Laboratory for Turbulence Research in Aerospace and Combustion

The laminar boundary layer
It is not all about the wall

Enrique Millán Valbuena[1,2]
Julio Soria[2]

[1]*Delft University of Technology, Netherlands* [2]*LTRAC, Monash University, Australia*

Technical Report: LTRAC-TR 2018.001

Laboratory for Turbulence Research in Aerospace and Combustion
Department of Mechanical and Aerospace Engineering
Monash University
Melbourne, VIC 3800

The laminar boundary layer
It is not all about the wall

Enrique Millán Valbuena[1,2]
Julio Soria[2]

Contents

List of Figures

List of Tables

1 Preface and acknowledgements

This report is the conclusion of the Internship carried out at the Mechanical and Aerospace Engineering Department in Monash University, Melbourne. It consisted on a 8 month research project to study the laminar boundary layer with an alternative formulation to the usual Navier-Stokes equations in primitive variables, in order to answer the open questions that it entails. What was expected was completed, to write a robust code that would solve the often complex system of equations that any fluid motion modelling requires.

To be able to carry out this work at the other side of the world, as I am natural from Spain, was a chance and a luxury for which I am deeply grateful towards my parents and *Prof. Julio Soria* for the economical support and all that I have learned through the freedom in speech and act the environment he has built provides. His supervising strategy is unusual, but it teaches you more about yourself and how to strive in a world full of uncertainty that a hundred coaching sessions would ever do.

Additionally, I would like to thank *Prof. Miguel Hermanns Navarro* and *Prof. Gonzalo Rubio* for the insightful conversations that we had, providing me guidance and allowing me to take bigger steps in order to finish my project.

2 Executive Summary

The problem here addressed is the one of solving the Navier-Stokes equations in derived variables, particularly for the streamfunction and the vorticity. The aim is to be able to study a flow over a flat plate in a boundary layer scenario, that is, with an irrotational far-field away from the plate where one of the velocity components is given. We avoid the use of the traditional primitive variables, velocity and pressure, because the boundary conditions in the far-field are ill-conditioned. In that formulation, one cannot assure continuity and irrotationality at the same time, obliging the user to have huge computational domains to avoid contamination of results.

The main goals are to test the range of validity for the self-similar boundary layer theory, perhaps shading new light on its usage, and to study the properties of the flow for a boundary layer on the edge of separation.

In order to do so, we have discretized the non-linear system of equations that arise in this derived formulation via a Finite Differences method. This non-linear algebraic system is then solved using a Newton-Rapshon procedure. This step is of critical importance, since both equations need to be solved simultaneously to guarantee numerical and physical robustness. This is the case because no proper boundary condition can be derived through physical reasoning for the vorticity value or flux at the boundary with a rigid wall. To solve the constantly changing Jacobians of the linearized system that are used in this procedure, we have employed a *matrix-free* iterative linear solver, the Generalised Minimal Residual (GMRES).

The main issue faced by the project was how to deal with the boundary condition for the vorticity at the wall. We coded many other approaches beforehand: solving the biharmonic equation, using a transitory approach with artificial boundary conditions and even a boundary element method in order to enforce an integral constraint. At the end, the simultaneous solution of both equations proved to be the way.

Once the code was validated by means of the Method of Manufactured Solutions, some preliminary results on different pressure gradients over the flow could be obtained. We present a recirculation bubble, a favourable pressure gradient (FPG) and a zero pressure gradient (ZPG) boundary layers. We start to see here how some classical results from the boundary layer solutions are valid and others not so much. The physical domain corresponds to almost a hundred initial boundary layer thickness on the stream-wise direction, and about ten to twenty on the transversal one. Although useful remarks can be drawn from these results, larger domains still need to be studied.

The main conclusions that we can draw from this project are of technical and not so much of theoretical character. Coding a DNS code from almost scratch is not at

all trivial, specially if flexibility in the possible scenarios to study is a condition on its design. It does not only suffice to compute the derivatives and solve the linear system, it is challenging to understand the behaviour of the code, or if the real physics are been tracked and no spurious solutions are being introduced. In a sense, those who do DNS are very close to craftsmanship, where sometimes specific solutions are required, despite the general and abstract mathematical context of the all.

3 Introduction

The boundary layer is perhaps the most studied type of flow in the past century in the field of Fluid Mechanics. It was a major theoretical breakthrough, finally establishing a link between what experimentalist saw in their laboratory tunnels and what equations predicted. Indeed, flow movement around bodies can be divided in two regions: an internal one near the wall, the boundary layer, where viscous effects dominate and the no-slip condition is met, and an external one, where viscosity does not have a significant influence. However, despite the amount of research and works that have been produced and published regarding the subject, many of its secrets remain still unveiled.

When it comes to flow movement around bodies, the boundary layer encounters different pressure gradients as it moves downstream, induced by the effect of the geometrical curvature of the body. It is precisely this effect the one that allows airfoils to generate lift, but it can also lead to critical conditions for the operation and even existence of the boundary layer.

Indeed, when the flow is under an adverse pressure gradient, a deceleration in the external streamwise velocity $U_e(x)$ and a suction effect $V_e(x)$ outwards from the surface take place in the far-field. This alters the streamlines pattern, deflecting them away from the body, inducing an earlier transition from laminar to turbulent movement. Additionally, if the pressure gradient becomes strong enough, it can even separate completely the streamlines from the body. This phenomenon is known as *flow separation and reversal*, and it changes completely the flow properties and the forces exerted on the body, mostly raising the drag.

Despite the undesirable physical events that can take place under an adverse pressure gradient, engineering applications very often require their use to accommodate the coming flow. From leading edges in wing sections or turbomachinery blades, to diffusers in air-breathing engines, adverse pressure gradients are unavoidable. For this reason, it remains an active research topic nowadays.

From a theoretical point of view, many interesting analytical results for the laminar boundary layer can be derived when the equations are parabolized under the right scaling, and they can be found in most engineering textbooks [7]. However, when dealing with adverse pressure gradients, all of them hit inevitably with the same wall: flow reversal. In this scenario, information does not only propagate downstream, but also upstream. The terms that were neglected when the parabolized equations were derived, mainly the derivatives in the x-direction and the v-velocity components, are no longer small and thus the parabolic approximation is no longer valid.

3.1 Pressure gradients in a flat plate

When theory fails, we reach out for experimentation. Adverse pressure gradient flows are studied in laboratories with PIV techniques or in high performance computers with Direct Numerical Simulation codes [19].

This allows the fully Navier-Stokes equations to be solved, often in what is called the *primitive variables*: pressure p and velocity $\vec{v}$. In this field, an inverse relationship between geometry and physical detail exists. The simpler the geometry, the more details we are able to resolve. This is the reason behind the fact that most DNS codes solve flat plates or square cavities. Unfortunately, this necessary geometrical simplification to remove curvature complicates the enforcement of a desired and consistent pressure gradient in the simulation.

To overcome this difficulty, we are obliged to play with the velocity boundary conditions in the far-field. Often one of the velocities components is given and the flux for the other one is derived with the continuity equation $\nabla \cdot \vec{v} = 0$ or the planar irrotationality condition $\nabla \wedge \vec{v} = 0$, respectively:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \qquad \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} = 0$$

If $U_e(x)$ is given, which is quite common since most of the theoretical results are derived for that component, from the continuity equation we can obtain the vertical flux of $V_e(x)$ through the far-field boundary:

$$\frac{\partial V_e}{\partial y}(x) = -\frac{\partial U_e}{\partial x} \tag{1}$$

Instead, and more rare, if the $V_e(x)$ wants to be enforced, from the irrotationality condition one can obtain the vertical flux of $U_e(x)$:

$$\frac{\partial U_e}{\partial y}(x) = \frac{\partial V_e}{\partial x} \tag{2}$$

Both strategies enforce a pressure gradient on the flow. Yet nothing assures that the imposition of continuity will ensure irrotationality and viceversa. In order to avoid this inconsistent boundary condition from contaminating the results, large computational domains are used. The values at the wall are adequate, but their quality will degrade with distance from it.

Therefore, the main research question to be addressed is: *for a given $U_e(x)$, what is the $V_e(x)$ that satisfies both the continuity and the irrotationality conditions?*

3.2 Scope and challenges of the project

As a first step forward for the answer to this fundamental question, we shall not confront directly a three-dimensional turbulent boundary layer. Instead, we will start with the laminar boundary layer under some necessary simplifications. Consequently, we will be working with the implicit assumption that the regime of the flow will not affect the far-field boundary conditions.

Since the classical primitive variable strategy seems to be ill-posed in the far-field, we will change to a new set of variables where this inconsistency disappears. However, as it is customary with the Navier-Stokes equations, this will create new problems in other regions of the domain.

Not only the challenges will be theoretical, most of the computational framework to carry out the work had to be built from scratch. Previous numerical works have been done in these new set of variables, but we wish to take an approach that does not break down when the Reynolds number increases. Or at least if it does, that it shall happen because of physical reasons and not computational ones.

Finally, we must understand this work in the context of the first step in a research project. The time-frame in which it was developed was tight, for all that had to be understood and poured down into the computer.

4 Problem formulation

The path to solve the Navier-Stokes in other variables distinct from the primitive ones has been walked before many times [13], [18, 15, 14, 2, 17, 1, 6, 20]. Under certain assumptions, either in the viscous behaviour or the number of dimensions, we can find new sets of variables to solve the equations. Sometimes this sheds a brighter light over certain aspects that else remain hidden in the pressure-velocity formulation [9].

For our problem, we will be using the streamfunction-vorticity formulation. The streamfunction ψ is the function whose isolines coincide with the streamlines of the flow movement. The vorticity ω is a kinematic property linked with the rotation of the fluid particles. With these variables, continuity is given by definition and irrotationality can be imposed as a boundary condition.

There exist other formulations which would have allowed us to impose irrotationality, such as the velocity-vorticity formulation. One could fix a given $U_e(x)$, obtain $\partial V_e/\partial y$ from the continuity equation and additionally fix $\omega = 0$. However, this formulation does not inherently conserve mass, which introduces another major problem. Thus, it was avoided.

4.1 Framework and limitations

The physical scenario that will be solved is a boundary layer developing over a flat plate, so the equations will be written in Cartesian coordinates. The integration domain will be a rectangular box of length L and height y_e. These dimensions need to be taken in context with the length scale of the boundary layer. An horizontal length $L \sim 2.5$ can actually be equal to 30 boundary layer thickness units.

In order to use ψ as a dependent variable to describe the fluid motion, we need to assume incompressible flow and planar motion. Additionally, we have assumed a steady-state situation, but this hypothesis is not required for what follows. Under these assumptions, the velocity components can be rewritten in terms of partial derivatives of a function ψ. The aim with this function is to satisfy identically the differential equation for mass conservation:

$$u = \frac{\partial \psi}{\partial y}, \qquad v = -\frac{\partial \psi}{\partial x} \tag{3}$$

where we leave for the reader to verify the claim. The first equation that we derive is simply the definition of vorticity ω in terms of the streamfunction. When the new definition for the velocity components is introduced, the Poisson equation for ψ with $-\omega$ as a forcing term arises. The second equation, the transport equation for vorticity

ω, is obtained by taking the curl of the momemtum equation [10]. For reasons that will become clear in Section 5, we have labeled both equations as F_1 and F_2 respectively:

$$\nabla^2\psi + \omega := F_1(\psi, \omega) = 0 \tag{4}$$

$$\nabla^2\omega - Re\left(\frac{\partial\psi}{\partial y}\frac{\partial\omega}{\partial x} - \frac{\partial\psi}{\partial x}\frac{\partial\omega}{\partial y}\right) := F_2(\psi, \omega) = 0 \tag{5}$$

Evident to the eye, but necessary to be pointed out, the pressure term has disappeared from the equations. This removes some important issues concerning the boundary conditions when solving the equations for the primitive variables, but it will also have an impact in the way we handle the outflow, later discussed in Section 4.2.4.

The Reynolds number Re has been changed from its usual position in the denominator of the diffussive term. This has been done so that the equations solved do not become singular in the Stokes limit for $Re \to 0$. When this parameter is based on the unit length and velocity:

$$Re = \frac{1}{\nu}$$

the equations are considered to be non-dimensional. It is important to take this into account, since it could lead to mismatches between the viscosity of the inlet profile and the one encountered by the flow in the integration domain. This situation is physically possible, but it would create an unecessary region at the inlet where the flow needs to adapt to the new environment.

The major difficulties these equations present are their non-linearity, the explicit coupling through the velocity components and the implicit coupling in the boundary conditions at the wall that shall be discussed in Section 4.2.

Unfortunately, when it comes to fluid physics modeling, they have a hidden limitation caused by the planar motion constraint: they cannot model turbulence. The identical vanishing of the vortex stretching term $(\vec{\omega} \cdot \nabla) \cdot \vec{v}$ in the vorticity transport Equation (5) impedes the energy cascade that happens in turbulent flow movement.

4.2 Description of the boundary conditions

The boundary conditions of this system are quite straightforward to derive from a physical point of view. However, they can lead to headaches when it comes to their numerical implementation.

4.2.1 Inlet

We shall start by the *inlet* boundary conditions. The streamfunction and vorticity profiles are given as Dirichlet conditions:

$$\psi_{\text{in}} = \psi_{\text{BL}}(y), \qquad \omega_{\text{in}} = \omega_{\text{BL}}(y) \tag{6}$$

where $\psi_{\text{BL}}(y)$ and $\omega_{\text{BL}}(y)$ can be obtained from numerical solutions of the parabolic approximations for the boundary layer. The streamfunction defines uniquely the u-velocity component. With ψ_{in} fixed, the vorticity profile defines a unique flux of v-velocity.

Some natural choices could be solutions like the Blasius or the Falkner-Skan profiles. But a very interesting one could be Howarth's solution for a boundary layer near the point of separation. With nowadays computers, it would take no effort to solve the equations he faced with a higher degree of accuracy. The aim would be to shorten the length of numerical domain required to get to the separation point.

4.2.2 Far-field

The following boundary that is easy to deal with is the *far-field*. The irrotationality condition is a Dirichlet condition and the external velocity $U_e(x)$ is a Von Neumann condition:

$$\omega_\infty = 0, \qquad \left.\frac{\partial \psi}{\partial y}\right|_\infty = U_e(x) \tag{7}$$

Another possible combination would be to use a Dirichlet condition for both variables:

$$\omega_\infty = 0, \qquad \psi_\infty = \psi_E(x)$$

where $\psi_E(x)$ could come, for example, from a solution to the inviscid Euler equations.

4.2.3 Wall

After that, the difficult boundary conditions arrive. At the *wall*, the no-slip condition has to be satisfied. However, from this kinematic condition we can only obtain boundary conditions for the streamfunction:

$$\psi_w = 0, \qquad \left.\frac{\partial \psi}{\partial y}\right|_w = 0 \tag{8}$$

For the vorticity at the wall, we can relate the influx with the pressure gradient and the Re number by evaluating the x-momentum equation at the wall and combining it

with the definition of vorticity:

$$\left.\frac{\partial \omega}{\partial y}\right|_w = -Re \left.\frac{\partial p}{\partial x}\right|_w \tag{9}$$

where the derivation is left to the reader. At a first glance, this could solve the problem for the boundary conditions at the wall, but what is the appropriate value for the pressure gradient at the wall such that no-slip is enforced?

One could think about using the pressure gradient from the far-field, but that will only hold when the parabolic boundary layer assumptions remain true, which will not be the case with an adverse pressure gradient where the flow separates and v becomes important. This theoretical issue will be addressed with the choice of the numerical method.

4.2.4 Outlet

Finally, we come to the Holy Grail of Computational Fluid Mechanics: the outflow. How to make the flow leave the domain without losing the physical meaningfulness of the whole movement is a fine art.

In primitive variables, the numerical procedures to make the flow leave the domain without becoming unstable have been thoughtfully studied and explained. Very often, the procedure employed consists of imposing the total derivative of the velocity components equal to zero.

$$\frac{\partial \vec{v}}{\partial t} + U_c \frac{\partial \vec{v}}{\partial x} = 0$$

where U_c is a parameter that is often used to enforce globally mass conservation.

Unfortunately, this approach can make the outflow and a close region nearby meaningless from a physical point of view, but it allows the simulation to be stable and have a large valid domain where the physical laws are correctly solved.

Instead, with a weak formulation of the primitive equations, the pressure and the stress tensor can be used to fix a robust condition for the outlet [3], but as stated before, we do not have the pressure any more in our formulation.

For what concerns us, we will be using an outflow derived through physical reasoning by Briley almost 40 years ago [4]. The clever idea follows: when the Navier-Stokes equations (4) and (5) are parabolized with the scaling $1/\sqrt{Re}$, the parabolic boundary layer equations arise, since the x-derivatives drop out. So we enforce this fact by setting in the outflow the conditions:

$$\frac{\partial^2 \omega}{\partial x^2} = 0, \qquad \frac{\partial^2 \psi}{\partial y^2} + \omega(L, y) = 0 \tag{10}$$

Such an astute move makes the outflow physically meaningful, and the simulation stable.

To sum up, we have the right amount of boundary conditions for this problem, but they are not distributed evenly. We have five conditions for the Laplacian operator and only three for the transport equation. To bind them all together when we chose a numerical method, we shall pick one that solves the equations simultaneously. More on this and the boundary conditions in the Section 5.

4.3 Linearized equations

In order to prepare the differential equations and boundary conditions for their numerical implementation and solution, we linearize them. Each dependent variable can be expanded as an equilibrium solution with an added increment:

$$\psi = \psi_0 + \delta\psi$$
$$\omega = \omega_0 + \delta\omega$$

Once this terms are introduced in the equations and that the second order terms are neglected, we find the following linear system of PDEs in terms of $\delta\psi$ and $\delta\omega$:

$$\nabla^2\delta\psi + \delta\omega = -F_1\left(\psi_0, \omega_0\right) \tag{11}$$

$$\nabla^2\delta\omega - Re\left[\left(\frac{\partial\delta\psi}{\partial y}\frac{\partial\omega_0}{\partial x} + \frac{\partial\psi_0}{\partial y}\frac{\partial\delta\omega}{\partial x}\right) - \left(\frac{\partial\delta\psi}{\partial x}\frac{\partial\omega_0}{\partial y} + \frac{\partial\psi_0}{\partial x}\frac{\partial\delta\omega}{\partial y}\right)\right] = -F_2\left(\psi_0, \omega_0\right) \tag{12}$$

$$\tag{13}$$

These equations can be expressed in a matrix form, using the notation expressed in Equations (4) - (5):

$$\begin{bmatrix} \nabla^2 & I \\ -Re\left(\frac{\partial()}{\partial y}\frac{\partial\omega_0}{\partial x} - \frac{\partial()}{\partial x}\frac{\partial\omega_0}{\partial x}\right) & \nabla^2 - Re\left(\frac{\partial\psi_0}{\partial y}\frac{\partial()}{\partial x} - \frac{\partial\psi_0}{\partial x}\frac{\partial()}{\partial x}\right) \end{bmatrix}\begin{pmatrix}\delta\psi \\ \delta\omega\end{pmatrix} = -\begin{pmatrix}F_1(\psi_0,\omega_0) \\ F_2(\psi_0,\omega_0)\end{pmatrix} \tag{14}$$

This matrix represents only the continuous differential operators. It has to be completed with boundary conditions for $\delta\psi$ and $\delta\omega$.

4.3.1 Linearized boundary conditions

Using the same expansion, we have to linearize the boundary conditions. This will result in Dirichlet and Von Neumann boundary conditions for the functions $\delta\psi$ and $\delta\omega$ as a function of the equilibrium states ψ_0 and ω_0. The boundary conditions are now listed starting with the inlet, followed by the far-field, the wall and finally the

outlet:

$$\delta\psi(0,y) = \psi_{\mathrm{BL}}(y) - \psi_0(0,y)$$
$$\delta\omega(0,y) = \omega_{\mathrm{BL}}(y) - \omega_0(0,y)$$

$$\frac{\partial\delta\psi}{\partial y}(x,y_e) = U_e(x) - \frac{\partial\psi_0}{\partial y}(x,y_e)$$
$$\delta\omega(x,y_e) = 0 - \omega_0(x,y_e)$$

$$\frac{\partial\delta\psi}{\partial y}(x,0) = 0 - \frac{\partial\psi_0}{\partial y}(x,0)$$
$$\delta\psi(x,0) = 0 - \psi_0(x,0)$$

$$\frac{\partial^2\delta\psi}{\partial y^2}(L,y) + \delta\omega(L,y) = -\left(\frac{\partial^2\psi_0}{\partial y^2}(L,y) + \omega_0(L,y)\right)$$
$$\frac{\partial^2\delta\omega}{\partial x^2}(L,y) = 0 - \frac{\partial^2\omega_0}{\partial x^2}(L,y)$$

Another option could have been to set the increments corresponding to the Dirichlet boundaries to zero, and force the initial equilibrium states ψ_0 and ω_0 to have the desired values in the boundary. However, we prefer this approach for two reasons: the boundary conditions are coherent with the rest of the body of equations and in the case that we would like to introduce suction at the wall or a different value for the vorticity in the far-field it is pretty straightforward.

Another advantage of this technique is that it could allow us to work with non-linear boundary conditions. For instance, it could be the case for instance that we wanted to impose a given pressure gradient $\left.\frac{\partial p}{\partial x}\right|_\infty$ and not directly $U_e(x)$.

5 Numerical method and solution

For the previously derived system of equations and boundary conditions, no closed analytical solution exists. There are some promising works that use symbolic software packages to compute Taylor expansions of the dependent variables [16]. However, all those solutions are constrained to specific scenarios and have a small domain of validity.

Therefore, we have to discretize the continuous operators in order to find a numerical solution. The discretization method chosen is the Finite Difference Method, which approximates the differential operators as algebraic sums of nodal values of the functions. When this approximation is introduced in the governing equations, it would lead to a non-linear algebraic system of unknowns, which we solve through a Newton-Rapshon iterative process. The discrete version of the equations and the boundary conditions in the domain are the functions whose zeroes we are after. Not surprisingly, this is equivalent to find the solution for the linearized equations presented before in Section 4.3.

5.1 Differential problem discretization

Once the analytical problem has been formulated, the numerical methods with which the equations are going to be solved are presented, through a discretization of the continuous functions that represent the solution.

To discretize the problem consists of representing the work domain through a set of discrete points:

$$\begin{aligned} x \in [0, L] &\rightarrow \{x_i\} \quad \text{with} \quad i = 0, \ldots, N_x \\ y \in [0, y_e] &\rightarrow \left\{y_j\right\} \quad \text{with} \quad j = 0, \ldots, N_y \end{aligned}$$

called interpolation nodes, where $x_0 = 0$, $x_{N_x} = L$ and $y_0 = 0$, $y_{N_y} = y_e$. We take as the unknowns of the problem the values of ψ and ω in those nodes. Then an interpolating polynomial is constructed whose value is forced to coincide with that of the solution functions in the nodes and is used to evaluate or approximate both the derivatives that appear in the problem.

In what follows, the details will be explained for a one-dimensional function, since it would be cumbersome and redundant to do it for both directions, since the generalization is quite straightforward and this problem does not have any crossed derivatives.

5.1.1 Piecewise interpolation

Historically, the first method of polynomial approximation that was used was an interpolation with a polynomial $I(x)$ that passed through all the nodes of the domain. However, one can also choose to use a variation: the piecewise polynomial interpolation.

The piecewise polynomial interpolation consists of representing a f function with a set of interpolating polynomials $I_i(x)$ of degree $q \leq N$, instead of a single polynomial covering the entire domain.

This approach to the problem of interpolation involves the use of a set of subdomains σ_i that do not overlap and whose union is the entire working domain. This set of subdomains is usually referred to in the literature as a domain partition. Thus, each polynomial $I_i(x)$ will only be valid in its subdomain σ_i that includes the node x_i.

5.1.2 Interpolation molecules

To facilitate the nomenclature of certain expressions that will be developed later, the s_i seeds that are used to reference nodes are defined. In this way, the interpolating polynomials $I_i(x)$ will use from the node s_i to the node $s_i + q$ to build the polynomial. It should be clarified that the subdomain σ_i of validity of polynomials does not coincide with the domain defined by the nodes used in the interpolation[11], it will be smaller.

Figure 1 shows a particular case of piecewise polynomial interpolation with a polynomial of order $q = 6$ for a number of nodes $N + 1 = 11$. The interpolation of the function is carried out through molecules of nodes of size $q + 1$, and it is verified that the validity intervals σ_i of each interpolating polynomial are smaller than the subdomains delimited by the nodes employed for the interpolation. In addition, for points far away from the boundary, node molecules take the same number of points on both sides of the interpolation node when q is even, which is usually referred to as centred schemes. On the other hand, near the contour of the domain, more points will be taken towards the interior of the domain to perform the approximation, achieving to use only interpolation nodes within the domain.

5.1.3 Mesh

The type of mesh employed is borrowed from the work of M. Hermanns [11], with whom the author already worked during his Bachelor Thesis. We shall not go into further detail about how and why this mesh is constructed, since it already has a paper of its own. The important thing worth pointing out is that it is equispaced throughout most of the domain, but it will cluster more points towards the boundary

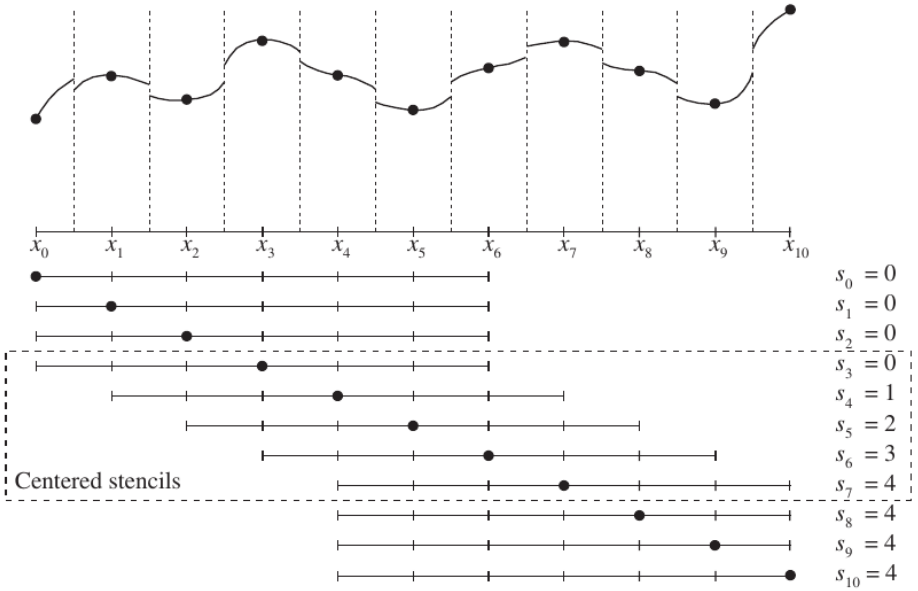

Figure 1: Particular case of piecewise polynomial interpolation with a polynomial of order $q = 6$ for a number of nodes $N+1 = 11$. The interpolation of the function is carried out through molecules of nodes of size $q+1$, and it is verified that the validity intervals σ_i of each interpolating polynomial are smaller than the subdomains delimited by the nodes employed for the interpolation. Near the contour of the domain, more points will be taken towards the interior of the domain to perform the approximation, achieving to use only interpolation nodes within the domain.

of the domain. The stretching will become more acute depending on the interpolation degree.

5.1.4 Evaluation of derivatives

To obtain the numerical expression of the derivative, the previously defined interpolator is analytically derived. In this way, the approximation of the derivative of a function in a point can be expressed as a linear combination of the images of the function around that point. The coefficients that multiply the images to form the linear combination are called weights and are denoted by w_{ij}. The weights have two indexes, the first one marks in which node the derivative is being calculated and the second one moves inside the nodes that are used to form the derivative:

$$f'(x_i) \simeq \sum_{j=0}^{q} w_{ij} \cdot f\left(x_{s_i+j}\right) \tag{15}$$

As for the interpolating polynomial, the seeds s_i are used to mark which nodes are used for the approximate calculation of the derivative, with the node s_i the first and

$s_i + q$ the last.

As an example to clarify the ideas, if we were using a centred scheme on an equispaced grid of mesh parameter h, the discrete derivative would be evaluated as:

$$f'(x_i) \simeq \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + O\left(h^2\right) \tag{16}$$

and so the vector of weights to compute the sum would be:

$$w_{i0} = -\frac{1}{2h}, \qquad w_{i1} = 0, \qquad w_{i2} = \frac{1}{2h}$$

Naturally, higher order derivatives will have other weights.

5.2 Solution of the algebraic system of equations

The matrix shown in Equation (14) corresponds to the Jacobian of the non-linear system. Thus, each iteration of the Newton-Raphson procedure can be expressed like:

$$[JF(\psi_0, \omega_0)]\delta\vec{x} = -\vec{F}(\psi_0, \omega_0) \tag{17}$$

$$\vec{x}_{n+1} = \vec{x}_n + \gamma\delta\vec{x} \tag{18}$$

where $\vec{x} = (\psi, \omega)^T$ is the unknowns vector and represents the nodal values of the discrete functions. It has been chosen to organize the unknowns vector such that the two values $\delta\psi_{ij}$ and $\delta\omega_{ij}$ at each node are placed one after the other.

The matrix $[JF(\psi_0, \omega_0)]$ changes in each iteration when the new equilibrium point is updated. It is not symmetric due to the node distribution and the boundary conditions. Additionally, we have introduced a relaxation factor γ to avoid divergence of the method due to the high amount of degrees of freedom.

To deal with the changing matrix on each iteration and to cope with the high demand of storage this matrix would demand, we chose to use an iterative solver. The Krylov-based Generalised Minimal Residual (GMRES) is chosen, since it can deal with non-symmetric matrices [12], [5]. This method was used in [8] and proved to solve effectively the boundary condition problem with vorticity for high Reynolds numbers.

5.2.1 Numerical implementation of the boundary conditions

Typically, when working with Finite Differences Methods, the boundary conditions are fixed at the nodes corresponding to the boundaries.

Regarding the Dirichlet type conditions, several ways exist to implement the boundary values. Some prefer to remove the algebraic known values from unknown vector and pass them to the other side, into the right-hand side. Other set the corresponding row in the matrix with all zeros except the entry corresponding to the value, where a multiplicative factor c is set to avoid un excess of difference between the entries in the matrix. This factor c is also multiplied on the RHS of the equations.

With respect to the Von Neumann boundary conditions, it is not uncommon the use of ghost nodes outside the domain to preserve the centered character of the scheme, and thus the order of interpolation. However, with the used distribution of nodes, it is not necessary to employ such a technique. The order is preserved and the scheme is stable even if high and uneven interpolation nodes are used.

The no-slip Von Neumann boundary condition is imposed on the nodes corresponding to the values of vorticity $\delta\omega$ at the wall. This small detail obliges us to solve the equations simultaneously, but it removes all constraint from the mesh, the interpolation order and the type of discretization used. Additionally, the code will not break down for high Reynolds numbers unless numerical accuracy is compromised.

6 Preliminary comment for the physical solutions

Since the code has been written from scratch, it is very important at the beginning to be very rigorous when validating the results. Because we are working with fluid motion, there are some integral values that can be derived from the equations of motion that have to be preserved, such as the mass. Therefore, one of the figures of merit that we shall be presenting in the following sections will be the value for the following sum of integrals:

$$I_{\text{mass}} = \int_0^{y_e} [u(L, y) - u(0, y)] \, \mathrm{d}y + \int_0^{L} [v(x, y_e) - v(x, 0)] \, \mathrm{d}x$$

This conservation metric is very useful to validate at once several aspects of the numerical implementation of the system of equations. First of all, even if we are using the streamfunction formulation and the conservation of mass is given on paper, the outlet could create problems. Because we have enforced two terms of the equations to be zero, this could create a situation where mass is not preserved, thus making this integral an important flag to validate the code. Secondly, to compute it we need to use the derivatives of the streamfunction, so it will indirectly measure the accuracy of the Finite Differences implementation.

7 Laminar separation and reattachment

In this section we present a scenario inspired in the work realized by Briley [4]. We pursue the creation of a laminar recirculation bubble. Back in the days, Briley realized this first study using the method of Finite Differences as well, but he addressed the issue of the boundary conditions by means of a modification of the interpolant. His idea was to introduce a ghost node whose value he would adjust such that the no-slip condition was enforced. Later, the value of vorticity at the wall could be introduced by using the wall expression for vorticity in terms of the streamfunction:

$$\omega_w = -\frac{\partial^2 \psi}{\partial y^2}$$

This technique has been widely used by many other authors [21], and it seems to work for low Reynolds numbers and low number of discretization nodes. If any of these two is augmented, problems might arise due to the Runge phenomenon that appears when high order polynomials are used on equispaced grids.

The parameters for this simulation are given in Table 1. The Reynolds number is

low, since we were interested to see how the code would deal with a big recirculation bubble. The interpolation degree q is mildly high, and as expected and reached with satisfaction, the mass integral is close to machine accuracy.

Table 1: Recirculation bubble simulation parameters.

L	y_e	N_x	N_y	Re	q	$\log I_{\text{mass}}$
5	0.9	70	60	500	8	-7.5

7.1 Boundary conditions at the inlet and far-field

For the streamfunction and the vorticity shapes at the inlet we have used the results from the Blasius parabolic solution at the given Reynolds.

For the far-field boundary condition, we have applied an irrotationality condition to the vorticity, and a piecewise $U_e(x)$ such that a strong deceleration takes place at the beginning, enough to produce a laminar separation of the flow. Towards the end of the domain we introduce a strong acceleration to force reattachment in order to make the code stable and converge to a solution.

$$U_e(x) = \begin{cases} (x_0 + x)^{-m}, & \text{if } 0 \leq x/L < 0.65 \\ U_1 \left(\frac{x}{x_1}\right)^m, & \text{if } 0.65 \leq x/L \leq 1 \end{cases} \tag{19}$$

where $m = -0.15$ and the pair (U_1, x_1) corresponds to the velocity value at the change of regime and the point of change. In Figure 2 the velocity profile is shown. We point out that the profile is continuous but not differentiable. This was made on purpose, to see and understand how this could affect the functioning of the code.

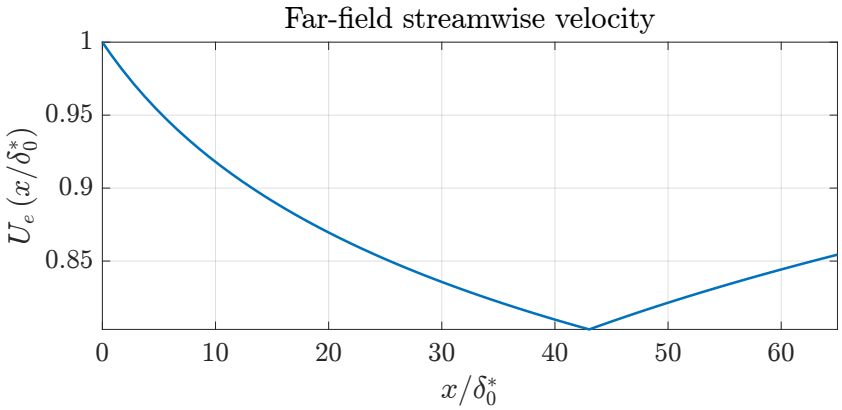

Figure 2: Stream-wise velocity component of the external field $U_e(x)$. We can easily appreciate that it is continuous but not differentiable, which was made on purpose to test the robustness of the code.

7.2 Results

In this section we show some figures with the results. Since our goal is to study the boundary layer on the edge of separation, it is encouraging to realize that the code is stable with massive separations. This means that as long as the flow is reattached when it exits the domain, we can allow for simulations where small separations occur.

In Figure 3 the vorticity distribution and the streamlines pattern are shown. In blue we have highlighted the limiting streamline below which the flow is recirculation, generating a bubble. The axes have been scaled with the initial boundary layer thickness δ_0^*. It is interesting to see that despite the fact that we have barely located one node per initial boundary layer thickness, the flow pattern seems to be perfectly resolved. This is due to the high order interpolation used to discretize the equations.

7.2.1 Velocity gradient in the far-field and pressure gradients

Two other interesting figures of merit worth taking a look at are $\left.\frac{\partial u}{\partial y}\right|_{y\to\infty}$ and the pressure gradients in the stream-wise direction. In Figure 4 we have plotted all of them. Regarding the velocity gradient, it is interesting to see how it goes through zero three times in the whole domain. At the beginning, due to the assumptions introduced when computing the Blasius solution, somewhere after the recirculation bubble and finally at the end of the domain. The fact that this happens at the end is satisfactory, as it is an implicit condition that has been fixed with the parabolic outflow.

When it comes to the pressure gradients, it is interesting to see how at the inlet and almost until the change of regime, the suction velocity is so important that it should not be neglected when computing the pressure gradient in the far-field. Only at the outlet both expressions seem to match:

$$\frac{\partial p_e}{\partial x} = -U_e \frac{\partial U_e}{\partial x} + O\left(\frac{1}{Re}\right) \tag{20}$$

$$\frac{\partial p_e}{\partial x} = -\left(U_e \frac{\partial U_e}{\partial x} + V_e \frac{\partial V_e}{\partial x}\right) \tag{21}$$

When it comes to the pressure gradient at the wall, it seems to plateau at the recirculation bubble. This leads to the following conclusion: when a recirculation bubble happens, all the velocity profiles along the separated region collapse into the same line near the wall. Let's reformulate this for the sake of clarity: due to the nature of the velocity gradient, at each station the boundary layer velocity profile near the wall can be approximated by a straight line. But this line will change with the stream-wise position, except for the separated region, where the line is approximatively the same.

This is the reflected in the momentum equation when the wall kinematic conditions are applied:

$$\left.\frac{\partial p}{\partial x}\right| w = \frac{1}{Re}\frac{\partial^2 u}{\partial y^2} \tag{22}$$

If $\left.\frac{\partial p}{\partial x}\right| w = 0$, then the velocity $u(y)$ is given by:

$$u(y) = Ay \tag{23}$$

once the no-slip condition has been applied. Since $\left.\frac{\partial p}{\partial x}\right|_w$ remains almost zero along the recirculation bubble, in Figure 5 we can see this phenomenon, where the black lines at the beginning differ quickly in the slope, whereas many lines seem to collapse around the same one in the reversed region of the bubble.

7.2.2 Self-similarity with flow reversal

Finally, we show a very interesting picture concerning the property of self-similarity and how it is portrayed when a recirculation bubble appears in the domain. In Figure 6 self-similar profiles of the stream-wise velocity component are shown. In Table 2 the colour code is given. It is interesting to see that the profiles before and after do not collapse identically one on top of each other, but they do *gather* around the same region of the self-similarity plot. Inside the bubble, the velocity profiles tend to collapse as well towards the same region, but it is clearly different from the one where the flow is not reversed.

Table 2: Self-similar velocity profiles colour code in the recirculation bubble.

Before	Bubble	After
Red dashed	Green cont.	Blue dotted

8 Results on self-similar boundary layer solutions

We shall now present the results of a boundary layer scenario with a favourable pressure gradient and a zero pressure gradient. The idea behind these results is to understand how much influence do the inlet and the outlet have on the rest of the domain.

The parameters for these simulation are given in Table 3. The Reynolds number is still small, but sufficient to test some of the boundary layer properties. Additionally, we can see that the mass conservation integral decreases for the ZPG. This is possibly due to the small amount of nodes in the y-direction, but it is interesting to note that despite that handicap, it is still well preserved. For the FPG we have picked a smaller

Table 3: Boundary layer with FPG simulation parameters.

Case	L	y_e	N_x	N_y	Re	q	$\log I_{\text{mass}}$
FPG	5	0.5	60	60	1000	8	-7.5
ZPG	5	1.0	80	50	1000	8	-6.5

physical domain, since the boundary layer is becoming thinner and not growing.

8.1 Remarks on the inlet

For the inlet profile we are introducing the boundary layer profile from the Blasius solution. Two issues arise with the inlet profile: matching the Reynolds number and the vorticity value in the far-field.

The first problem has to do with the fact that since the Blasius profile is obtained for a given Re which grows as we get further and further away from the stagnation point, it will not match perfectly the viscosity conditions in the numerical domain. Thus, a region of the inlet will suffer a small transition due to the fact that the flow comes from a different viscosity to a new one. This is physically possible, since we could have a flow that changes from water to oil, but it introduces undesired dynamics for our study.

Secondly, the classical solution is not irrotational in the far-field, so this is going to introduce an extra effect in the form of an artificial pressure gradient in the far-field. This problem could be solved by taking a profile in the middle of the flow of a previous simulation, which does satisfy the boundary condition. However, this will not fix the problem of the Reynolds number, so we have decided to leave it as it is for the moment.

8.2 Favourable pressure gradient (FPG)

We shall use the Falkner-Skan $U_e(x)$ for which self-similarity should be preserved:

$$U_e(x) = (x_0 + x)^{0.2} \tag{24}$$

For this solution, there are several aspects that we are going to consider. The first one is the momentum thickness θ, shown in Figure 7. We have introduced the Blasius solution as a reference, together with the momentum thickness expected from the integral method due to Thwaites:

$$\theta^2(x) = \frac{a}{ReU_e^b(x)}\left[\frac{1}{m(b-1)+1}\left((x_0+x)^{m(b-1)+1} - 1\right) + \frac{Re\theta_0^2}{a}\right] \tag{25}$$

where $b = 6$, $a = 0.45$ and θ_0 is the momentum thickness at the inlet. We can see that the slope of the momentum thickness is similar if not the same for the numerical solution and for the theoretical result. We believe that the difference by a constant lies in those irregularities introduced by the mismatch in the inlet. The fact that the slope is so similar serves as an index of good physical representation of the physics in the code.

Secondly, we can take a look at the self-similar profiles. We can see that the pressure gradient introduces some slight variations, but self-similarity is achieved across most of the domain, even at the outlet. This is not necessarily surprising, since precisely the terms that we have neglected in that boundary are the ones that classical theories neglect to obtain their self-similar solutions.

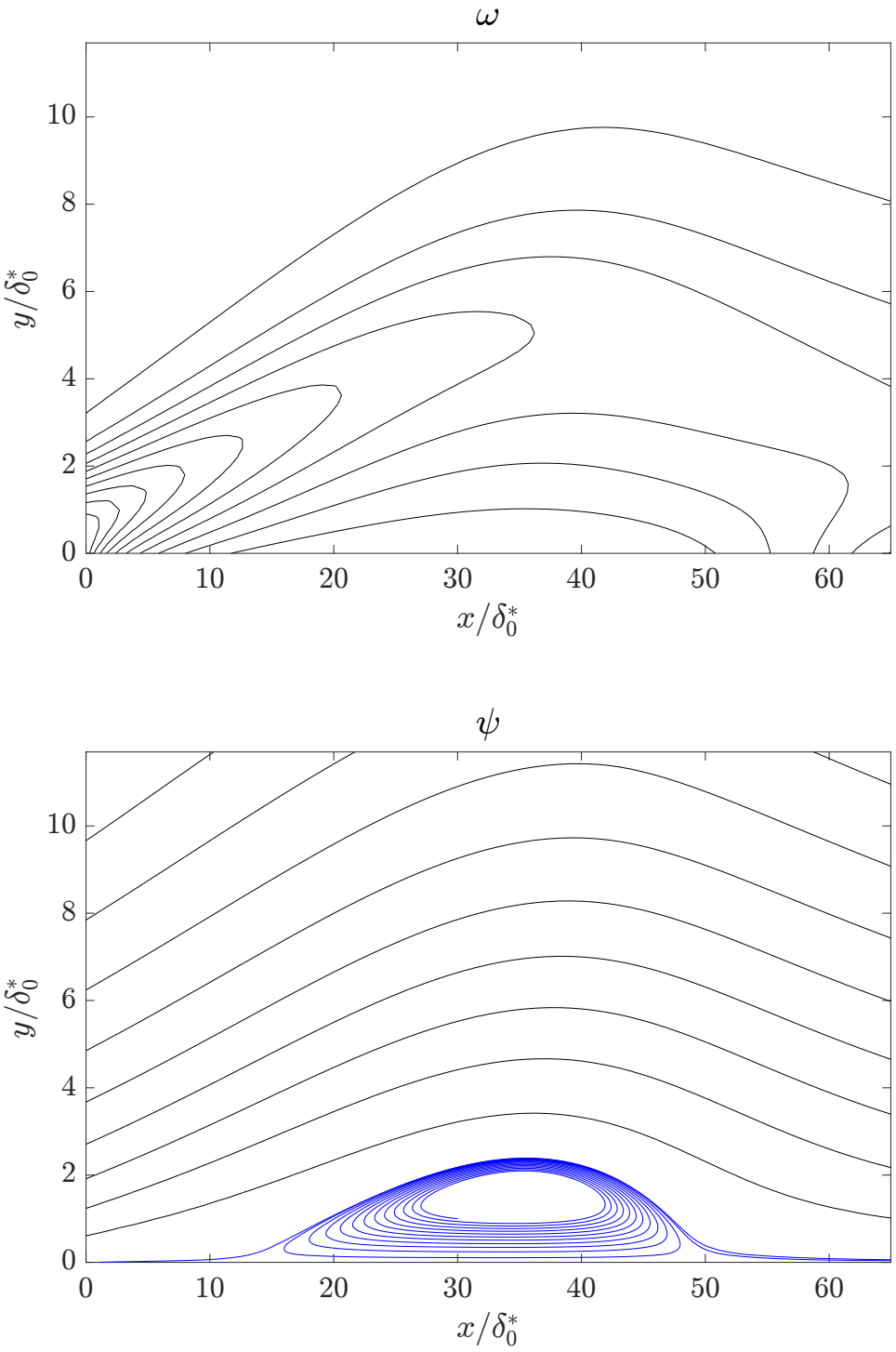

Figure 3: **Top:** vorticity distribution. **Bottom:** Streamlines of the flow. In blue the recirculation bubble has been highlighted.

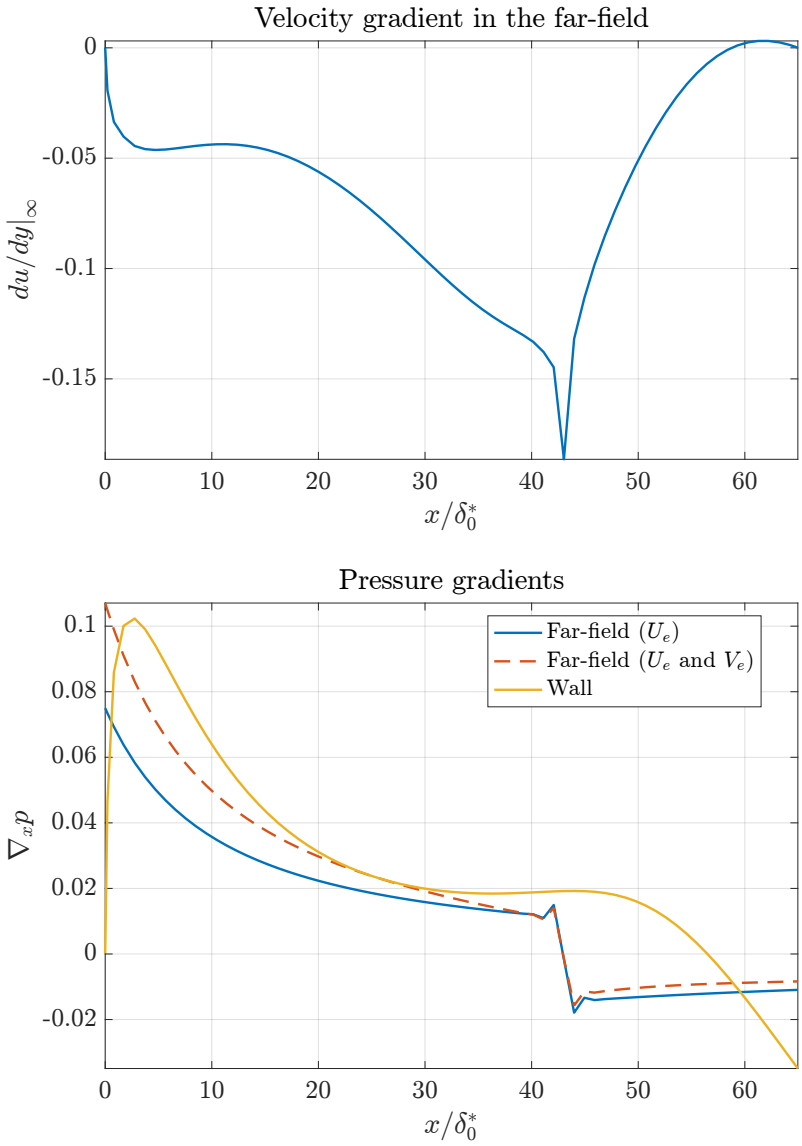

Figure 4: **Top:** steam-wise velocity gradient in the far-field $\frac{\partial u}{\partial y}$. It is very interesting to see how the flow comes back to $\frac{\partial u}{\partial y} = 0$ at the end of the domain, due to the parabolic outflow condition we have fixed. **Bottom:** Pressure gradients in the far-field and the wall. We have included the Bernoulli version of the external pressure gradient with and without taking into consideration the suction velocity $V_e(x)$.

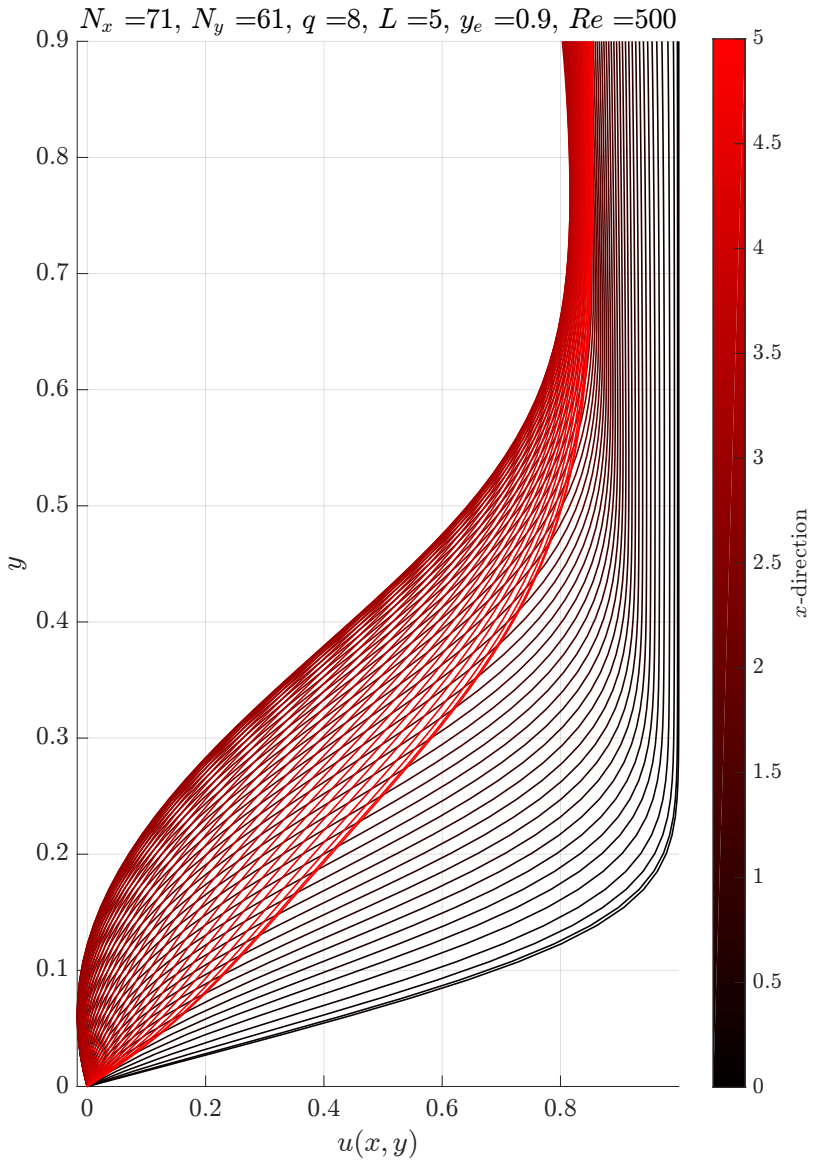

Figure 5: Stream-wise velocity component $u(y)$. The x-direction is shown with the gradient of colour used to plot the profiles.

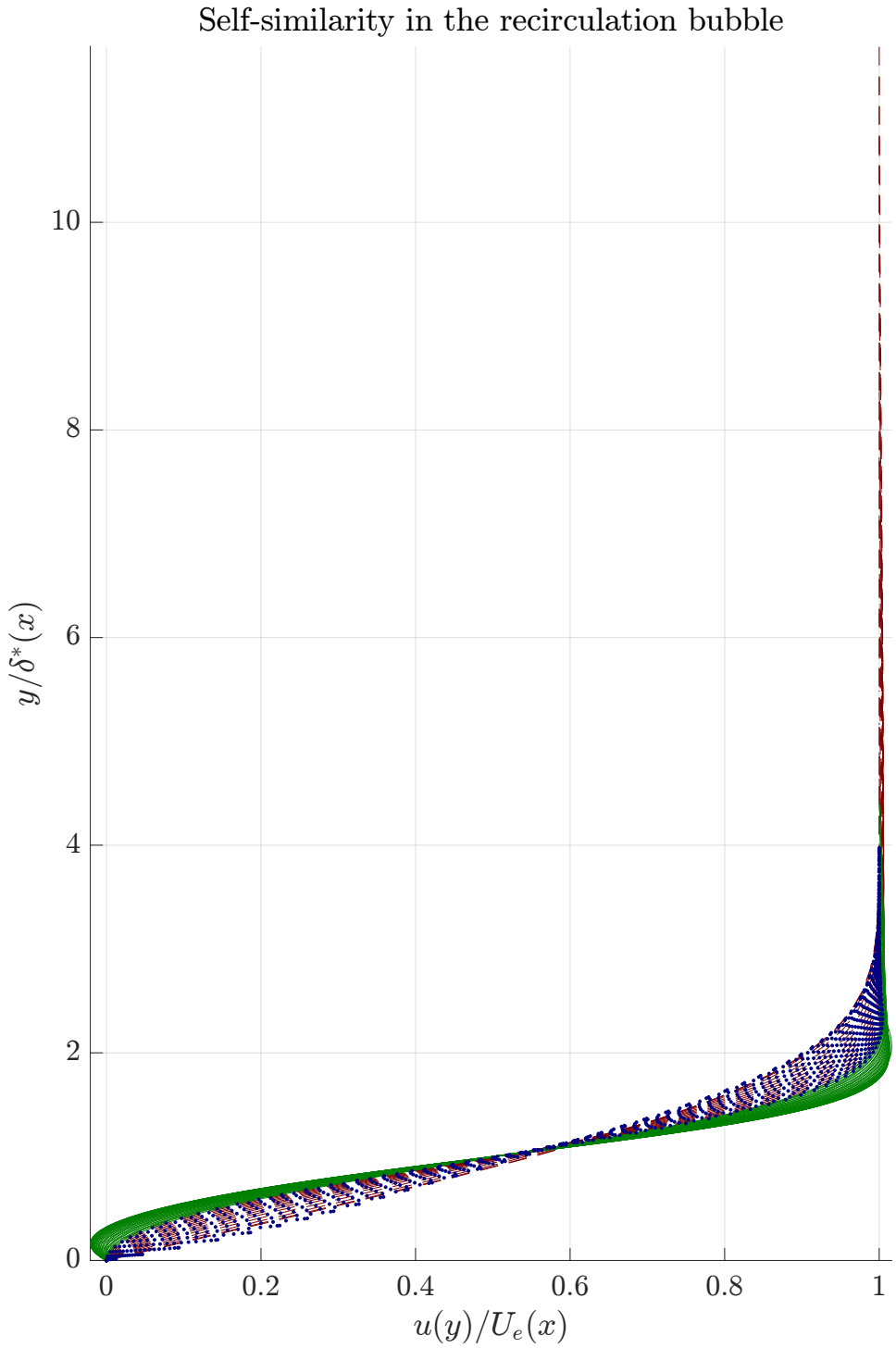

Figure 6: **Red, dashed:** Profiles before the bubble. **Green, continuous:** Profiles in the bubble. **Blue, dotted:** Profiles after the bubble.

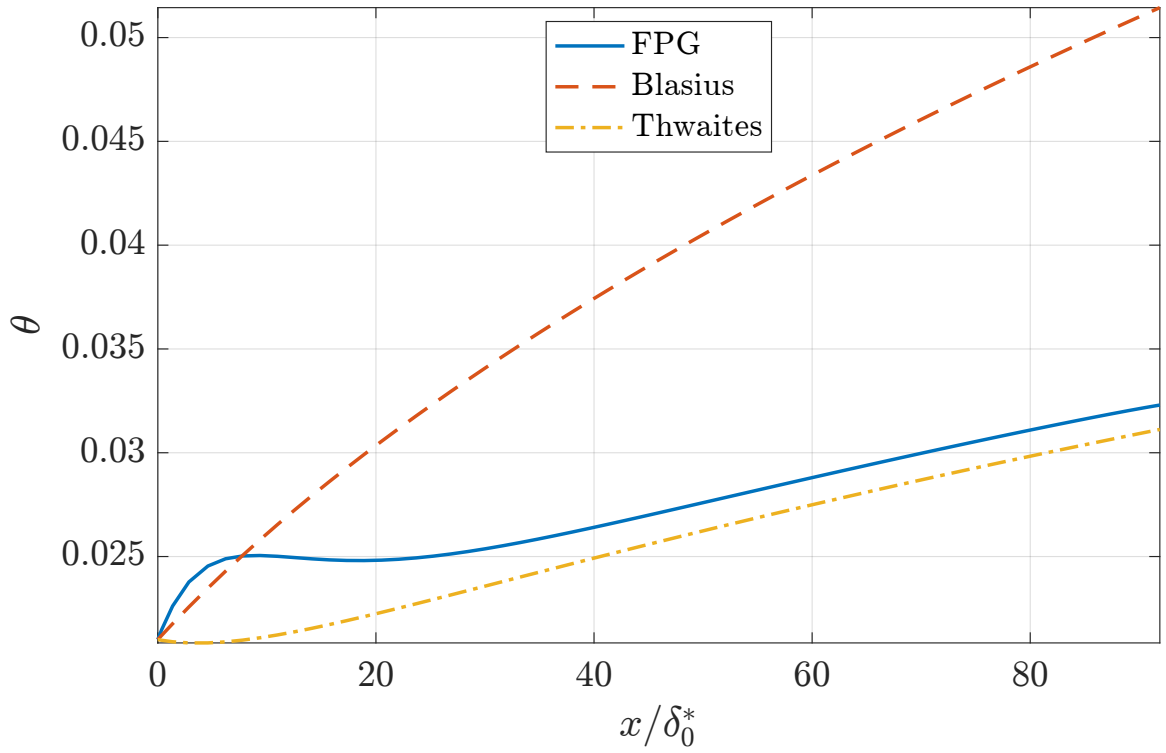

Figure 7: Momentum thickness in the FPG compared against the Blasius solution and Thwaites integral method.

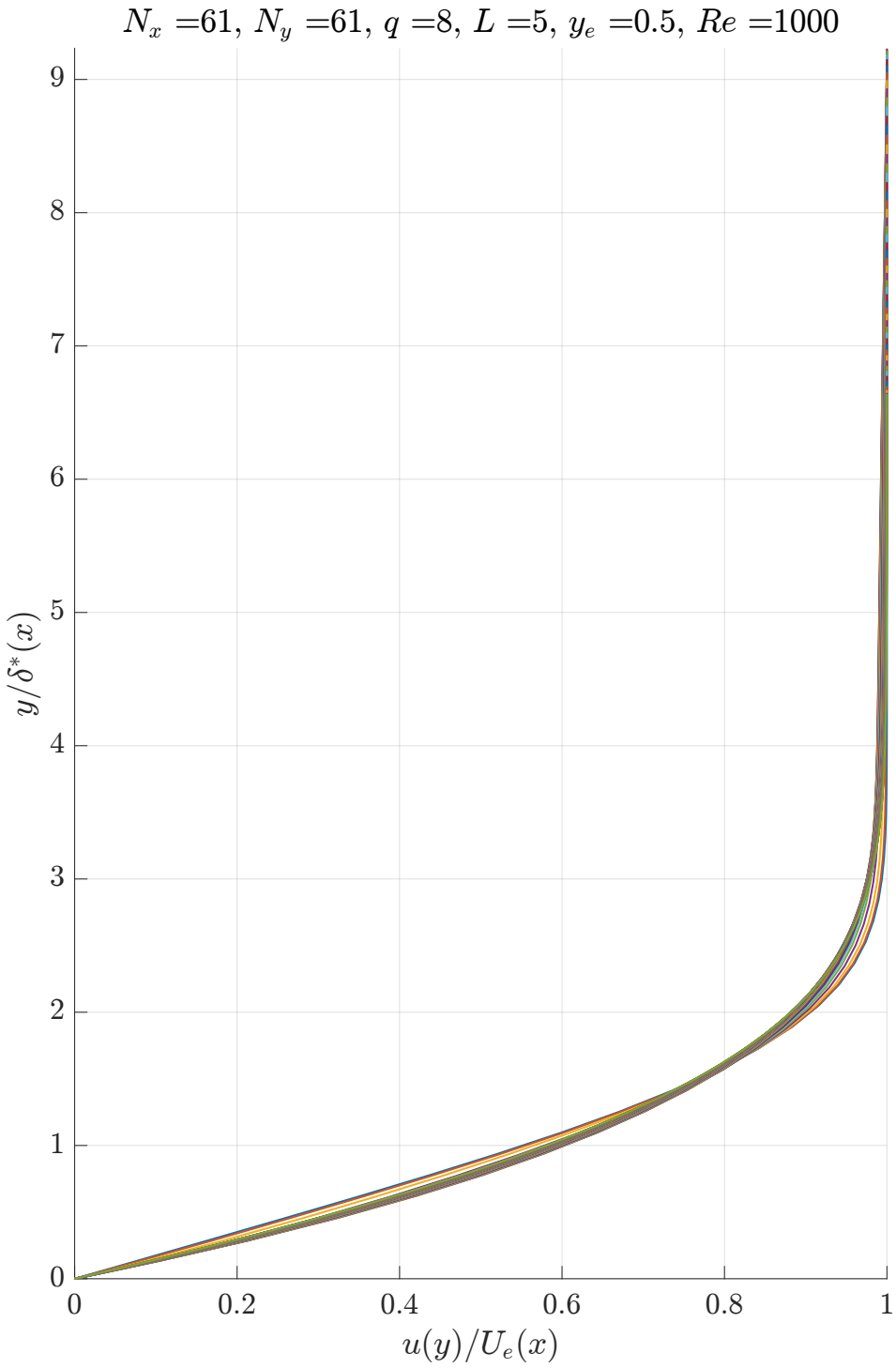

Figure 8: Self-similar velocity profiles in the FPG.

9 Conclusions

We have addressed the problem of solving the Navier-Stokes equations in derived variables, particularly for the streamfunction and the vorticity. The aim was to be able to study a flow over a flat plate in a boundary layer scenario, that is, with an irrotational far-field away from the plate where one of the velocity components is given. We avoided the use of the traditional primitive variables, velocity and pressure, because the boundary conditions in the far-field are ill-conditioned. In that formulation, one cannot assure continuity and irrotationality at the same time, obliging the user to have huge computational domains to avoid contamination of results.

After the completion of this work, we have *successfully* built a Direct Numerical Simulation code to solve the governing equations. Both equations have been be solved simultaneously to guarantee numerical and physical robustness. This is the case because no proper boundary condition can be derived through physical reasoning for the vorticity value or flux at the boundary with a rigid wall.

Different boundary conditions have been tested and simulated to test the robustness of the code. With those results, we have evaluated how the assumptions and simplifications required for the numerical implementation affect the tracing of the physics and the preservation of conservation laws or mathematical properties such as self-similarity. The remaining goal that was not completed due to a lack of time was the study of the boundary layer on the edge of separation.

Several conclusions can be drawn from this tests, both regarding the physics and the code. When it comes to the physics, the outflow is physically meaningful, which is by itself an achievement, since other type of outflows cause trouble or invalidate the results nearby. Despite the fact that some terms of the equations have been removed, it preserves self-similarity and mass. Additionally, we have proved that the code is robust when facing flow reversal, which gives us a peace of mind for the future works where we shall be studying the boundary layer on the edge of separation.

Regarding the code, the boundary condition implementation does not rely on the discretization method or the mesh, which makes the code *flexible* and *reusable*. Although it was not a requirement for the Internship, big parts of the code structure can be inherited for other projects to solve other problems.

On the whole, given the amount of time and the quality of the final code, the project was a success.

References

[1] K. E. Barrett. A variational principle for the stream function-vorticity formulation of the Navier-Stokes equations incorporating no-slip conditions. *Journal of Computational Physics*, 26(2):153–161, 1978.

[2] M. Ben-Artzi, J. P. Croisille, and D. Fishelov. A high order compact scheme for the pure-streamfunction formulation of the navier-stokes equations. *Journal of Scientific Computing*, 42(2):216–250, 2010.

[3] Erik Boström. Investigation of Outflow Boundary Conditions for Convection-Dominated Incompressible Fluid Flows in a Spectral Element Framework. page 120, 2015.

[4] W. R. Briley. A numerical study of laminar separation bubbles using the Navier–Stokes equations. *Journal of Fluid Mechanics*, 47(4):713–736, 2006.

[5] P. N. Brown and Y. Saad. Hybrid Krylov methods for nonlinear systems of equations. *SIAM Journal on Scientific and Statistical Computing*, 11(3):450–481, 1990.

[6] A Campion-Renson and M. J. Crochet. On the stream function-vorticity finite element solutions of Navier-Stokes equations. *International Journal for Numerical Methods in Engineering*, 12(January):1809–1818, 1978.

[7] F. White. *Viscous Flows*. 1974.

[8] B Fornberg. Steady viscous flow past a sphere at high Reynolds numbers. *Journal of Fluid Mechanics*, 190(1988), 1988.

[9] P. M. Gresho. Some current CFD issues relevant to the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 87(2-3):201–252, 1991.

[10] P M Gresho. Some Interesting Issues in Incompressible Fluid-Dynamics, Both in the Continuum and in Numerical-Simulation. *Advances in Applied Mechanics*, 28:45–140, 1992.

[11] M. Hermanns and J. A. Hernández. Stable high-order finite-difference methods based on non-uniform grid point distributions. *International Journal for Numerical Methods in Fluids*, (October 2007):601–629, 2008.

[12] D. A. Knoll and David E. Keyes. Jacobian-free Newton-Krylov methods: A survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.

[13] L. Quartapelle. *Numerical Solution of the Incompressible Navier-Stokes Equations*. 1993.

[14] S. J. Liao. Higher-order streamfunction-vorticity formulation of 2D steady-state Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 15(5):595–612, 1992.

[15] M Napolitano. Efficient solution of two-dimensional steady separated flows. *Comput. Fluids*, 20(3):213–222, 1991.

[16] A. E. Perry and M. S. Chong. A series-expansion study of the Navier-Stokes equations with applications to three-dimensional separation patterns. *Journal of Fluid Mechanics*, 173:207–223, 1986.

[17] H.A Rodriguez-Prada, F. F Pironti, and A. E Sáez. FUNDAMENTAL SOLUTIONS OF THE STREAMFUNCTION-VORTICITY FORMULATION OF THE NAVIER-STOKES EQUATIONS. *International Journal for Numerical Methods in Fluids*, 10(February 1989):1–12, 1990.

[18] R Schreiber and H.B Keller. Driven cavity flows by efficient numerical techniques. *Journal of Computational Physics*, 49(2):310–333, 1983.

[19] M. P. Simens, J. Jiménez, S. Hoyas, and Y. Mizuno. A high-resolution code for turbulent boundary layers. *Journal of Computational Physics*, 228(11):4218–4231, 2009.

[20] T. E. Tezduyar. Finite element formulation for the vorticity-stream function form of the incompressible euler equations on multiply-connected domains. *Computer Methods in Applied Mechanics and Engineering*, 73(3):331–339, 1989.

[21] E. Weinan and J G Liu. Vorticity boundary condition and related issues for finite difference schemes. *Journal of Computational Physics*, 124(2):368–382, 1996.