFOIL and experiments [62]; cut off at 0.3 for legibility.*

Now turning to the pressure distribution in Figure 5.3, obtained using the displacement thickness from Figure 5.2 as input, decent correspondence between the two graphs can be seen

except when nearing the trailing edge of the airfoil, where a small underestimation can be seen. Moreover, there is a small overprediction near the suction peak. Experimental results, produced by Ladson [62], have been added for additional reference. It is repeated that the simulation is not run exactly symmetrically, but rather at 0.021° angle of attack, to stay more loyal to the ambiance in which the experiments were performed. Furthermore, it should be noted that the graph is cut off to show the differences between the results better. Both numerical methods report a leading edge stagnation point value of 1, whereas the experimental results show a stagnation point value of 1.0416, reminding that experiments are not without error; the experimental results show a larger discrepancy between pressure and suction side values than the numerically produced results.

Another method of comparing results, is by looking at the aerodynamic coefficients generated by the solvers. These coefficients are presented in Table 5.1. As the foil is a symmetric profile and the flow around the body is (practically) symmetric, no significant lift is produced by the airfoil. Therefore, a qualitative comparison of lift coefficients is not very sensible. There is decent correspondence between drag coefficients, however, with a slight underprediction of drag from the present boundary layer solver.

	BL solver	*XFOIL*	Experiment
C_L	0.0013	0.0024	-0.0052
C_D	0.0069	0.00742	0.00765

Table 5.1: Aerodynamic coefficients for NACA 0012 at 0.021°.

To verify the numerics, a truly symmetric case was produced as well, where the boundary layer parameters from upper and lower side, and the pressure from both sides lie perfectly on top of each other. This result can be seen in Figure B.3, in Appendix B.

5.2 Symmetrical airfoil at an angle

A different situation is considered; the symmetrical NACA 0012 airfoil is set at an angle of attack of 6°. Firstly, the momentum thickness and shape factor are obtained from the simulation.

In the momentum thickness graph of Figure 5.4, four curves can be distinguished, namely from suction and pressure side, and produced by higher-order Galerkin and *XFOIL*. Here as well, one can see that the boundary layer model adopted in the present study overestimates the growth of the boundary layer thickness on both suction and pressure side. Turning to the shape factor graph, the solver produces results that align closely with the *XFOIL* curves in the turbulent region, especially for the lower part. Moreover, some differences are observed on both sides along the laminar portion, as also noted in the previous section. The fluctuations after transition to turbulent flow can still be seen, but they are less intense when compared to the fluctuations in Figure 5.1. Overall, if the angle of attack is not too large, there is relatively good correspondence for the shape factor. The result of the overprediction of the momentum thickness, however, is visible in the displacement thickness in Figure 5.5, where the curves for both sides differ from *XFOIL* in the turbulent region as well.

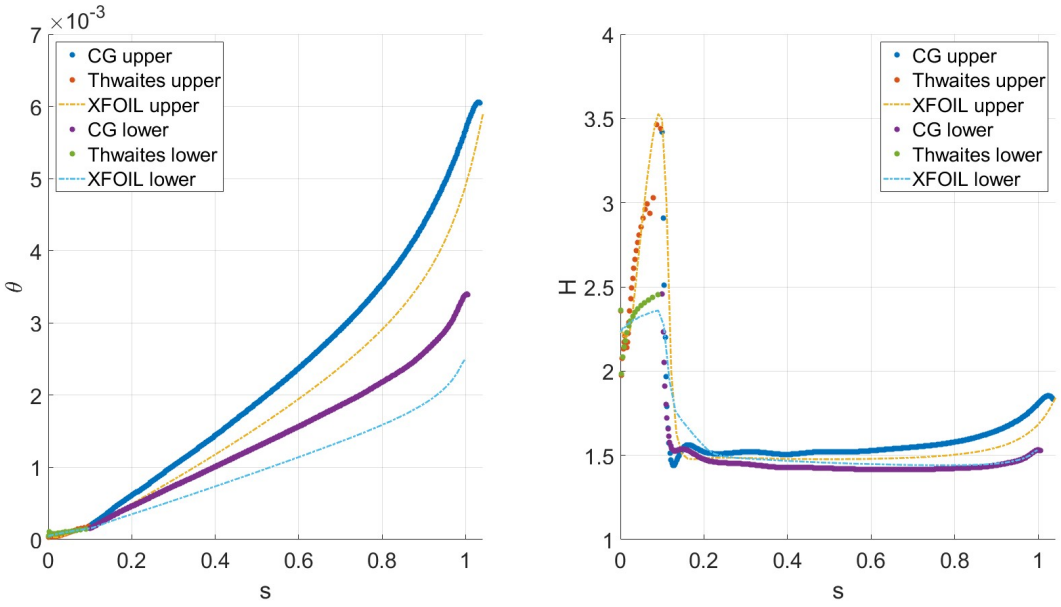

Figure 5.4: The momentum thickness and shape factor results for NACA 0012 at 6°, approximated with a Lagrange polynomial of order 20 at Reynolds of $1 * 10^6$*, compared with XFOIL.*

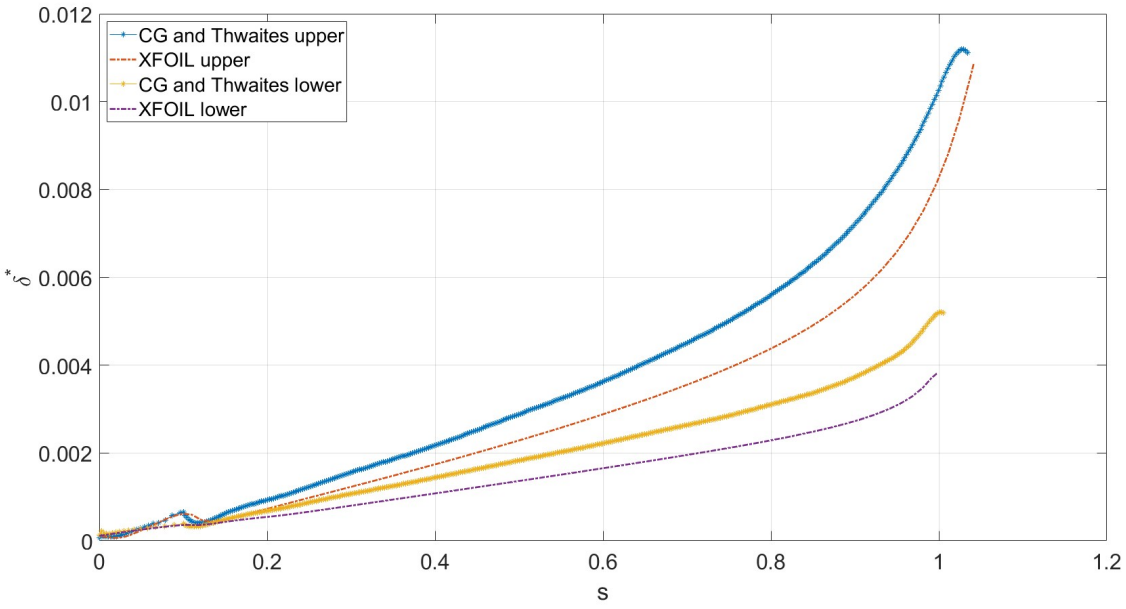

Figure 5.5: The displacement thickness for NACA 0012 at 6°, approximated with a Lagrange polynomial of order 20 at Reynolds of $1 * 10^6$*, compared with XFOIL.*

In Figure 5.6, the pressure distribution for the viscous case can be seen. The pressure difference is underpredicted on suction and pressure side, but arrives, flowing off of the trailing edge smoothly, and corresponding well with *XFOIL*. This underprediction will be discussed in more detail in section 5.4. Moreover, in the *XFOIL* curve, the forced transition is more explicitly visible as a bump, however, being milder in the present boundary layer model.

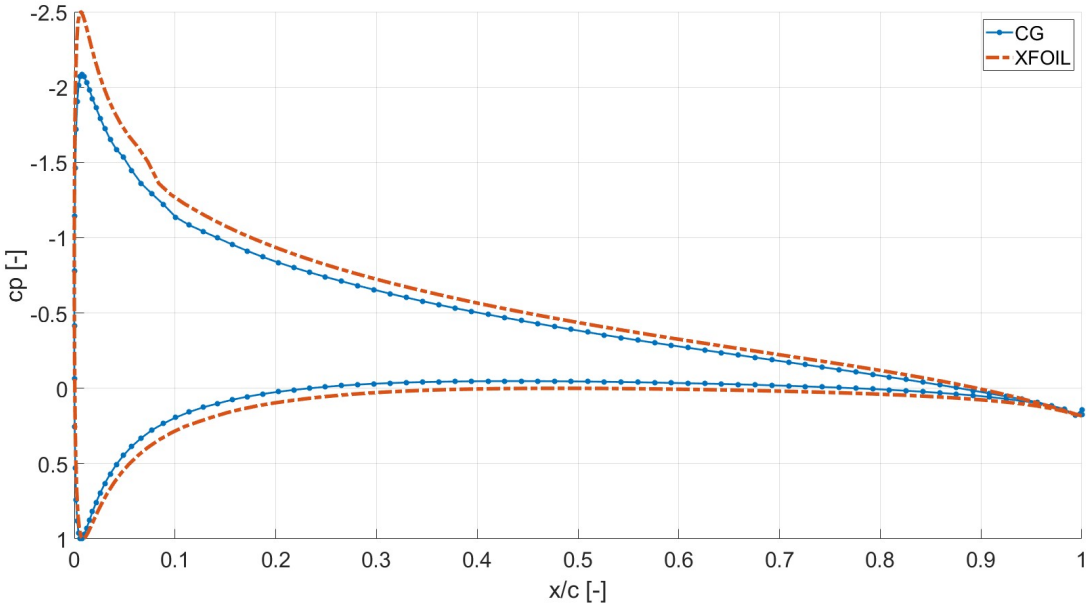

Figure 5.6: The viscous pressure distribution for NACA 0012 at 6° at Reynolds of $1 * 10^6$*, compared with XFOIL.*

Lastly, the aerodynamic coefficients are presented in Table 5.2. The underprediction of the pressure coefficient, especially near the suction peak, has a direct impact on the lift coefficient, which is underpredicted as well. The higher drag coefficient might be attributed to a higher pressure drag, whose contribution to the total drag is increasing, likely because the flow is tending to form a separation bubble on the suction side trailing edge.

	BL solver	*XFOIL*
C_L	0.5476	0.6693
C_D	0.0144	0.01193

Table 5.2: Aerodynamic coefficients for NACA 0012 at 6°.

5.3 Cambered airfoil at an angle

As a last result, a cambered NACA 2412 airfoil at an angle of attack of 6° is presented.

In Figure 5.7, on the trailing edge, the shape factor can be seen to reach the value of 2, indicating that a separation bubble is beginning to form. In a way, *XFOIL* seems to 'catch up' with the overpredicted boundary layer parameters and can be seen to surpass them in Figure 5.7 and Figure 5.8, on the suction side.

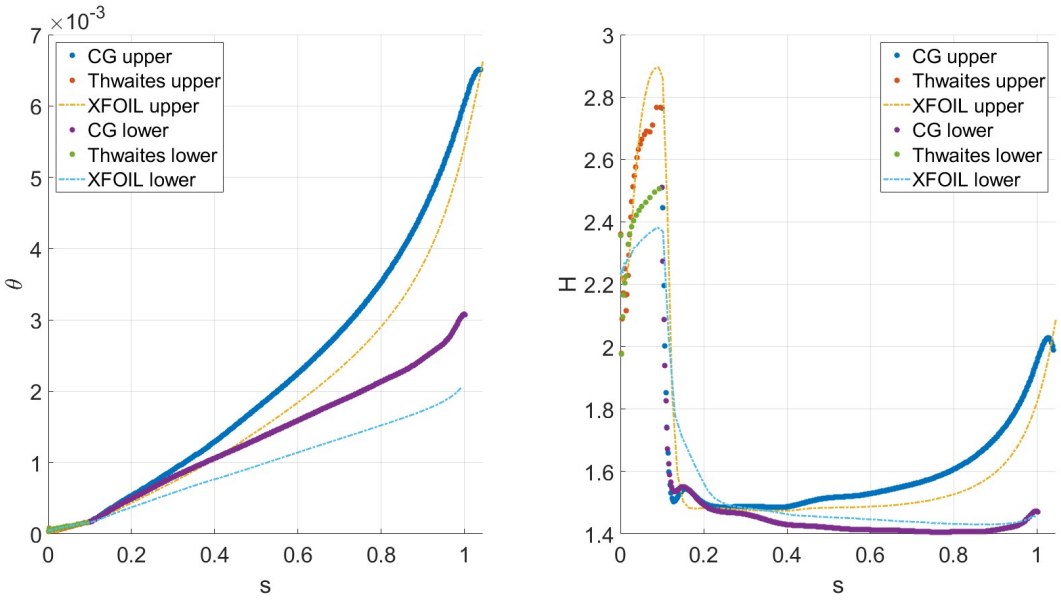

Figure 5.7: The momentum thickness and shape factor results for NACA 2412 at 6°, approximated with a Lagrange polynomial of order 20 at Reynolds of $1 * 10^6$*, compared with XFOIL.*

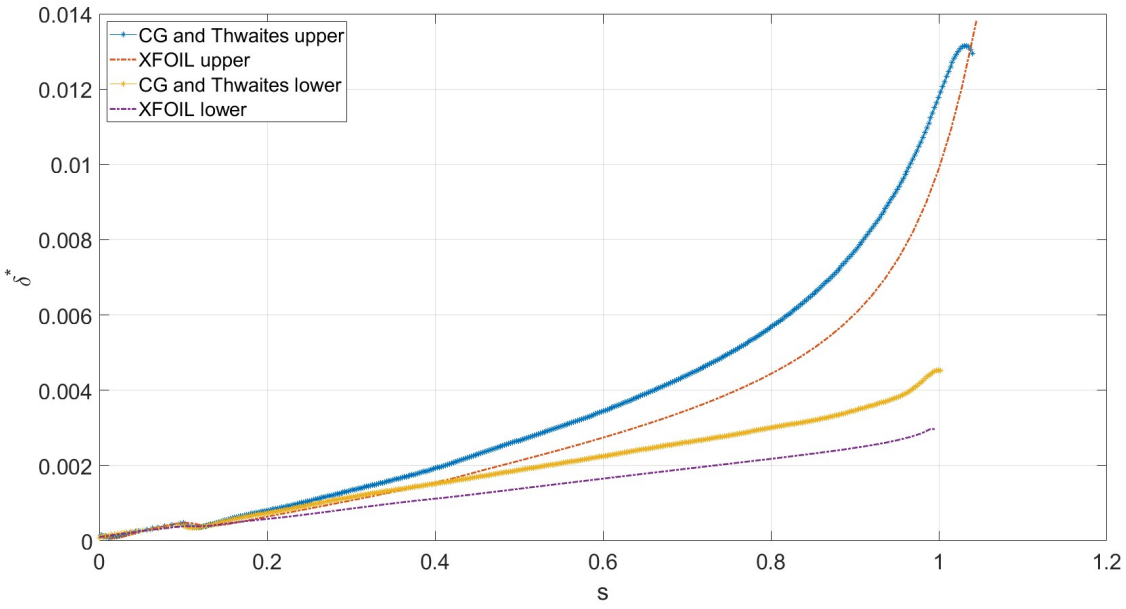

Figure 5.8: The displacement thickness for NACA 2412 at 6°, approximated with a Lagrange polynomial of order 20 at Reynolds of $1 * 10^6$*, compared with XFOIL.*

Once more, the pressure distribution along both sides of the airfoil is quite underestimated, as can be seen in Figure 5.9.

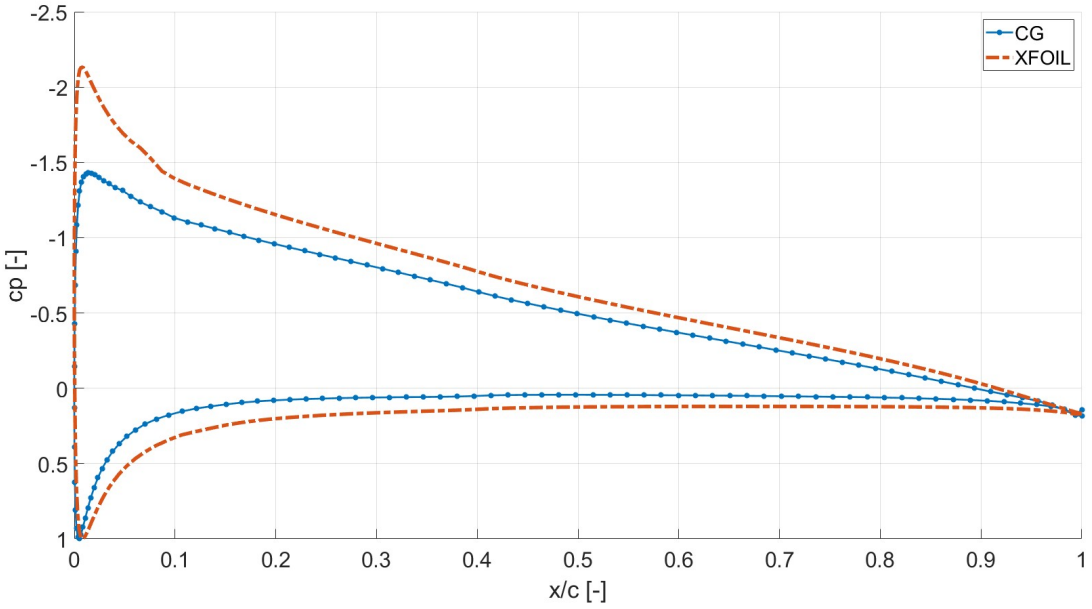

Figure 5.9: The viscous pressure distribution for NACA 2412 at 6° at Reynolds of $1 * 10^6$*, compared with XFOIL.*

Lastly, the aerodynamic coefficients can be seen in Table 5.3. The underprediction in lift is larger than in the case of the symmetrical airfoil at an angle in Table 5.2, which can be explained by the fact that the pressure distributions in Figure 5.9 differ more than in Figure 5.6. Moreover, the drag is overpredicted in comparison to *XFOIL*'s value. This could be attributed to a separation bubble having formed on the suction side trailing edge, as indicated by the shape factor in Figure 5.7. Part of the flow leaving the trailing edge being separated drastically increases the pressure drag, which could be the cause of a significantly higher drag coefficient in the boundary layer model.

	BL solver	*XFOIL*
C_L	0.6460	0.8813
C_D	0.0211	0.01265

Table 5.3: Aerodynamic coefficients for NACA 2412 at 6°.

5.4 Some general remarks on the results

The capability of the present boundary layer model was limited to an angle of attack of 10°. Above 10°, the simulation had a high tendency towards imaginary values, which would cause the system of equations to diverge. The reason for this divergence most likely is an overprediction of the shape factor H in the laminar region of the boundary layer, calculated by Thwaites' explicit integration. The value for the shape factor at transition was taken as Dirichlet boundary condition for the higher-order Galerkin scheme, and would become higher for laminar boundary layers at higher angles of attack, leading to a steeper drop in shape factor at transition. If the drop was too high, the simulation would overshoot, surpassing the shape factor value of 1.4 for turbulent boundary layers, and eventually diverge.

An alternative explanation for the divergence could be a too steep decrease in the pressure distribution, or inversely a too high acceleration of the flow on the suction side, causing the momentum thickness to grow too rapidly and the simulation breaking down. One more possibility, is the instability of the scheme itself, namely the combination of higher-order Galerkin spatial discretization with the point-implicit scheme. However, the spectral vanishing viscosity was added to stabilize the discretization. Moreover, to try to resolve the divergence, the time step in the point-implicit scheme was reduced, but it was found that in the majority of simulations, this only delayed the divergence. Using a previous result from a lower angle as input for the simulation would accelerate the convergence of the scheme, especially for lower angles of attack, but above 10° angle of attack, the simulations would break down nonetheless.

There is a general underprediction in the pressure distributions and a general overprediction in the boundary layer itself. Firstly, the underprediction in the pressure can be explained by the fact that a simplified viscous wake was used, as the boundary layer was assumed to remain constant, and only the effect of the velocity change was used as input for the blowing of the panels. Essentially, the boundary layer would stay 'open' after the trailing edge, even though there is not a friction generating surface anymore. A fully viscous wake would account for the effect of a diminishing boundary layer as well, which could have an impact on the overall pressure distribution. Due to time limitations, a fully viscous wake could not be implemented.

The wake did however already fulfill its purpose, as a wrinkle accumulation was visible before the simplified wake was implemented, that became more extreme per iteration between panel method and viscous model. Namely, without the wake, a small wrinkle in velocity at the trailing edge would translate in a similar wrinkle in the momentum thickness in the viscous region. This would cause a wrinkle in the displacement thickness, and as such, the wrinkles would become larger per simulation. Without a wake, this accumulation of wrinkles had no way to 'escape', and ultimately the simulation would diverge. With the wake, any wrinkle would only appear after the first iteration, as the first cycle uses the fully inviscid pressure distribution as input. In the subsequent viscous pressure distributions, in which the velocities are slightly lower, the wrinkles would disappear.

The overprediction in the present boundary layer model could be attributed to the use of a different set of equations than *XFOIL*. Where in the present study the Von Kármán momentum integral equation and Head's entrainment equation are used to close the system, *XFOIL* uses the compressible integral momentum and kinetic energy equations, and Green's shear stress lag equation [63] to close the system in turbulent conditions. It would not be extraordinary that a different set of equations does not produce identical different results.

A thought that comes to mind is the following: as the displacement thickness of the boundary layer is overpredicted by the present solver, one can assume that the entire viscous region is in fact estimated larger than it is in reality. Comparing *XFOIL*'s inviscid pressure distributions with its viscous solutions, the inviscid peaks are always reported higher than in viscous situations. This gives reason to believe that a larger effective aerodynamic body would result in a lower suction peaks in its pressure distribution. It should be noted that the suction peak occurs in the laminar region, and as such, the calculation of the laminar boundary layer parameters using Thwaites' method of explicit integration could also be a source of error.

Another source of error affecting the overall accuracy must be mentioned. In the panel

method, circulations are integrated over all the panels of the airfoil, which is performed numerically by MATLAB. As the control points are the midpoints of each panel, possible singularities along the panel are avoided by reaching sufficiently close and then bypassing them. These singularities do not occur in the viscous case, as the control points are then set one displacement thickness δ^* away from the airfoil surface. One should remember that a numerical error is introduced in the process of numerical integration. Analytical expressions are, however, readily available from Kuethe and Chow [38], and could potentially provide better results. It should be noted that these analytical expressions are derived for a fully inviscid case, where no boundary layer exists. As such, it shows accurate inviscid pressure distributions, but its implementation (and performance) for viscous cases could prove more cumbersome.

Lastly, it was found that, for higher angles of attack, Thwaites' method of explicit integration of the laminar boundary layer was not fully reliable anymore. For an angle of attack of 8° and higher, free transition would already occur before the streamwise coordinate of $s = 0.1$ for forced transition, with the exact position of free transition depending on the Reynolds number. Hence, the laminar region would become smaller, resulting in the need for a higher resolution to obtain accurate momentum thickness and shape factor from Thwaites' integration. Moreover, the pressure distributions would show high suction peaks for these (and higher) angles, resulting in a large acceleration of the flow on the suction side. As seen in Equation 3.2.2, high acceleration directly results in a very high λ, giving off a high shape factor that could cause divergence, as explained in the beginning of this section. As such, Thwaites' method could have a tendency to produce inaccurate results for the laminar boundary layer, and perhaps more importantly, to provide an unreliable Dirichlet boundary condition for the turbulent domain.

Chapter 6

Conclusions and recommendations

6.1 Conclusions

An interactive, quasi-simultaneous flow solver, based on a simple semi-strong coupling law between boundary layer and inviscid model, has been developed that is able to predict the momentum thickness and shape factor of a turbulent boundary layer. The present study proved successful in its primary objective: to demonstrate the viability of a semi-strongly coupled scheme, based on a higher-order Galerkin formulation, to solve the integral boundary layer equations over airfoils. It was found that the solver could produce results up to an angle of attack of 10° (this angle would probably be lower for strongly cambered airfoils), after which the simulations would diverge due to a sharp drop in the shape factor when transition sets in. Three specific cases have been analyzed, namely a fully symmetrical case, a symmetrical foil at 6° angle, and a cambered airfoil at 6° angle. The obtained results showed consistency with the results produced by *XFOIL*, however, in some aspects, a certain lack of accuracy was observed.

In all cases, an overestimation was seen in the prediction of the momentum thickness in comparison to *XFOIL*. Due to the quickly growing momentum thickness, a small separation on the trailing edge on the suction side would occur earlier than *XFOIL* results showed. The shape factor resembled the curves from *XFOIL* closely, in the laminar boundary layer region as well as in the turbulent region after transition had taken place. However, due to the overprediction of the momentum thickness, the displacement thickness was also overestimated, and as such, the borders of the viscous region were predicted larger as well.

Subsequently, a large underprediction of the pressure distribution was seen in the results, which grew more present in the test case of the cambered airfoil at an angle, leading to believe that an increasing angle would augment the underprediction. The underprediction of the pressure distribution can likely be attributed to the simplified viscous wake, in which the displacement thickness leaving the trailing edge was kept constant. Smooth outflow at the trailing edge, which had been a problem anteriorly due to the accumulation of velocity wrinkles at the trailing edge, was guaranteed by this wake in all cases. Furthermore, any other wrinkles or oscillations that could cause divergence in the system at lower angles of attack were dampened successfully by the implementation of spectral viscosity in the higher-order Galerkin scheme, additionally stabilizing the numerics.

6.2 Recommendations

In future work, a fully viscous wake should be implemented. Currently, two wake lines are extrapolated from the inviscid flow distribution, and only the change in velocity is accounted for when calculating the blowing velocity that the wake panels should have, according to the surface transpiration model. It should be noted that this is an accelerating velocity, as the flow at the trailing edge is generally slower than the uniform flow, and as such, it induces a positive outward velocity on the wake panels, or a blowing velocity. In any case, the boundary layer thickness is held constant in the wake, but from a physical point of view, it should slowly start to diminish until it disappears sufficiently far away from the trailing edge. This diminishing effect has a negative contribution to the transpiration, and hence, an inward, sucking velocity is induced on the panels. This sucking can be interpreted as the 'closing' of the boundary layer, and the flow proceeding uniformly again. Leaving the boundary layer 'open', as it is currently done, likely exerts a large influence in the underprediction of the produced pressure distributions from the simulations. This probably has an impact on the Kutta condition, which in turn impacts the pressure distribution. Another factor impacting the pressure near the leading edge and nearby regions (often including suction and pressure peaks) is the solution of the laminar boundary layer based on the Thwaites' approximations. A less simplified version of the integral boundary layer equations for the laminar part could be adopted instead of Thwaites' method. Using a more accurate method for the laminar region would probably result in a more accurate prediction of suction and pressure peaks, which strongly impact overall pressure distribution.

An alternative approach to the solver process would be to divide the turbulent boundary layer in two segments, the first being the 'normal' turbulent boundary layer and the second segment being the region where the flow might be prone to separation. In the first region, the integral boundary layer equations would be solved using an explicit ODE solver (e.g. MATLAB's ODE45), such that the reported sharp decrease in the shape factor at the point of transition would not cause any divergence in the model. The segment where separation might occur would be solved by the higher-order Galerkin scheme. This would have the advantage of additional robustness, as the shape factor decrease would not cause the simulations to diverge. Moreover, the simulations would be accelerated, as the explicit ODE solver can produce results more quickly than the point-implicit scheme; in addition, the computational cost would be less, as the domain for higher-order Galerkin would be smaller, thus requiring less nodes, or degrees of freedom, to represent this 'reduced' part of the turbulent boundary layer accurately. Potentially, this structure would allow for calculations at higher angles of attack.

Lastly, the second equation from the governing system of equations in the present study, being Head's entrainment equation, might not form an adequate model to produce very accurate results. Although Veldman reports a good correspondence to the lift coefficients generated by his model [18], it does not guarantee excellent agreement in boundary layer profiles and parameters generated by both models. Head's closure is known to be a somewhat obsolete model. In fact, alternative, more sophisticated models are widely employed in boundary layers solvers nowadays [11][65]. An example of an alternative formulation of the integral boundary layer equations, employed by *XFOIL*, would be to set up the system of equations with the more sophisticated Green's lag-entrainment equation [63].

Bibliography

[1] L. Prandtl. "Über Flüssigkeitsbewegung bei sehr kleiner Reibung". In: *Verhandlungen des dritten internationalen Mathematiker-Kongresses* (1905).

[2] Th. Von Kármán. "Über laminare und turbulente Reibung". In: *Zeitschrift für angewandte Mathematik und Mechanik* (1921).

[3] H. Schlichting and K. Gersten. *Boundary Layer Theory*. Springer, 2017.

[4] T. Cebeci and J. Cousteix. *Modeling and Computation of Boundary-Layer Flows*. Springer, 2010.

[5] M. Drela. *Flight Vehicle Aerodynamics*. The MIT Press, 2014.

[6] H. Özdemir. "Interacting Boundary Layer Methods and Applications". In: *Handbook of Wind Energy Aerodynamics* (2020).

[7] S. Goldstein. "On laminar boundary-layer flow near a point of separation". In: *The Quarterly Journal of Mechanics and Applied Mathematics* (1948).

[8] K. Stewartson. "On the flow near the trailing edge of a flat plate II". In: *Mathematika* (1969).

[9] A.F. Messiter. "Boundery-Layer Flow Near the Trailing Edge of a Flat Plate". In: *SIAM Journal on Applied Mathematics* (1970).

[10] V.V. Sychev. "Laminar separation". In: *Fluid Dynamics* (1972).

[11] M. Drela. *XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils*. Tech. rep. MIT Department of Aeronautics and Astronautics, 1989.

[12] S. Zhang. "Three-dimensional Integral Boundary Layer Method for Viscous Aerodynamic Analysis". PhD thesis. Massachusetts Institute of Technology, 2022.

[13] M. Drela. "Three-Dimensional Integral Boundary Layer Formulation for General Configurations". In: *21st AIAA Computational Fluid Dynamics Conference*. 2013.

[14] A.E.P. Veldman. *A Calculation Method for Incompressible Boundary Layers with Strong Viscous-Inviscid Interaction*. Tech. rep. National Aerospace Laboratory NLR, 1979.

[15] A.E.P. Veldman. "New, quasi-simultaneous method to calculate interacting boundary layers". In: *American Institute of Aeronautics and Astronautics* (1981).

[16] A.E.P. Veldman, J.P.F. Lindhout, E. De Boer and M.A.M. Somers. "Vistrafs: a simulation method for strongly interacting viscous transonic flow". In: *Numerical and physical aspects of aerodynamic flows IV*. Springer. 1990, pp. 37–51.

[17] A.E.P. Veldman. “Quasi-simultaneous viscous-inviscid interaction for transonic airfoil flow”. In: *4th AIAA Theoretical Fluid Mechanics Meeting*. 2005, p. 4801.

[18] A.E.P. Veldman. “A simple interaction law for viscous-inviscid interaction”. In: *Journal of Engineering Mathematics* (2009).

[19] M.R. Head. *Entrainment in the Turbulent Boundary Layer*. Tech. rep. Aeronautical Research Council, Reports and Memoranda, 1960.

[20] L. Euler. *Institutiones calculi differentialis*. Petropolis, 1755.

[21] A.T. Patera. “A Spectral Element Method for Fluid Dynamics: Laminar Flow in a Channel Expansion”. In: *Journal of Computational Physics* (1984).

[22] Y. Maday and A.T. Patera. “Spectral element methods for the incompressible Navier-Stokes equations”. In: *American Society of Mechanical Engineers* (1989).

[23] E. Faccioli, F. Maggio, R. Paolucci and A. Quarteroni. “2D and 3D elastic wave propagation by a pseudo-spectral domain decomposition method”. In: *Journal of Seismology* (1997).

[24] D. Komatitsch and J.P. Vilotte. “The spectral element method: An efficient tool to simulate the seismic response of 2D and 3D geological structures”. In: *Bulletin of the Seismological Society of America* (1998).

[25] G.E. Karniadakis and S.J. Sherwin. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, 1999.

[26] R.C. Moura. “Lecture notes from Spectral Element Methods”. Instituto Tecnológico de Aeronáutica. Jan. 2020.

[27] P.E. Vincent and A. Jameson. “Facilitating the adoption of unstructured high-order methods amongst a wider community of fluid dynamicists”. In: *Mathematical Modelling of Natural Phenomena* (2011).

[28] B. Van Es. “Comparison and Application of Unsteady Integral Boundary Layer Methods using various numerical schemes”. MA thesis. Delft University of Technology, 2009.

[29] E.F. Van den Boogaard. “High-order discontinuous Galerkin method for unsteady integral boundary layer equations”. MA thesis. Delft University of Technology, 2010.

[30] F.M. Haciahmetoğlu. “Investigation of unsteady viscous-inviscid interaction schemes”. MA thesis. Delft University of Technology, 2013.

[31] J.H. Seubers. “Path-consistent schemes for interacting boundary layers”. MA thesis. Delft University of Technology, 2014.

[32] F. Passalacqua. “Implementation of Unsteady Two-Dimensional Interacting Boundary Layer Method”. MA thesis. School of Industrial and Information Engineering Politecnico di Milano, 2015.

[33] H. Özdemir, A. Van Garrel, A. Koodly Ravishankara, F. Passalacqua and H. Seubers. “Unsteady interacting boundary layer method”. In: *35th Wind Energy Symposium*. 2017.

[34] B. Ye. “The Modeling of Laminar-to-turbulent Transition for Unsteady Integral Boundary Layer Equations with High-order Discontinuous Galerkin Method”. MA thesis. Delft University of Technology, 2015.

[35] S. Zhang, M. Drela, S.R. Allmaras, M.C. Galbraith and D. Darmofal. "A strongly-coupled non-parametric integral boundary layer method for aerodynamic analysis with free transition". In: *AIAA Scitech 2019 Forum*. 2019.

[36] R.C. Moura, J. Peiró and S.J. Sherwin. "Under-resolved DNS of non-trivial turbulent boundary layers via spectral/hp CG schemes". In: *Direct and Large Eddy Simulation XII*. Springer. 2020.

[37] H. Schlichting and E. Trukenbrodt. *Aerodynamics of the Airplane*. McGraw-Hill, 1979.

[38] A.M. Kuethe and C-Y. Chow. *Foundations of Aerodynamics: Bases of Aerodynamic Design*. John Wiley & Sons, 1998.

[39] J.D. Anderson. *Fundamentals of Aerodynamics*. McGraw Hill, 2016.

[40] P.A. Henne. "An inverse transonic wing design method". In: *AIAA 18th Aerospace Sciences Meeting*. 1980.

[41] B. Thwaites. *Incompressible Aerodynamics*. Oxford University Press, 1960.

[42] J.S. Turner. *Buoyancy effects in fluids*. Cambridge university press, 1973.

[43] J.C. LeBalleur. "Couplage visqueux-non visqueux: analyse du problème incluant décollements et ondes de choc". In: *La Recherche Aérospatiale* (1977).

[44] R.C. Lock. "Prediction of the drag of wings at subsonic speeds by viscous/inviscid interaction techniques". In: *AGARD-R-723* (1985).

[45] R. Houwink and A. Veldman. "Steady and unsteady separated flow computations for transonic airfoils". In: *17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference*. 1984.

[46] A.E.P. Veldman. "Matched asymptotic expansions and the numerical treatment of viscous-inviscid interaction". In: *Journal of Engineering Mathematics* (2001).

[47] K. Kwon, B. Chang, J. Lee and S.O. Park. "Boundary Layer Transition Measurement over an Airfoil by Using PIV with High Magnification". In: *13th Int Symp on Applications of Laser Techniques to Fluid Mechanics*. Citeseer. 2006.

[48] J. Carter and S. Wornom. "Solutions for incompressible separated boundary layers including viscous-inviscid interaction". In: *Aerodynamic Analysis Requiring Advanced Computers* (1975).

[49] T. Cebeci, H.H. Chen and J.A. Majeski. "Indented plate problem revisited". In: *International Journal for Numerical Methods in Fluids* (1993).

[50] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*. Springer, 2007.

[51] E. Waring. "Vii. problems concerning interpolations". In: *Philosophical transactions of the royal society of London* (1779).

[52] E.T. Whittaker and G.N. Watson. *A course of modern analysis: an introduction to the general theory of infinite processes and of analytic functions; with an account of the principal transcendental functions*. University press, 1920.

[53] C.G.J. Jacobi. "Über Gauß' neue Methode, die Werthe der Integrale näherungsweise zu finden". In: *Journal für die Reine und Angewandte Mathematik* (1826).

[54] I.A. Abramowitz M.; Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover Publications, 1983.

[55] N. Kovvali. *Theory and Applications of Gaussian Quadrature Methods*. Springer, 2011.

[56] C.F. Gauss. *Methodus nova integralium valores per approximationem inveniendi*. 1814.

[57] R.C. Moura, A.F.C. Silva, E.D.V. Bigarella, A.L. Fazenda and M.A. Ortega. "Lyapunov exponents and adaptive mesh refinement for high-speed flows using a discontinuous Galerkin scheme". In: *Journal of Computational Physics* (2016).

[58] R.R. Thareja, J.R. Stewart, O. Hassan, K. Morgan and J. Peraire. "A point implicit unstructured grid solver for the Euler and Navier–Stokes equations". In: *International journal for numerical methods in fluids* (1989).

[59] Y. Maday, S.M.O. Kaber and E. Tadmor. "Legendre Pseudospectral Viscosity Method for Nonlinear Conservation Laws". In: *Siam Journal on Numerical Analysis* (1993).

[60] R.C. Moura, S.J. Sherwin and J. Peiró. "Eigensolution analysis of spectral/hp continuous Galerkin approximations to advection–diffusion problems: Insights into spectral vanishing viscosity". In: *Journal of Computational Physics* (2016).

[61] R.C. Moura, M. Aman, J. Peiró and S.J. Sherwin. "Spatial eigenanalysis of spectral/hp continuous Galerkin schemes and their stabilisation via DG-mimicking spectral vanishing viscosity for high Reynolds number flows". In: *Journal of Computational Physics* (2020).

[62] C.A. Ladson. *Effects of independent variation of Mach and Reynolds numbers on the low-speed aerodynamic characteristics of the NACA 0012 airfoil section*. Tech. rep. NASA Langley Research Center Hampton, 1988.

[63] J.E. Green, D.J. Weeks and J.W.F. Brooman. *Prediction of turbulent boundary layers and wakes in compressible flow by a lag-entrainment method*. Tech. rep. Aeronautical Research Council, 1977.

[64] L.N. Trefethen. *Approximation theory and approximation practice, extended edition*. SIAM, 2019.

[65] M. McMullen, S. Shah and N. Rajmohan. "Methodology for a Robust Interactive Boundary Layer Solver". In: *AIAA Aviation 2019 Forum*. 2019.

[66] T.A. Talay. *Introduction to the aerodynamics of flight*. Tech. rep. NASA Langley Research Center Hampton, 1975.

Appendices

Appendix A

Derivation of the lift and drag coefficients

From the simulations, pressure and skin friction distributions are obtained. Using derivations in Anderson [39], the pressure and skin friction are transformed to lift and drag coefficients. Lift and drag are defined as the forces perpendicular and parallel to the onset flow, and can be non-dimensionalized into coefficients, similar to pressure and skin friction. Lift and drag are visualized in Figure A.1.

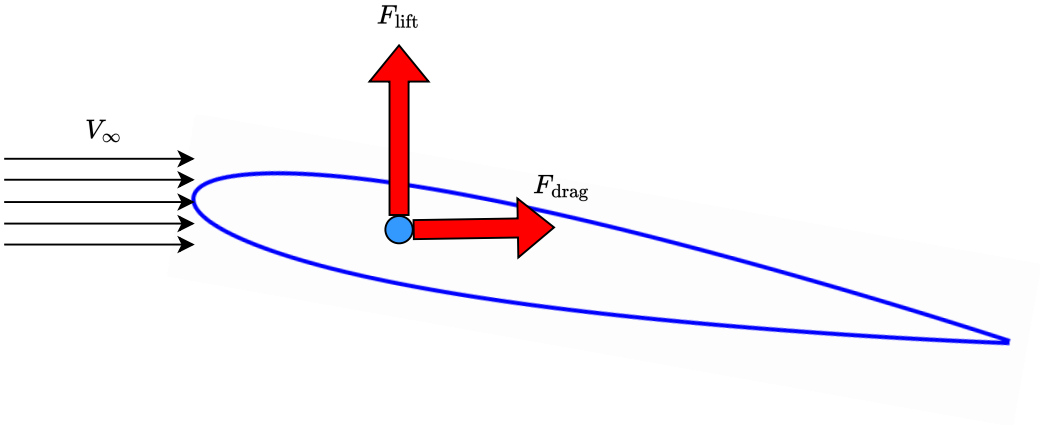

Figure A.1: Lift and drag on an airfoil.

As mentioned before, pressure and skin friction are obtained from the simulation. The formulas for all coefficients are given below.

$$q_\infty = \frac{1}{2}\rho V_\infty^2 \quad \rightarrow \text{dynamic pressure,} \tag{A.0.1}$$

$$C_p = \frac{p - p_\infty}{q_\infty} \quad \rightarrow \text{pressure coefficient,} \tag{A.0.2}$$

$$C_f = \frac{\tau}{q_\infty} \quad \rightarrow \text{skin friction coefficient,} \tag{A.0.3}$$

$$C_L = \frac{L}{q_\infty c} \quad \rightarrow \text{lift coefficient,} \tag{A.0.4}$$

$$C_D = \frac{D}{q_\infty c} \quad \rightarrow \text{drag coefficient.} \tag{A.0.5}$$

In the above equations, the forces are defined per unit span, as everything is dealt with in 2D. The variables p, τ and c denote the static pressure, the shear stress and the chord length, respectively. Taking Bernoulli's law,

$$p_\infty + \frac{1}{2}\rho V_\infty^2 = p + \frac{1}{2}\rho V^2, \tag{A.0.6}$$

which is valid in inviscid flow, and rewriting,

$$p - p_\infty = \frac{1}{2}\rho V_\infty - \frac{1}{2}\rho V = \frac{1}{2}\rho(V\infty^2 - V^2), \qquad \text{for incompressible flow.} \tag{A.0.7}$$

Equation A.0.7 can now be substituted in Equation A.0.2, giving

$$C_p = \frac{\frac{1}{2}\rho(V_\infty^2 - V^2)}{q_\infty} = \frac{\frac{1}{2}\cancel{\rho}(V_\infty^2 - V^2)}{\frac{1}{2}\cancel{\rho}V_\infty^2} = 1 - \frac{V^2}{V_\infty^2} = 1 - \left(\frac{V}{V_\infty}\right)^2, \tag{A.0.8}$$

the definition used in section 2.4. To obtain the lift and drag coefficients, firstly, the pressure and skin friction coefficients have to be transformed to pressure and skin friction forces. This is done by multiplication with the dynamic pressure, following Equation A.0.2 and Equation A.0.3. Hence,

$$p - p_\infty = C_p q_\infty, \qquad \tau = C_f q_\infty, \tag{A.0.9}$$

where $p - p_\infty$ will be substituted with a net pressure p_{net} for simplicity. The force per unit span is obtained by integrating both pressure and skin friction over a length, namely the airfoil contour. As such, a total pressure and skin friction force per unit span can be defined,

$$F_{pres} = \oint p_{net}\mathrm{d}s = \oint C_p q_\infty \mathrm{d}s, \qquad F_{skinf} = \oint \tau \mathrm{d}s = \oint C_f q_\infty \mathrm{d}s. \tag{A.0.10}$$

The orientation of these forces should be known to be able to derive lift and drag. Therefore, the unit vectors are needed. Pressure is always defined inward normal, and shear stress is defined tangential to the surface, in positive streamwise coordinate. For a NACA 0012 at 0° angle of attack, the unit vectors for pressure and skin friction can be seen in Figure A.2. Now, let i be the number of panels in which the airfoil is discretized. Each panel has its own inward normal unit vector and streamwise positive tangential unit vector. Each of these vectors has an x and a y component. The normal and tangential vectors of the ith panel are defined as

$$\overrightarrow{n} = [n_{i_x}, n_{i_y}], \qquad \overrightarrow{t} = [t_{i_x}, t_{i_y}]. \tag{A.0.11}$$

Now, the pressure and skin friction force can be written in a direction,

$$F_{pres} = \oint C_p q_\infty \mathrm{d}s \quad \rightarrow \quad \overrightarrow{F}_{pres} = q_\infty \sum_i S_i C_{p,i} \begin{bmatrix} n_{i,x} \\ n_{i,y} \end{bmatrix}, \tag{A.0.12}$$

$$F_{skinf} = \oint C_f q_\infty \mathrm{d}s \quad \rightarrow \quad \overrightarrow{F}_{skinf} = q_\infty \sum_i S_i C_{f,i} \begin{bmatrix} t_{i,x} \\ t_{i,y} \end{bmatrix}, \tag{A.0.13}$$

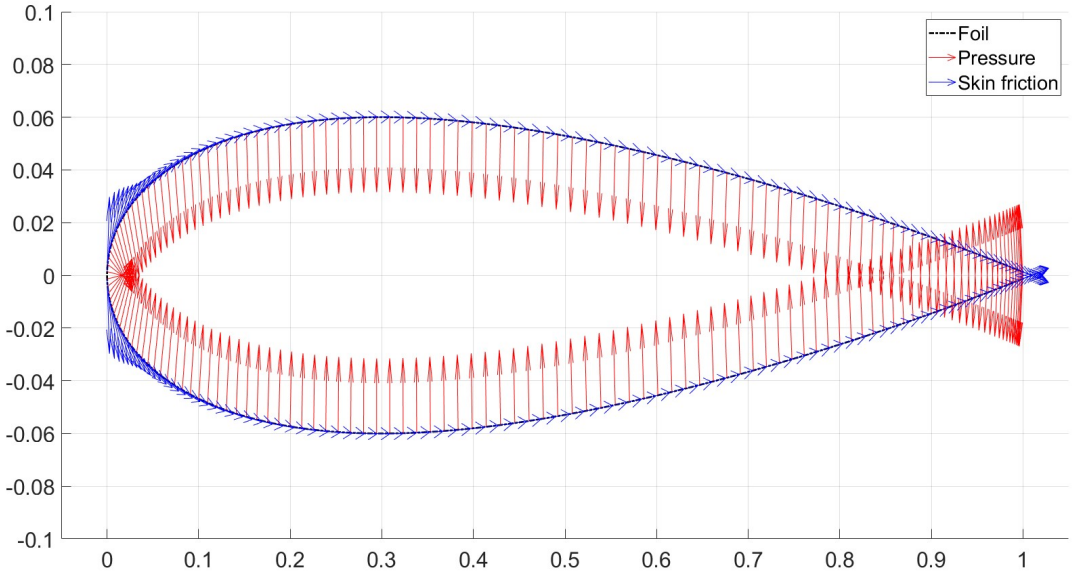

Figure A.2: Unit vectors for pressure and skin friction.

where the contour integral can be simplified by multiplying coefficient i, be it pressure or skin friction coefficient, with the length of panel i, and summing all of them to a total. This will give a resultant pressure and skin friction force, which can be decomposed into a normal and an axial force. The former is perpendicular to the chord c, while the latter is parallel to c. In the case of 0° angle of attack, adding the x and y components of pressure and skin friction forces will already be equal to the lift and drag forces. In the case of an angle, the upper definitions will give the normal and axial forces. To determine the lift and drag forces, which are perpendicular and parallel to the onset flow, the normal and axial forces have to be transformed by a rotation matrix, namely

$$R = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}, \quad \text{such that} \quad \begin{Bmatrix} F_L \\ F_D \end{Bmatrix} = \begin{bmatrix} R \end{bmatrix} \begin{Bmatrix} F_{norm} \\ F_{ax} \end{Bmatrix}, \tag{A.0.14}$$

where α is the angle of attack. Putting together Equation A.0.12, Equation A.0.13 and Equation A.0.14, one arrives at

$$\begin{Bmatrix} F_L \\ F_D \end{Bmatrix} = q_\infty \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \sum_i S_i \begin{Bmatrix} C_{p,i} n_{i_y} + C_{f,i} t_{i_y} \\ C_{p,i} n_{i_x} + C_{f,i} t_{i_x} \end{Bmatrix}. \tag{A.0.15}$$

Note that the y components are now mentioned first, as the normal force is mentioned as the first component of the vector to be rotated. The last step is to non-dimensionalize the lift and drag forces, thereby obtaining the lift and drag coefficients. Using Equation A.0.4 and Equation A.0.5, one arrives at

$$\begin{Bmatrix} C_L \\ C_D \end{Bmatrix} = \frac{\cancel{q_\infty}}{\cancel{q_\infty} c} \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \sum_i S_i \begin{Bmatrix} C_{p,i} n_{i_y} + C_{f,i} t_{i_y} \\ C_{p,i} n_{i_x} + C_{f,i} t_{i_x} \end{Bmatrix}. \tag{A.0.16}$$

Generally, the chord length is taken as 1, and can thus be discarded as well, yielding the final

equations for lift and drag coefficients,

$$\begin{Bmatrix} C_L \\ C_D \end{Bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \sum_i S_i \begin{Bmatrix} C_{p,i} n_{i_y} + C_{f,i} t_{i_y} \\ C_{p,i} n_{i_x} + C_{f,i} t_{i_x} \end{Bmatrix}. \tag{A.0.17}$$

Note that, for an angle of attack of 0°, the rotation matrix simply becomes the identity matrix, giving the more simplified equation

$$\begin{Bmatrix} C_L \\ C_D \end{Bmatrix} = \sum_i S_i \begin{Bmatrix} C_{p,i} n_{i_y} + C_{f,i} t_{i_y} \\ C_{p,i} n_{i_x} + C_{f,i} t_{i_x} \end{Bmatrix}. \tag{A.0.18}$$

It is interesting to note that, following Talay [66], for a streamlined body in a flow that is not separated, the drag caused by skin friction is the dominating component, and the effect of pressure drag is very small. In inviscid cases, pressure drag can even become negative, functioning as a thrust force on an airfoil. In any case, the drag force is mainly decomposed of the skin friction. However, for a separated flow, the pressure drag takes on a crucial parasitic role and surpasses the skin friction drag. Additionally, the principal contributor to lift is the pressure, and not so much the skin friction, which is generally two to three orders smaller than the pressure.

Appendix B

Additional results

This appendix shows results that are complementary to chapter 5, namely the result of the exactly symmetrical case.

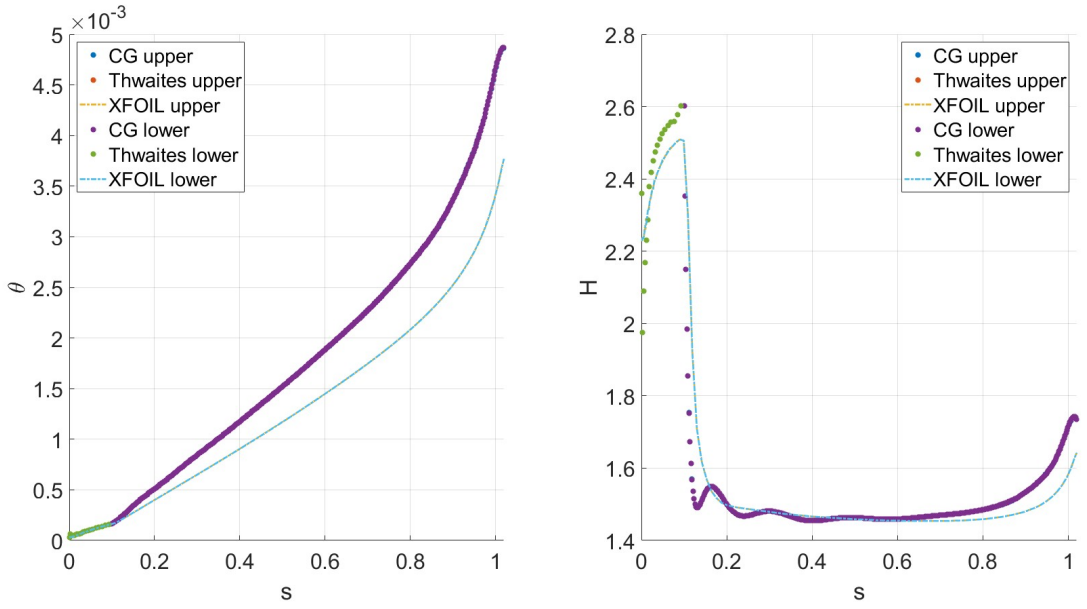

Figure B.1: The momentum thickness and shape factor results for NACA 0012 at 0°, approximated with a Lagrange polynomial of order 20 at Reynolds of $1 * 10^6$*, tripping at* $s = 0.1$*, compared with XFOIL.*

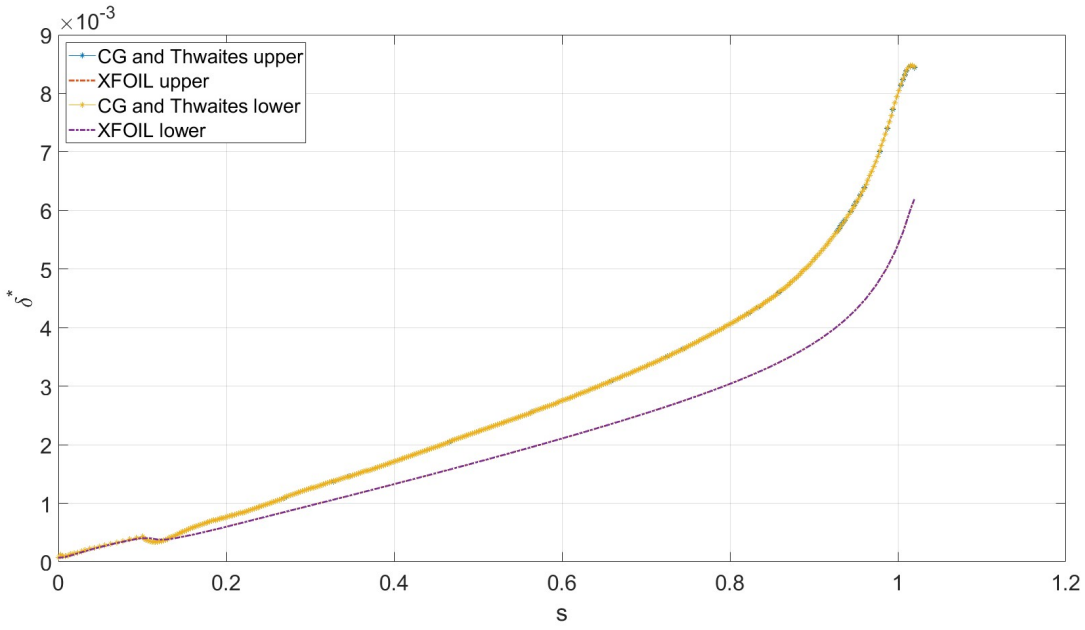

Figure B.2: The displacement thickness for NACA 0012 at 0°, approximated with a Lagrange polynomial of order 20 at Reynolds of $1 * 10^6$*, tripping at* $s = 0.1$*, compared with XFOIL.*

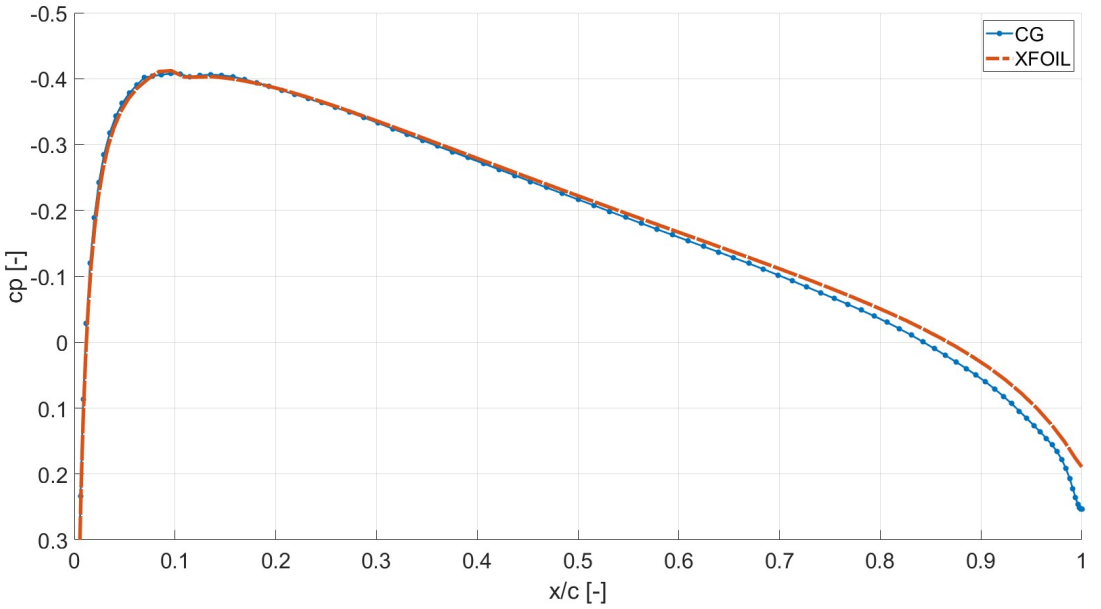

Figure B.3: Pressure distribution from NACA 0012 at 0°, tripping at $s = 0.1$*, compared with XFOIL; cut off at 0.3 for legibility.*

	BL solver	*XFOIL*
C_L	1.204e-6	0.000
C_D	0.0095	0.01061

Table B.1: Aerodynamic coefficients for NACA 0012 at 0°.