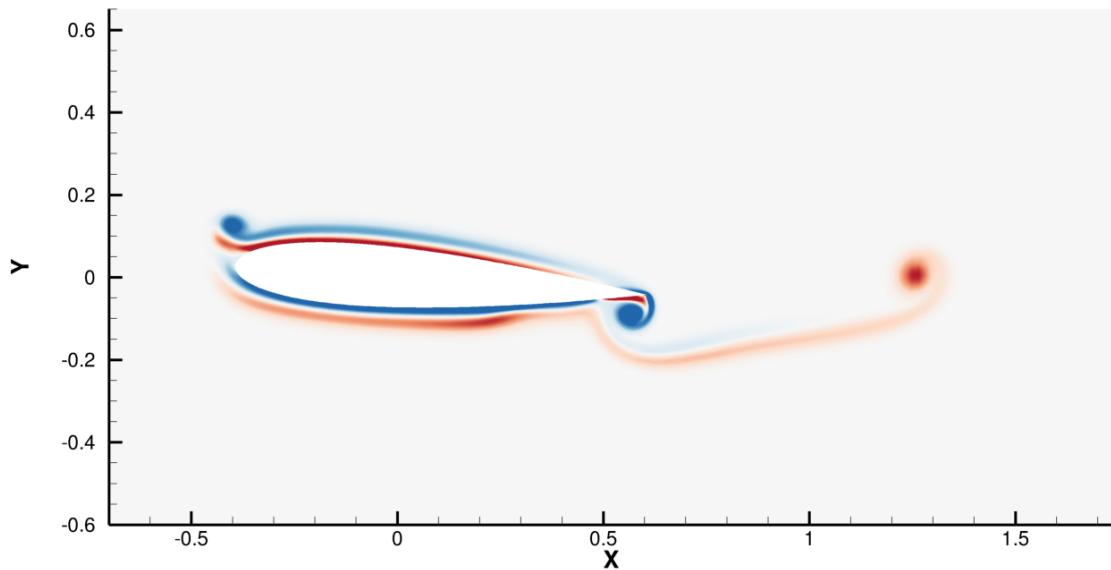


Examination of the Details of 2D Vorticity Generation Around the Airfoil During Starting and Stopping Phases

Alwin Wang
Supervised By: Hugh Blackburn

Department of Mechanical and Aerospace Engineering
2018 Final Year Project



Contents

I Project Outline	5
1 Introduction	5
1.1 Objectives	5
1.2 Previous Work	6
2 Methodology	11
2.1 Research Approach	11
2.2 Computational Method	11
II Results and Discussion	19
3 Vorticity and Vorticity Generation	19
3.1 Airfoil Boundary Properties	19
3.2 Establishment of the Kutta Condition	30
3.3 Circulation of Airfoil and Shed Vortices	32
4 Comparison to Unsteady Potential Theory	34
4.1 Lift, Drag and Added-Mass Forces	34
4.2 Unsteady Vortex Sheet Methods	37
III Closing Remarks	38
5 Conclusion	38
IV Appendix	41
A Proofs and Derivations	41
A.1 Vorticity Generation	41
A.2 Unsteady Thin-Airfoil Theory	42
A.3 Unsteady Panel Method	43
B Post-Processing Scripts	48
B.1 Semtex Files	48
B.2 Bash Scripts	51
B.3 R Scripts	54

List of Figures

1.1	Observation of starting and stopping vortex pair	6
1.2	Creation of the starting vortex demonstrating Kelvin's theorem	6
1.3	Coordinate system on the surface of a wall	7
1.4	Establishment of the Kutta condition	8
2.1	Boundary conditions imposed on the simulation	12
2.2	Coordinate system used for the surface of the airfoil	13
2.3	Comparison of methods to determine partial derivatives	13
2.4	Spectral element mesh of NACA0012 with unit chord at $\alpha = 4^\circ$	15
2.5	Increased mesh density of the original NACA0012 mesh	15
2.6	p -Convergence study of the variables of interest for a planar Kovasznay flow	16
2.7	Numerical error of $-\nu \vec{n} \cdot \nabla \vec{\omega} + \vec{n} \times \nabla \vec{P} + \vec{t} \cdot \vec{a}(t) = 0$ for $3 \leq p \leq 19$	17
2.8	p -Convergence study of the variables of interest for $3 \leq p \leq 18$	18
2.9	Convergence of the integral of the BVF around the airfoil for $3 \leq p \leq 19$	18
2.10	Convergence of the integral of the vorticity and vortex impulse for $3 \leq p \leq 19$	18
3.1	Integral of BVF versus time $0 \leq t \leq 2.5$ for $Re = 1,000$	19
3.2	Integral of BVF versus time $0 \leq t \leq 2.5$ for $Re = 10,000$	19
3.3	Acceleration stage for $Re = 1,000$. Top to bottom: $t = 0.05, 0.10, 0.15, 0.20$	20
3.4	Acceleration stage for $Re = 10,000$. Top to bottom: $t = 0.05, 0.10, 0.15, 0.20$	21
3.5	Steady-state stage for $Re = 1,000$. Top to bottom: $t = 0.25, 0.45, 0.65, 0.85$	22
3.6	Steady-state stage for $Re = 10,000$. Top to bottom: $t = 0.25, 0.45, 0.65, 0.85$	23
3.7	Deceleration stage for $Re = 1,000$. Top to bottom: $t = 0.90, 0.95, 1.00, 1.05$	24
3.8	Deceleration stage for $Re = 10,000$. Top to bottom: $t = 0.90, 0.95, 1.00, 1.05$	25
3.9	Stopped stage for $Re = 1,000$. Top to bottom: $t = 1.25, 1.45, 1.65, 1.85$	26
3.10	Stopped stage for $Re = 10,000$. Top to bottom: $t = 1.25, 1.45, 1.65, 1.85$	27
3.11	Vorticity on the airfoil surface at the leading and trailing edges	28
3.12	Vorticity on the airfoil surface at the leading and trailing edges	29
3.13	Vorticity distribution around the trailing edge during the starting flow	31
3.14	Vorticity field at $t = 2.5$ s for $Re = 1,000$ and $Re = 10,000$	31
3.15	Definition of circulation regions	33
3.16	Circulation versus time $0 \leq t \leq 2.5$ for $Re = 1,000$	33
3.17	Circulation versus time $0 \leq t \leq 2.5$ for $Re = 10,000$	33
4.1	Vortex impulse vs time, $0 \leq t \leq 2.5$	34
4.2	Lift versus time, $0 \leq t \leq 2.5$	35
4.3	Drag versus time, $0 \leq t \leq 2.5$	35
4.4	Added mass lift force coefficient, $0 \leq t \leq 2.5$	35
4.5	Added mass drag force coefficient, $0 \leq t \leq 2.5$	36
4.6	Unsteady lift estimation using equation (1.17)	37
A.1	Coordinate system on the surface of a wall	41

Listings

1	Exerts of modifications to addfield.cpp	48
2	Exerts of modifications to wallmesh.cpp	49
3	Header of session file for Re = 10,000	49
4	Bash script to perform DNS for convergence study	51
5	Bash script to integrate element-by-element across the domain	52
6	Bash script to generate Tecplot files for each time step	53
7	R script to batch process session and dump files	54
8	R script calculations of the session, mesh and batch files	56

Nomenclature

$V(t)$	Velocity of the body (airfoil)	\vec{P}_b	Body-volume impulse
$a(t)$	Acceleration of the body (airfoil)	\vec{P}_i	Image-vorticity impulse
σ	Boundary vorticity flux (BVF)	\vec{P}_v	Shed-vorticity impulse
$\vec{\sigma}$	BVF in 3D	L'	Lift per unit span (N per m)
$\vec{\sigma}_a$	BVF from acceleration	N_P	Polynomial order for SEM
$\vec{\sigma}_f$	BVF from body forces	p	Pressure (Pa)
$\vec{\sigma}_p$	BVF from pressure	P	Pressure normalised by density, $\frac{p}{\rho}$
$\vec{\sigma}_{vis}$	BVF from viscosity	Re	Reynolds number
Γ	Circulation (m^2/s)	S	Surface bounded by C
C	Counterclockwise closed curve	dS	Infinitesimally small surface
$d\vec{s}$	Tangent vector along boundary C	\vec{n}	Normal vector to a surface
$\vec{\nabla}$	Del operator	\vec{t}	Tangent vector to a surface
ρ	Local density (kg/m^3)	t	Time (s)
ρ_∞	Freestream density (kg/m^3)	\vec{u}	Velocity vector (u, v)
\vec{F}_{tot}	Total force (N)	V_∞	Freestream velocity (m/s)
\vec{F}_c	Circulatory force	ν	Kinematic viscosity (m^2/s)
\vec{F}_{am}	Added mass force (N)	$\vec{\omega}$	Vorticity vector
\vec{P}	Vortical impulse	ω	Vorticity into/out of a 2D plane
\vec{P}_ϕ	Potential impulse	z	Distance normal to the surface

Part I

Project Outline

1 Introduction

1.1 Objectives

Background and Motivation

The theory behind the generation of lift has been of interest long before heavier-than-air flight. With research into airfoils came many new theories on lift production mechanisms. Early experiments demonstrated the formation of starting and stopping vortices which are understood to be due to conservation of circulation. Multiple formations for calculating the lift exist in current literature and this project attempts to take a more novel approach by considering the boundary vorticity flux to explain phenomena observed such as the Kutta condition and added mass lift and drag force.

Need for Research

The primary variables for describing and analysing a flow field are the velocity and pressure components u , v and p . Vorticity is derived variable and so requires some justification for its use. As such, this project explores phenomena that are more difficult to understand using primary variables. Detailed investigation of flows around airfoils have been conducted, however many attempt to reduce the complexity of the problem by using a different solution formation of the problem such vorticity-stream function ($\omega - \psi$) to using conformal mapping around wedges. This analysis will be using a direct numerical solver Semtex which solves the problem directly.

Aims and Objectives

The first aim is to determine an accurate method of calculating the vorticity generation around an airfoil. As the vorticity generation is a boundary condition on the airfoil surface, the next aim is to use the boundary vorticity flux to explain observations in the flow field such as the establishment of the Kutta condition and formation of starting vortices.

Vorticity will also be used to predict the lift and drag as well as investigating the contribution of added mass effects during the starting and stopping phases. Time permitting, a comparison to unsteady potential theory will be performed to determine the effect of Reynolds number. Thus, two cases, $Re = 1,000$ and $Re = 10,000$ (based on steady state velocity) will be analysed using DNS and the potential theory case represents an infinite Reynolds number. A single airfoil, NACA0012, of unit chord is used at an angle of attack 4° accelerated to unit velocity in a fluid of unity density.

1.2 Previous Work

Observation of Starting and Stopping Vortices

Whenever an airfoil is accelerated or decelerated in a fluid, starting and stopping vortices are formed. Around 1930, Prandtl, Tietjens and Müller recorded the motion of fine particles around an airfoil in the starting (accelerating) and stopping (decelerating) phases of motion to observe transient, unsteady flows [14]. The original recordings have been analysed using modern particle image velocimetry (PIV) by Wilbert and Kompenhans [20] and the phenomena of starting and stopping vortices still remains of interest.

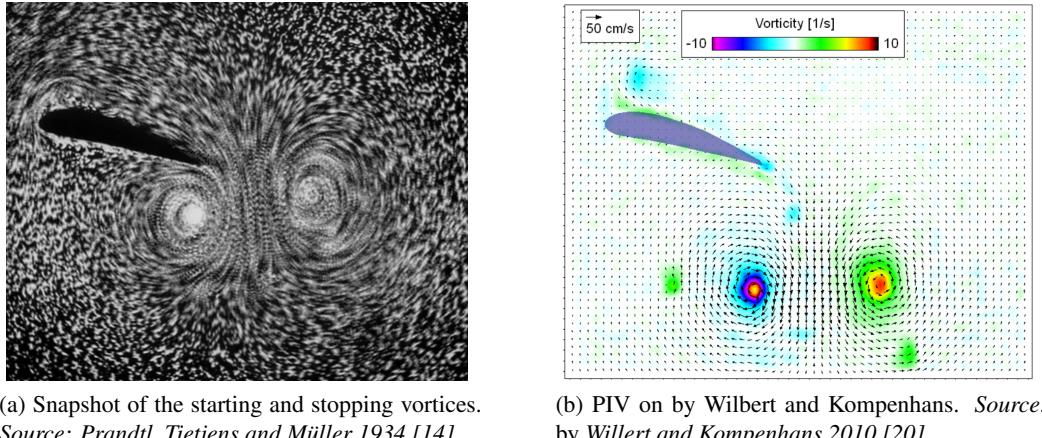


Figure 1.1: Observation of starting and stopping vortex pair

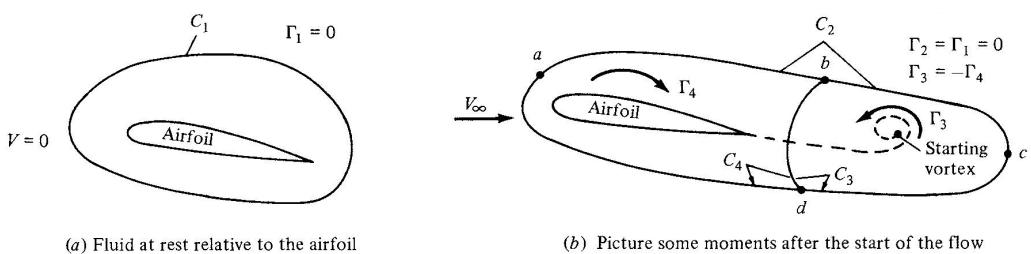


Figure 1.2: Creation of the starting vortex demonstrating Kelvin's theorem for circulation. Source: Anderson 2010, pg 336 [2]

The explanation for the formation of the starting vortex is well covered in existing literature by applying Kelvin's theorem, equation (1.1), to a contour around the airfoil, shown in Figure 1.2. Anderson [2] explains that initially when the fluid is at rest, the circulation around the curve C_1 is zero and when the fluid begins motion the flow will initially curl around the trailing edge creating a thin region of high vorticity (large velocity gradients). This vorticity is fixed to the same fluid elements and so travels downstream from the trailing edge due to the fluid velocity, V_∞ . As this thin sheet of vorticity is unstable, it tends to roll up to form a starting vortex as shown in Figure 1.2b.

From Kelvin's theorem, the circulation Γ_2 around curve C_2 is equal to the circulation Γ_1 around curve C_1 implying $\Gamma_1 = \Gamma_2 = 0$. Considering curve C_3 , bcd , and curve C_4 , $abda$, it can be seen that $\Gamma_3 + \Gamma_4 = \Gamma_2$ giving $\Gamma_3 = -\Gamma_4$. This indicates that the circulation bound to the airfoil, Γ_4 , is equal and opposite to the circulation around the starting vortex, Γ_3 .

$$\frac{D\Gamma}{Dt} = \oint_C \frac{D\vec{u}}{Dt} \cdot d\vec{s} + \oint_C \vec{u} \cdot \frac{Dd\vec{s}}{Dt} = 0 \quad (1.1)$$

Vincent and Blackburn [18] showed the formation of these vortices by performing a direct numerical simulation (DNS) of transient flow over a NACA0012 airfoil at $Re = 10,000$ and $\alpha = 4^\circ$ while Agromayo, Rúa and Kristoffersen [1] investigated a NACA4612 at $Re = 1,000$ and $\alpha = 16^\circ$ using OpenFOAM. Both studies determined coefficients of lift and drag during the starting and stopping phases and verified Kelvin's and Stoke's theorems, shown in equation (1.2b), for vorticity around various contours. This project expands on these studies by considering the vorticity generation mechanisms and exploring the physical phenomena behind vortices generated during the starting and stopping phases.

$$L' = \rho_\infty V_\infty \Gamma \quad (1.2a)$$

$$\Gamma \equiv - \oint_C \vec{u} \cdot d\vec{s} = - \iint_S (\vec{\nabla} \times \vec{u}) \cdot dS = - \iint_S \vec{\omega} \cdot dS \quad (1.2b)$$

Vorticity Generation at Solid Boundaries

It is recognised that the sources of vorticity must occur at the boundary of the fluid regions and for the starting and stopping phases of motion. Morton [15] outlines two production mechanisms for vorticity: tangential pressure gradients from the fluid side, and the acceleration of the surface from the wall side, shown in equation (1.3). These contributions were investigated by Blackburn and Henderson [3] for vortex shedding of oscillating cylinders and it was noted that the pressure-gradient generation mechanism could override the surface-acceleration generation mechanism and vice versa.

$$\underbrace{-v \left(\frac{\partial \vec{\omega}}{\partial z} \right)_{z=0}}_{\text{Vorticity Flux}} = -\underbrace{\frac{1}{\rho} [\vec{n} \times \nabla] \vec{p}}_{\text{"Pressure Gradient"}} \Big|_{z=0} - \underbrace{\vec{n} \times \frac{dV(t)}{dt}}_{\text{"Acceleration"}}$$

where $V(t)$ is the velocity of the wall moving in its own plane and \vec{n} is the normal vector to the surface.

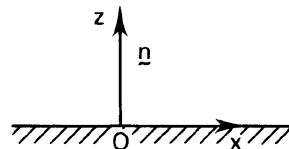


Figure 1.3: Coordinate system on the surface of a wall. *Source: Morton 1984, pg 283 [15]*

This expresses the idea that vorticity must diffuse away from a solid boundary at the same rate that it is produced by local conditions. During the starting and stopping phases of motion, there is no angular acceleration present. Thus, the integral around the body perimeter of the flux of vorticity is zero at every instant provided the pressure field is continuous [3, 24]. This is shown in equation (1.4).

$$\oint -v \vec{n} \cdot \nabla \vec{\omega} ds = \oint -v \frac{\partial \omega}{\partial n} ds = 0 \quad (1.4)$$

Zhu et al. [24] investigated the causal mechanisms for airfoil circulation using vorticity creation theory based on Lighthill's relations [11], shown in equation (1.5), instead of boundary-layer theory. Additionally, the realisation of the Kutta condition and creation of starting vortex were determined through a complex chain of processes which were also explained by considering boundary vorticity flux (BVF), σ . It should be noted there is a sign difference in term of interest on the left hand side of equation (1.3) and the definition of σ in equation (1.5).

$$v \frac{\partial \omega}{\partial n} \equiv \sigma \quad (1.5)$$

Zhou et al. [23] describes additional BVF terms to arrive at a general expression for $\vec{\sigma}$ shown in Equation (1.6a).

$$\vec{\sigma} = \vec{n} \times \left(\vec{a}_B - \vec{f} + \frac{1}{\rho} \nabla p \right) + \nu (\vec{n} \times \nabla) \times \vec{\omega} \equiv \vec{\sigma}_a + \vec{\sigma}_f + \vec{\sigma}_p + \vec{\sigma}_{vis} \quad (1.6a)$$

$$\vec{\sigma}_a = \vec{n} \times \vec{a}_B \quad \vec{\sigma}_f = -\vec{n} \times \vec{f} \quad (1.6b)$$

$$\vec{\sigma}_p = \frac{1}{\rho} \vec{n} \times \nabla p \quad \vec{\sigma}_{vis} = \nu (\vec{n} \times \nabla) \times \vec{\omega} \quad (1.6c)$$

Note that this expression collapses into equation (1.3) since \vec{f} is assumed to be 0 and $\vec{\sigma}_{vis} = 0$ for two dimensional problems. Thus,

$$\vec{\sigma} = \vec{\sigma}_p + \vec{\sigma}_a = \frac{1}{\rho} (\vec{n} \times \nabla) \vec{p} + \vec{n} \times \vec{a}_B = \nu \frac{\partial \omega}{\partial n} \quad (1.7)$$

For an incompressible flow problem, the forces on the airfoil can be determined using equation (1.8). In steady incompressible flow, the forces acting on a 2D airfoil can be determined solely by the boundary vorticity flux σ_p assuming the contribution of skin friction is negligible for $Re \gg 1$ [23] (equation (1.9)).

$$\vec{F} = - \int_{\partial B} \rho \vec{x} \times \left(\frac{1}{2} \vec{\sigma}_p + \vec{\sigma}_{vis} \right) dS \quad (1.8)$$

$$L = \rho \int_C x \sigma_p ds, \quad D = -\rho \oint_C y \sigma_p ds \quad (1.9)$$

Establishment of the Kutta Condition

When the airfoil first gains a non-zero relative velocity, the fluid pattern is very similar to the inviscid potential flow solution. Anderson [2] explains that particles very close to the lower surface turn rapidly around the trailing edge. The streamline curvature is finite and moves upstream over the upper surface leading (opposing the freestream flow) to a separation point on the upper surface of the trailing edge.

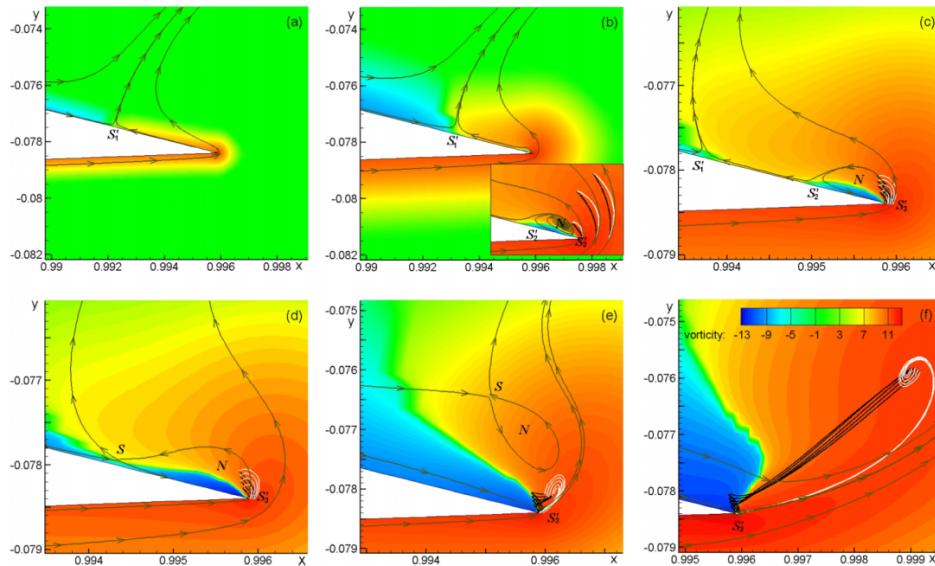


Figure 1.4: Establishment of the Kutta condition highlighting key stages. Pathlines (black), streaklines (white) and streamlines (gray) shown. Source: Zhu et al. 2015, pg 10 [24]

Using a vorticity-stream function ($\omega - \psi$) solver, Zhu et al. [24] investigated the causal mechanisms for the Kutta condition. They observed a thin sublayer on the trailing upper surface with vorticity $\omega < 0$ appears underneath the upper-surface boundary layer which has vorticity $\omega > 0$ (figure 1.4b). This forms an attached bubble with two new semi-saddles (separation points) and a node. Fluid particles from the lower surface continue to turn around the trailing edge, but have a tendency to lift up from the surface indicating fluid-particle separation (generic unsteady separation).

A critical event was identified, (figure 1.4d) where the two saddle point meet to form a single separation point which rises up into a very narrow irrotational zone. Below the saddle point, the two shear layers of $\omega < 0$ merge into a single layer and the lower boundary layer with strong $\omega > 0$ is now being shed from the trailing edge into the wake as a free shear layer. Particles that were moving upstream (black) are forced away by the large BVF causing the “kink” in the pathlines seen in figures 1.4e and 1.4f.

Bubbles of opposite vorticity were also observed by Xu [22] in a numerical study of impulsively started viscous flow past wedges. Kinks in the pathlines were also observed due to the regions of vorticity with alternating sign.

Added-Mass Effect for Unsteady Flows

In their review, Limacher, Morton and Wood outline the concept of that added-mass is due to the momentum a body imparts to the surrounding fluid. Assuming an irrotational flow for a body of mass M experiencing a linear acceleration $a(t)$ without rotation, the additional force is proportional to the instantaneous acceleration of the body [12]. Letting the constant of proportionality be m' , the total force on the body can be found as

$$\vec{F}_{\text{tot}} = -(M+m)a(t), \quad \vec{F}_{\text{am}} = -m \cdot \vec{a}(t) \quad (1.10)$$

where m' can also be interpreted as “added-mass” or “apparent mass” or “virtual mass” that must be accelerated with the body. While the apparent mass can be intuitively understood as the volume of fluid ahead of the body that needs to be accelerated around the body, there is no actual identifiable finite volume of fluid. All the fluid around the body (to infinity) is accelerated to varying degrees causing the force F_{am} .

For incompressible flows using potential flow analysis, $\vec{u} = \nabla\phi$ in the frame of reference where velocity decays to zero at infinity, Saffman [16] outlines that the conservation of energy can be used to determine m' . The added-mass force associated with the scalar potential can also be determined for a body with impermeable boundary condition on its surface.

$$m' = \rho \int_V \frac{u^2}{V(t)^2} dV, \quad \vec{F}_{\text{am}} = -\rho \frac{d}{dt} \oint_{S_b} \Phi \vec{n} dS \quad (1.11)$$

Limacher, Morton and Wood [12] performed a generalised derivation for a body of arbitrary shape to determine the added-mass force for separated viscous flows. The total impulse, \vec{P} , where $\vec{F} = -\rho \frac{d\vec{P}}{dt}$ is given in Equations (1.12a) and (1.12b). The arbitrary motion of the airfoil is characterised by instantaneous linear velocity of the body centroid, \vec{u}_c , and the angular velocity, $\vec{\Omega}$.

$$\vec{P} = \frac{1}{N-1} \int_V (\vec{x} \times \vec{\omega}) dV + \frac{1}{N-1} \int_{V_b} (\vec{x} \times \vec{\omega}) dV - \oint_{S_b} \vec{n} \Phi' dS - \vec{u}_c V_b \quad (1.12a)$$

$$\vec{P} \equiv \vec{P}_v + \vec{P}_i + \vec{P}_\Phi + \vec{P}_b \quad (1.12b)$$

where $N = \{2, 3\}$ is the dimension of space, \vec{x} is defined such that $\vec{x} = 0$ at body centroid for $\int_{V_b} \vec{\Omega} \times \vec{x} dV = 0$, Φ' is the velocity potential in the body-fixed frame of reference defined as $\nabla\Phi' = \nabla\Phi - \vec{u}_c$ and S_b is a piecewise smooth surface with outward unit normal \vec{n} of the body and encloses the volume V_b .

\vec{P}_v is the shed-vorticity impulse of all the vorticity in the fluid domain V up to the surface S_b . \vec{P}_i is the image-vorticity impulse for a non-unique vorticity distribution within the body V_b that maintains the

impermeable boundary condition on S_b . \vec{P}_Φ is the potential impulse introduced from using the body-fixed reference frame, $\vec{u}' = \vec{u} - \vec{u}_c - \vec{\Omega} \times \vec{x} \equiv \nabla \Phi' + \nabla \times \psi'$. \vec{P}_b is the body-volume impulse. The sum of first two terms, $\vec{P}_v + \vec{P}_i$ leads to the total vortical impulse and the sum of the other two terms, $\vec{P}_\Phi + \vec{P}_b$ leads to the added mass force.

$$\vec{F}_c = -\rho \frac{d}{dt} (\vec{P}_v + \vec{P}_i) \quad \text{and} \quad \vec{F}_{am} = -\rho \frac{d}{dt} (\vec{P}_\Phi + \vec{P}_b) \quad (1.13)$$

When the no slip condition holds, Limacher, Morton and Wood [12] state that $\vec{P}_i + \vec{P}_\Phi = 0$ which leads to their second, no slip formulation.

$$\vec{P} = \vec{P}_v + \vec{P}_b = \frac{1}{N-1} \int_V (\vec{x} \times \vec{\omega}) dV - \vec{u}_c V_b \quad (1.14)$$

Thin-Airfoil Theory Applied to Unsteady Flows

Another point of interest identified by Vincent and Blackburn [18] and Agromayo, Rúa and Kristoffersen [1] was the large value of lift during the starting (accelerating) phase. Kármán and Sears [10] attributed this to unsteady flows over airfoils which was later extended by Liu et al. [13] and Limacher, Morton and Wood [12].

After the starting phase, when the airfoil had attained a uniform velocity, it was also observed the lift force would asymptote to a steady-state value. An explanation for this was also provided by Kármán and Sears [10] due to a “lift deficiency” term from the effect of wake vorticity sheet generated during acceleration. This behaviour was also detailed in Saffman explaining latency in lift production known as the “Wagner effect” [16, 19].

According to classic thin airfoil theory provided in Anderson [2] the vortex sheet strength of an airfoil, $\gamma(\xi)$, can be determined as

$$\frac{1}{2\pi} \int_{-1}^1 \frac{\gamma(\xi)}{x-\xi} = V_\infty \left(\alpha - \frac{dz}{dx} \right) \quad (1.15)$$

where the terminals have been adjusted to match the definition provided in Kármán and Sears [10]. In this analysis, $\frac{dz}{dx} = 0$ as the NACA0012 is a symmetric airfoil and the transformations $\xi = -\cos(\theta)$ and $x = -\cos(\theta_0)$ are used to obtain the standard result

$$\gamma(\theta) = 2\alpha V_\infty \frac{1 + \cos(\theta)}{\sin(\theta)} \quad (1.16)$$

where V_∞ is the prescribed boundary condition, $V_\infty = -V(t)$.

This can be applied to Kármán and Sears [10] derivation for the lift of a thin airfoil in non-uniform motion:

$$L = \underbrace{\rho V_\infty \Gamma_0}_{\text{Quasi-Steady}} - \underbrace{\rho \frac{d}{dt} \int_{-1}^1 \gamma_0(x) x dx}_{\text{Apparent Mass}} - \underbrace{\rho V_\infty \int_1^\infty \gamma(\varepsilon) \frac{d\varepsilon}{\sqrt{\varepsilon^2 - 1}}}_{\text{Wake Effect}} \quad (1.17)$$

where $\gamma_0(x)$ and Γ_0 are the vortex sheet strength and circulation respectively, calculated from thin airfoil theory, equation (1.16). $\gamma(\varepsilon)$ is the vorticity of the wake assumed to be on the airfoil plane a distance ε from the mid chord ($x = 0$).

Kármán and Sears [10] present a solution for the wake effect term, however from PIV by Willert and Kompenhans [20] shown in figure 1.1b it is clear the assumption of the wake remaining in the same plane as the airfoil does not hold as wake vortex sheet rolls up to form the starting vortex. As such, the solution provided in equation (1.17) may not be appropriate for the analysis of the starting phase of the airfoil.

2 Methodology

2.1 Research Approach

While this project could be analysed experimentally, there would be significant barriers. Primarily, particle image velocimetry (PIV) would not be practical for such small time and length scales for vorticity generation. For instance, during the establishment of the Kutta condition would require impractically precise recording equipment and algorithms. Inducing the required acceleration and deceleration of the fluid and introducing particles to track in PIV without introducing turbulence would be another greatly difficult task.

A theoretical examination of the flow could be attempted. However as this is a viscous transient problem around a lift-producing airfoil there are few (if any) complete theoretical results to guide the project. There is no guaranteed closed-form theoretical solution as the vorticity generation is derived from the Navier Stokes equations and using alternative approaches such as potential flows would defeat the purpose of the “detailed” examination.

As such, a direct numerical simulation (DNS) was performed. The code used was a spectral element-Fourier DNS code, semtex [4]. The governing equations solved were the non-dimensionalised Navier-Stokes equations in the moving reference frame fixed to the airfoil stated in equations (2.1a) and (2.1b). In order to determine the vorticity generation and other quantities of interest such as the vortex impulse, significant post-processing of results was required.

$$\nabla \cdot \vec{u} = 0 \quad (2.1a)$$

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\nabla \vec{P} + \frac{1}{Re} \nabla^2 \vec{u} - a \quad (2.1b)$$

where $\vec{P} = \frac{\vec{p}}{\rho}$ and a is the acceleration of the reference frame. The boundary conditions at prescribed velocity boundaries are set as $u = -V(t)$ where $V(t)$ is the velocity of the reference frame such that $a = V'(t)$.

2.2 Computational Method

Direct Numerical Simulation Using Semtex

The acceleration profile for the airfoil was chosen to be the same as Vincent and Blackburn [18] which represents non-impulsively started flow to unity free-stream velocity (figure 2.1). In Saffman’s [16] the explanation of the “Wagner effect” [19], it is noted that the initial lift is one-half of the final steady-state lift after a time $O(c/v)$. As such, a period of 0.8 s of uniform flow between the starting and stopping phases allows the convergence of lift to be investigated.

$$a(t) = - \begin{cases} 0, & 0 \leq t \leq 0.05 \\ \frac{5\pi}{2} \sin(5\pi(t-0.05)), & 0.05 < t \leq 0.25 \\ 0, & 0.25 < t \leq 0.85 \\ -\frac{5\pi}{2} \sin(5\pi(t-0.05)), & 0.85 < t \leq 1.05 \\ 0, & t > 1.05 \end{cases} \quad (2.2)$$

$$u = \begin{cases} 0, & 0 \leq t \leq 0.05 \\ \frac{1}{2} - \frac{1}{2} \cos(5\pi(t-0.05)), & 0.05 < t \leq 0.25 \\ 1, & 0.25 < t \leq 0.85 \\ \frac{1}{2} + \frac{1}{2} \cos(5\pi(t-0.05)), & 0.85 < t \leq 1.05 \\ 0, & t > 1.05 \end{cases} \quad (2.3)$$

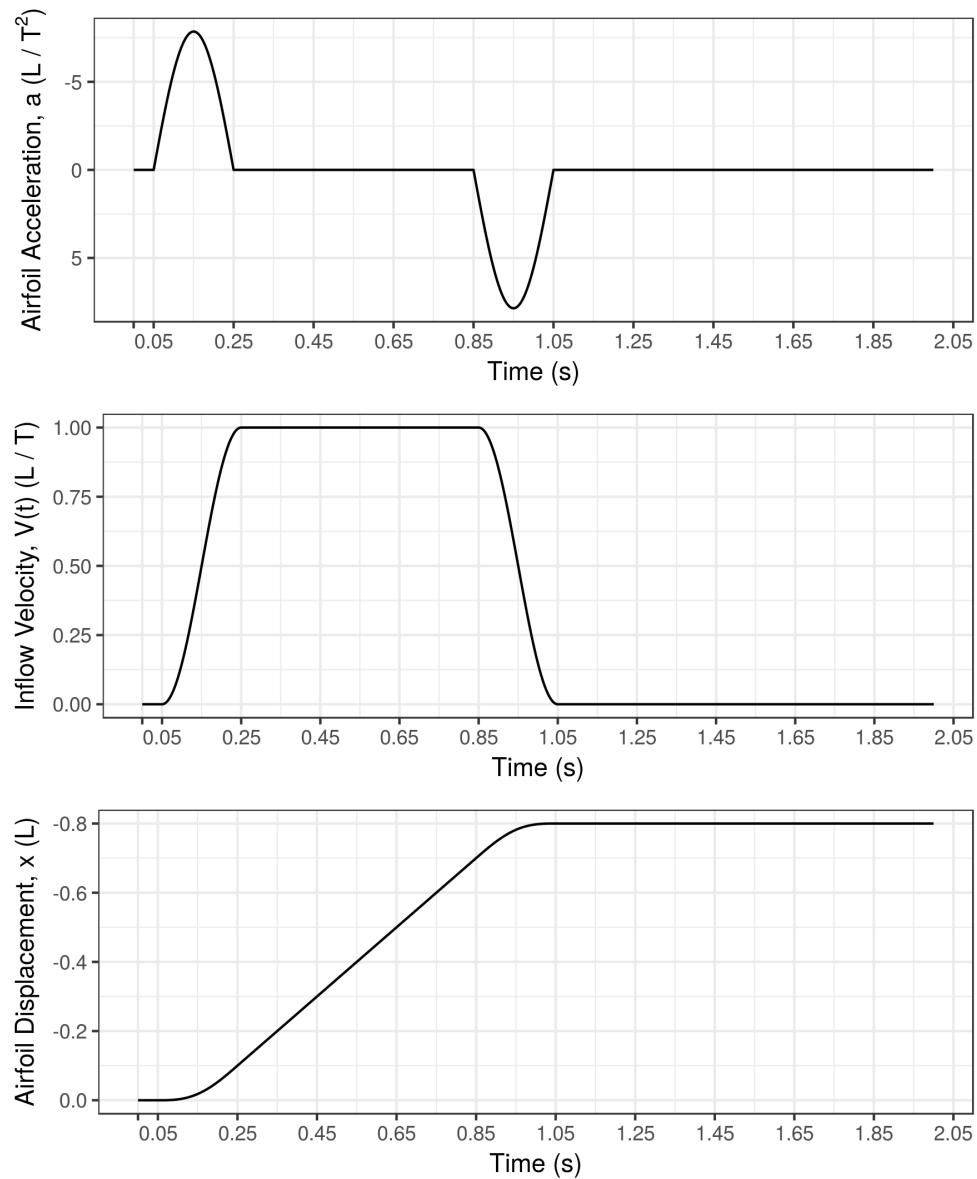


Figure 2.1: Boundary conditions imposed on the simulation during the starting and stopping phases. Note that the definition of x is consistent with the solution domain of the acceleration reference frame and the vertical axis has been inverted for airfoil properties.

Post Processing for Vortex Generation

For motion of a two-dimensional plane boundary moving in its own plane with velocity $\mathbf{V} = (V(t), 0)$, the diffusive flux density (flow per unit length per unit time) of positive vorticity outwards from the wall is given as

$$-\sigma \equiv -\nu \frac{\partial \vec{\omega}}{\partial z} \Big|_{z=0} = -\vec{n} \times (\nabla \vec{P} + \vec{a}(t)) \quad (2.4)$$

It is assumed that an infinitesimally small local section of airfoil can be modelled as an infinite plane with negligible curvature. The acceleration of the plane is given by $\vec{t} \cdot \vec{a}$ where \vec{t} is a unit tangent vector as shown in figure 2.2. Thus, the vorticity production around the airfoil is given by equation 2.5a in vector form and equation 2.5b for a particular point on the airfoil.

$$-\nu \vec{n} \cdot \nabla \vec{\omega} = -\vec{n} \times \nabla \vec{P} - \vec{t} \cdot \vec{a}(t) \quad (2.5a)$$

$$\underbrace{-\nu \frac{\partial \omega}{\partial z}}_{\text{"LHSG"}} = \underbrace{-\frac{1}{\rho} \frac{\partial p}{\partial s} - \frac{dV(t)}{dt} \cos(\theta)}_{\text{"RHSG"}} \quad (2.5b)$$

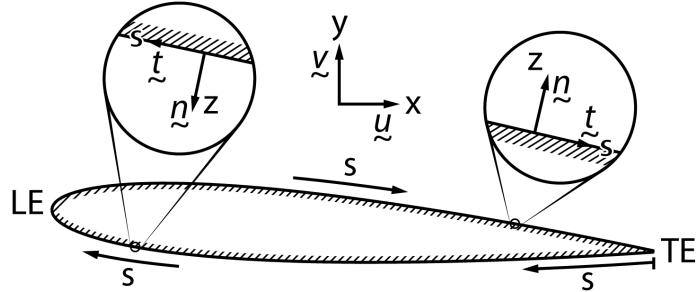


Figure 2.2: Coordinate system (s, z) used to traverse the surface of the airfoil. Directions of the tangent and normal vectors, \vec{t} and \vec{n} , are shown.

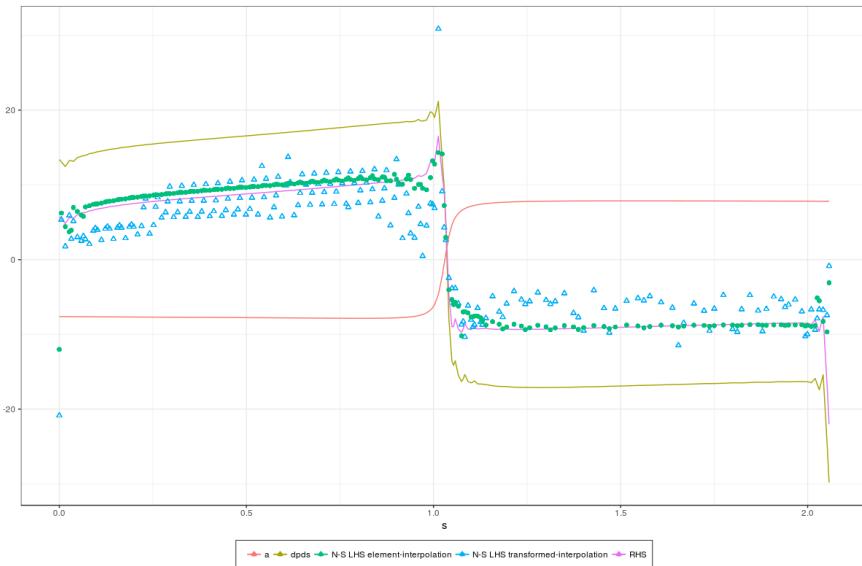


Figure 2.3: Comparison of methods to determine partial derivatives. RHS of equation (2.5b) shown in pink, element-wise finite difference shown by green dots and finite difference on a mapped plane shown by blue triangles.

Initially, two different methods were attempted. The first method used a finite difference method where four points normal to a node on the surface to estimate $\left(\frac{\partial \omega}{\partial z}\right)_0$ using a one-sided finite difference method for five equally spaced points, z_0, z_1, \dots, z_4 given by Fornberg [6]. Bicubic interpolate of points, z_0, z_1, \dots, z_4 , was used as this would give an accuracy of similar order to the finite difference used. However, a triangulation algorithm was used to determine the interpolation for irregularly spaced data. As such, a point that lay in an element on the lower surface, e.g. element 336, may use point(s) from the upper surface, e.g. element 143, as part of the triangulation. This lead to significant errors at the trailing edge.

$$\frac{\partial \omega}{\partial z}(z_0) \approx \frac{-\frac{25}{12}\omega(z_0) + 4\omega(z_1) - 3\omega(z_2) + \frac{4}{3}\omega(z_3) - \frac{1}{4}\omega(z_4)}{h} + O(h_z^4) \quad (2.6)$$

The second method was to recast the problem in the (x, y) domain to one in the (s, z) domain. Each point in (x, y) was mapped to the (s, z) domain forming a rectangular grid where the field properties could be interpolated. While the trailing edge problem was avoided, the coordinate transform introduced such great errors that any benefit was lost. This is seen in figure 2.3 where neither method provides a smooth solution.

Finally, an accurate method was determined by modifying `addfield` to determine the values of $\omega, \frac{\partial \omega}{\partial x}, \frac{\partial \omega}{\partial y}, \frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}$. The modification is shown in listing 1. By using the underlying GLL shape functions, $\nabla \vec{\omega}$ and $\nabla \vec{p}$ were determined by more accurately.

In a similar approach of using GLL shape functions, integration performed for the circulation, Γ , and vortex impulses, \vec{P} , were calculated using `massmat` for element-by-element point-by-point weightings m_i . The output of this method was compared against Semtex's `integral` function for the circulation of each individual element. The maximum per-element error was in the order of 10^{-7} while the error for the total circulation of the field was in the order of 10^{-10} .

$$\iint_{\text{elem}} (\omega \times x) dA = \sum_{i=1}^{N_P^2} (\omega \times x \times m_i) \quad (2.7)$$

Grid and Time Step Refinement

It is clear that the convergence of second derivatives for u and v are required to accurately determine $\nabla \vec{w}$. Several refinements were used in addition to testing different methods for determining partial derivatives. These included refining the mesh at the leading and trailing edges and increasing the order of the tensor-product GLL shape functions used in the spectral element, N_P . For the following convergence study, a Reynolds number of $Re = 10,000$ is used, $\nu = 0.0001$, and it is assumed for lower Reynolds numbers, e.g. $\nu = 0.0001$, the solution is well converged in space and time.

The starting point for the mesh was the same mesh as Vincent and Blackburn [18] and two major modifications were made. The first modification was that upper and lower airfoil splines were joined into a single spline to better represent the leading edge of the airfoil. The second modification was to increase the mesh density at the leading and trailing edges. The new mesh is referred to as “NACA0012r” and the original mesh is referred to as “NACA0012”.

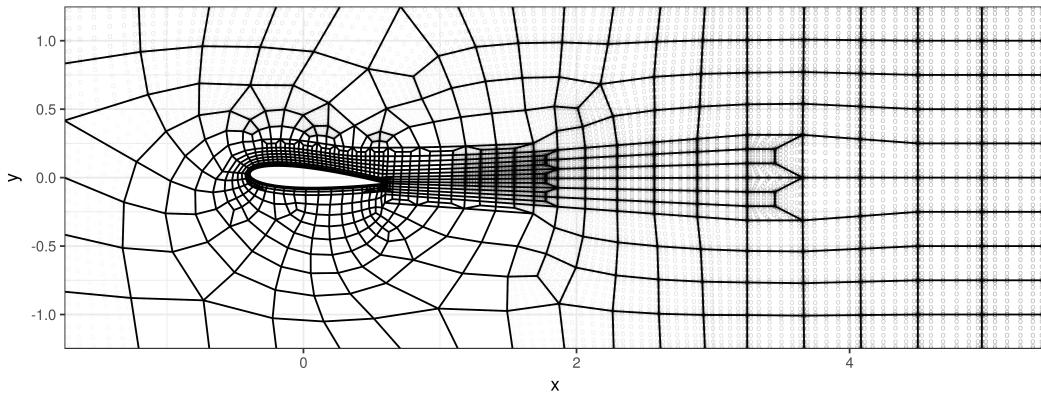


Figure 2.4: Spectral element mesh of NACA0012 with unit chord at $\alpha = 4^\circ$

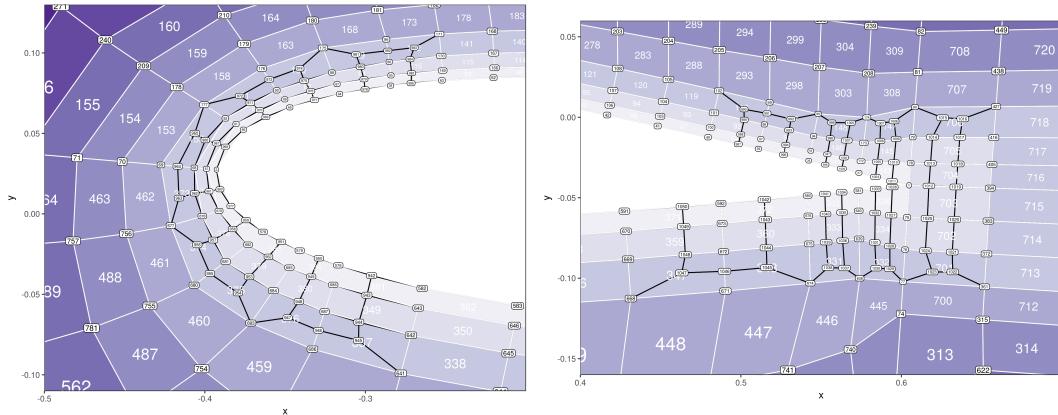


Figure 2.5: Increased mesh density shown in black for the leading edge (left) and trailing edge (right).

To assess Semtex's convergence of derivatives of partial derivatives, the test of case of planar steady Kovasznay flow was used from Blackburn and Sherwin [4]. This is shown in equation (2.8) where $\lambda = \frac{\text{Re}}{2} - \left(\frac{\text{Re}^2}{4} + 4\pi^2 \right)$. The analytical solution for $\frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}, \omega, \frac{\partial \omega}{\partial x}, \frac{\partial \omega}{\partial y}$ were determined and a convergence analysis performed. The norm_inf, maximum absolute difference, is used to determine the error between numerical and analytical solution. It can be seen in figure 2.6 that the convergence of the partial derivatives is approximately at the same rate of the primary variables u , v and p . However, for the same norm_inf, a much higher N_P is required for higher partial derivatives.

$$\begin{aligned}
 u &= 1 - e^{\lambda x} \cos(2\pi y) \\
 v &= \frac{2\pi}{\lambda} e^{\lambda x} \sin(2\pi x) \\
 p &= \frac{1}{2} (1 - e^{\lambda x})
 \end{aligned} \tag{2.8}$$

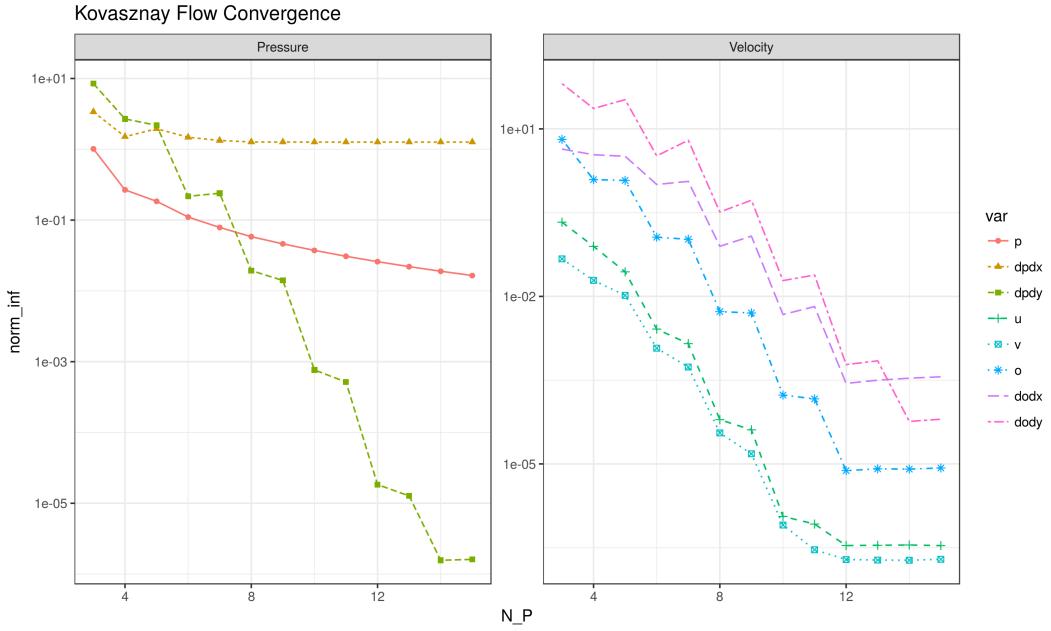


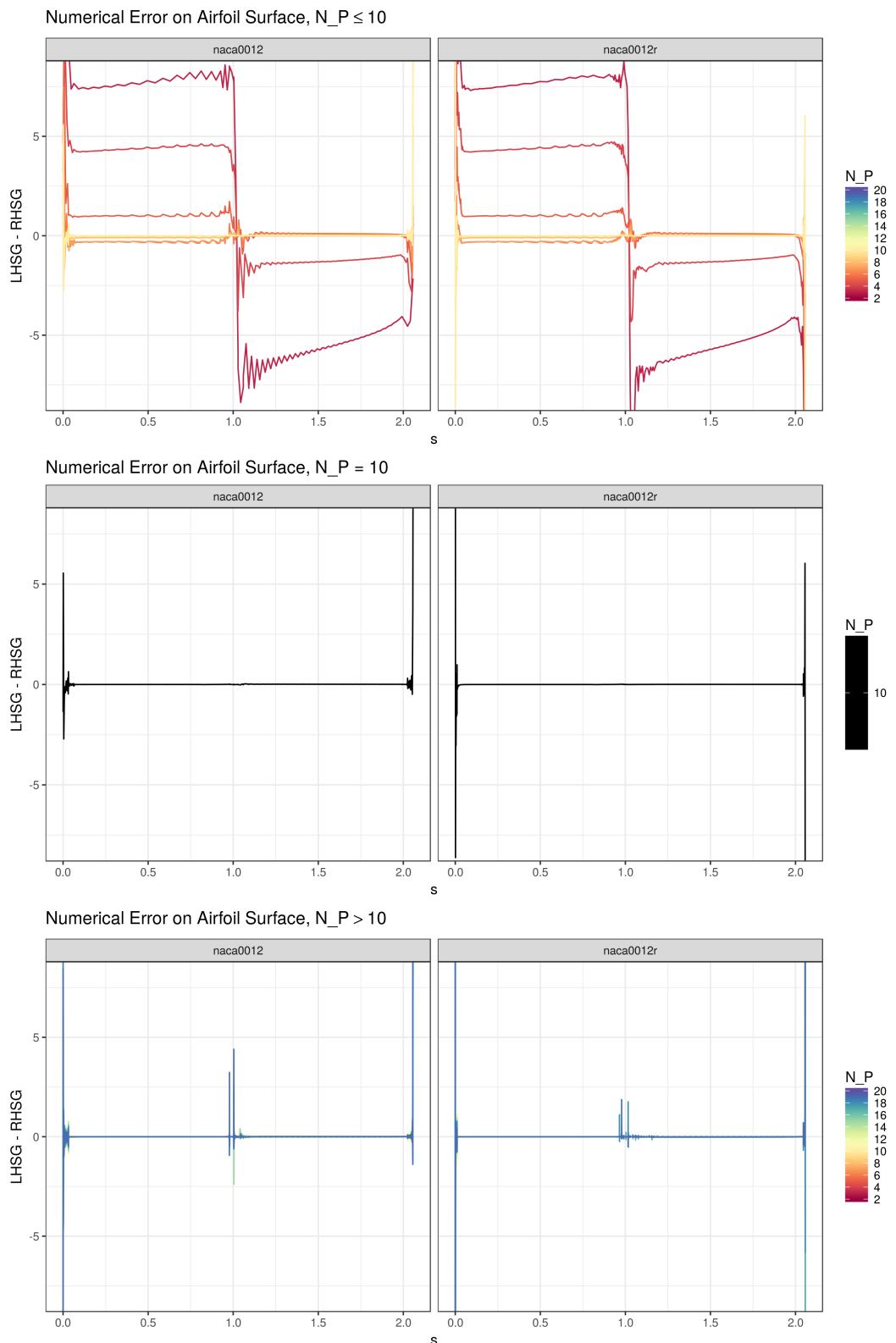
Figure 2.6: p -Convergence study of the variables of interest for a planar Kovasznay flow

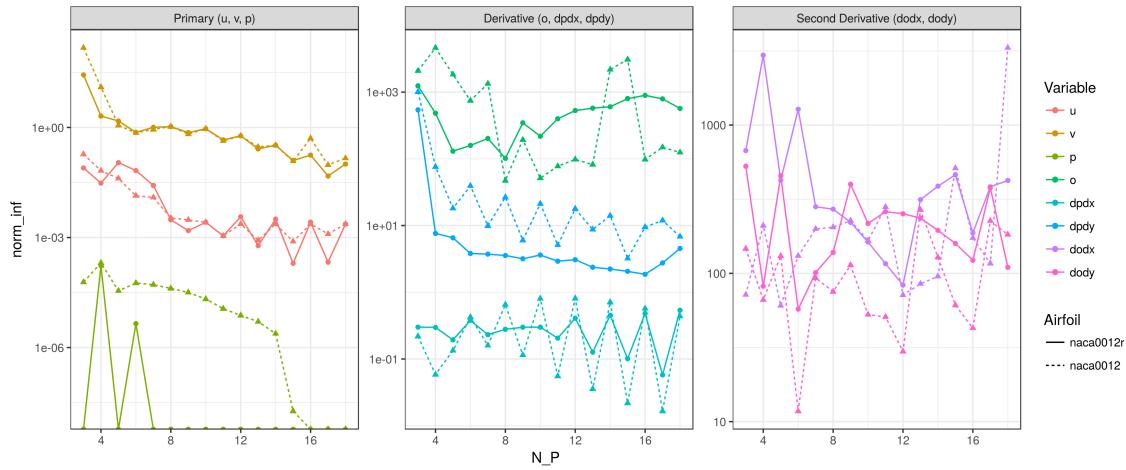
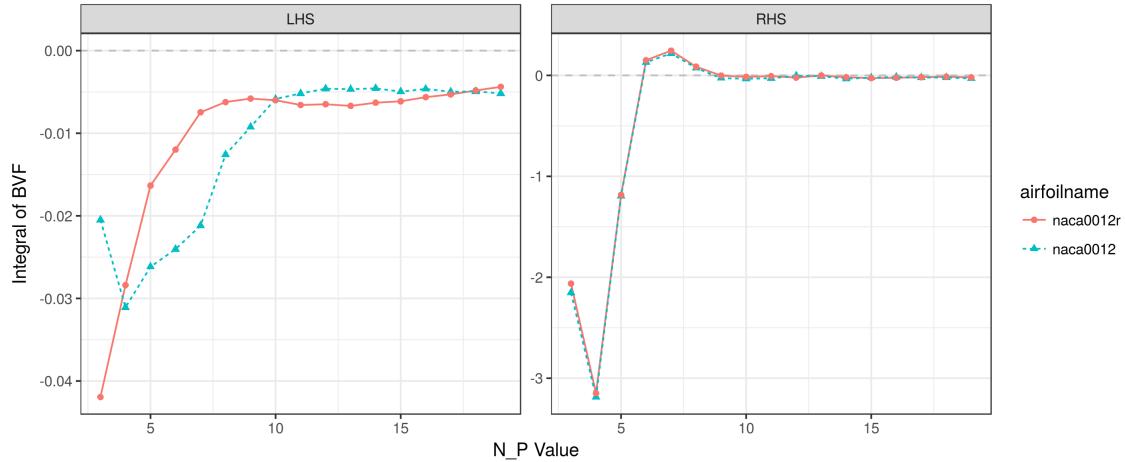
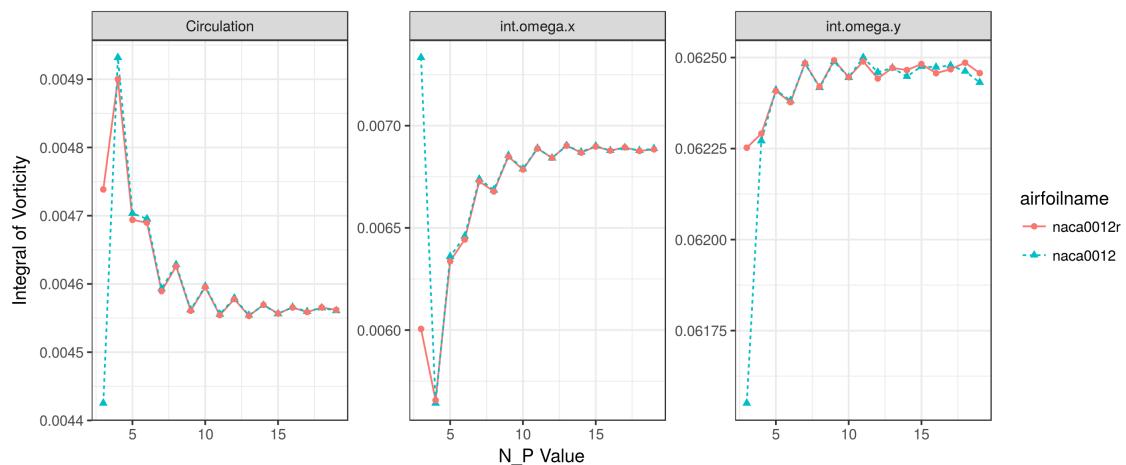
As such, numerical convergence of all variables for this problem must be investigated, not just u , v and p . To determine an appropriate choice for N_P , tests were conducted at $t = 0.15$ s which corresponds to the maximum forwards ($a(t) < 0$) acceleration of the airfoil. A p -convergence test where p is the order of the tensor-product (N_P) was performed. Values between 3 and 18 were used and their values compared to $p = 19$ is shown in figure 2.8. While the primary variables and first derivatives are well converged at $N_P \geq 9$, it is seen that the second derivatives of u and v are still highly variable. In general, it is seen that NACA0012r has better convergence due to the increased elements at the leading and trailing edges.

For vorticity generation, it is seen in equation (2.5b) that the “LHSG” vorticity derivative should be equal to the “RHSG” sum of pressure derivative and tangential acceleration. The difference between these two can be attributed to numerical error and should be minimised for accurate results of vorticity generation on the surface. Figure (2.7) shows that the increased leading edge and trailing edge greatly reduces the numerical error. However, at high N_P values, substantial “wiggles” are introduced. Thus, a value of $N_P = 10$ was decided.

The convergence of the integral of BVF, given in equation (1.4) was also checked. The value of this integral should be zero, and can be evaluated based on the “RHSG” or “LHSG” of equation (2.5b). Both of these integrals approach 0 as N_P is increased shown in figure (2.9) which is expected. The total circulation and the vortex impulse were also investigated and their results shown in figure (2.10).

$$\oint_{\Gamma} -v \vec{n} \cdot \nabla \vec{\omega} \, ds = \oint_{\Gamma} \underbrace{-v \frac{\partial \omega}{\partial n}}_{\text{"LHSG"}} \, ds = \oint_{\Gamma} \underbrace{-\vec{n} \times \nabla P - \vec{t} \cdot \vec{a}(t)}_{\text{"RHSG"}} \, ds = 0 \tag{2.9}$$

Figure 2.7: Numerical error of $-\nu \vec{n} \cdot \nabla \vec{\omega} + \vec{n} \times \nabla \vec{P} + \vec{t} \cdot \vec{a}(t) = 0$ for $3 \leq p \leq 19$

Figure 2.8: p -Convergence study of the variables of interest for $3 \leq p \leq 18$ Figure 2.9: Convergence of the integral of the BVF around the airfoil for $3 \leq p \leq 19$ Figure 2.10: Convergence of the integral of the vorticity and vortex impulse for $3 \leq p \leq 19$

Part II

Results and Discussion

3 Vorticity and Vorticity Generation

3.1 Airfoil Boundary Properties

Integral of Boundary Vorticity Flux

In theory, according to equation (1.4), the boundary vorticity flux should integrate to zero around the surface of the airfoil. In the same manner as the convergence study, the integral of the BVF was computed but now at varying time steps. It can be seen below that again the LHS and RHS do not agree with large discrepancies during the starting and stopping phases. It is difficult to ascertain if this is due to numerical error indicating insufficient convergence and/or the presence of a singularity of pressure at the trailing edge as equation (1.4) assumes a continuous pressure field. As this was used as the boundary condition for Zhu et al. [24], if this condition does not hold then it casts doubt on the results obtained.

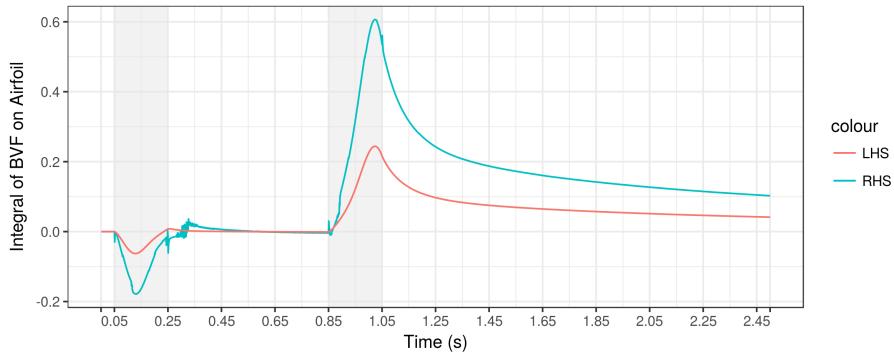


Figure 3.1: Integral of BVF versus time $0 \leq t \leq 2.5$ for $Re = 1,000$

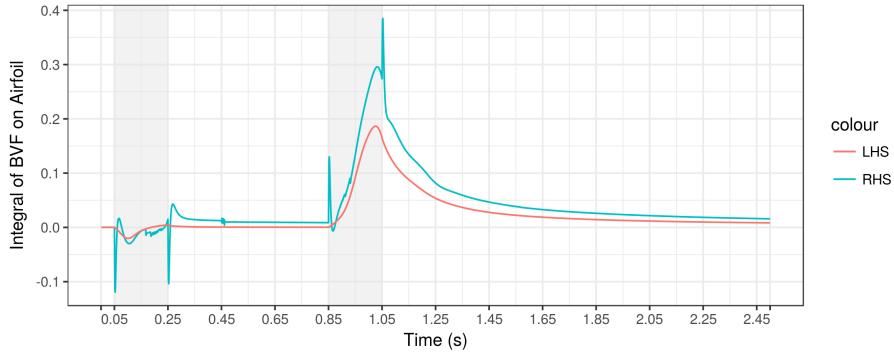
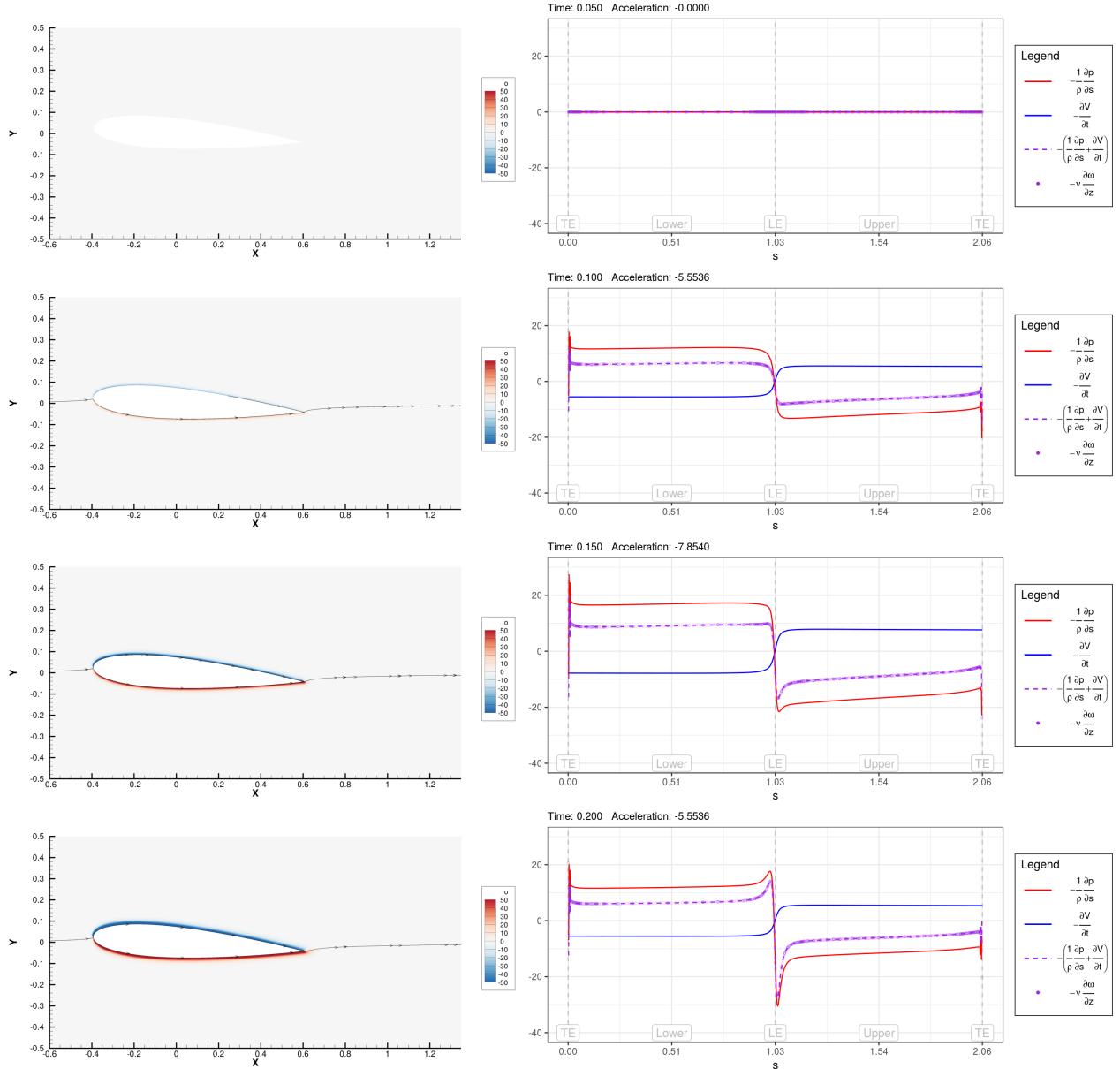
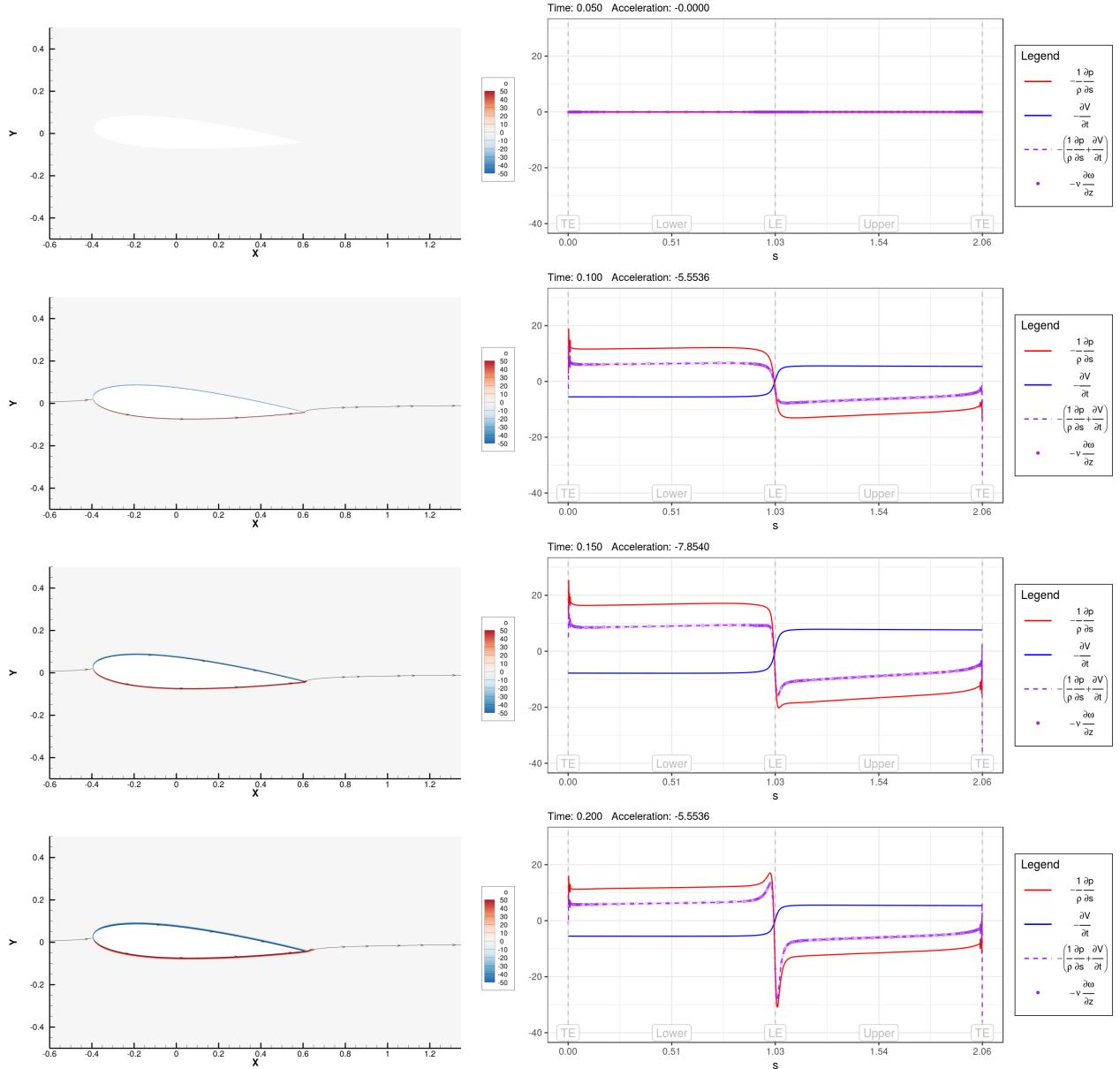
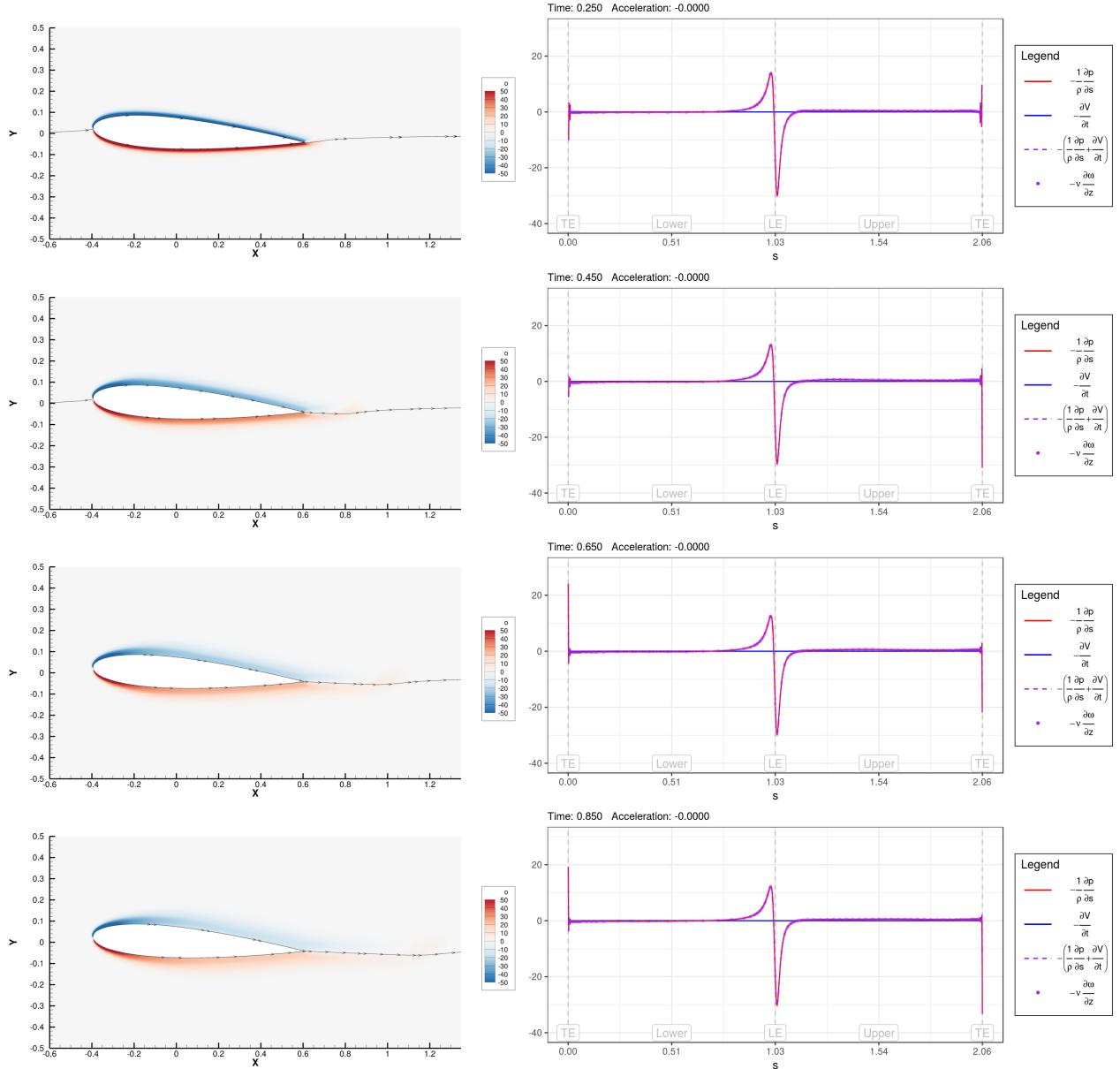
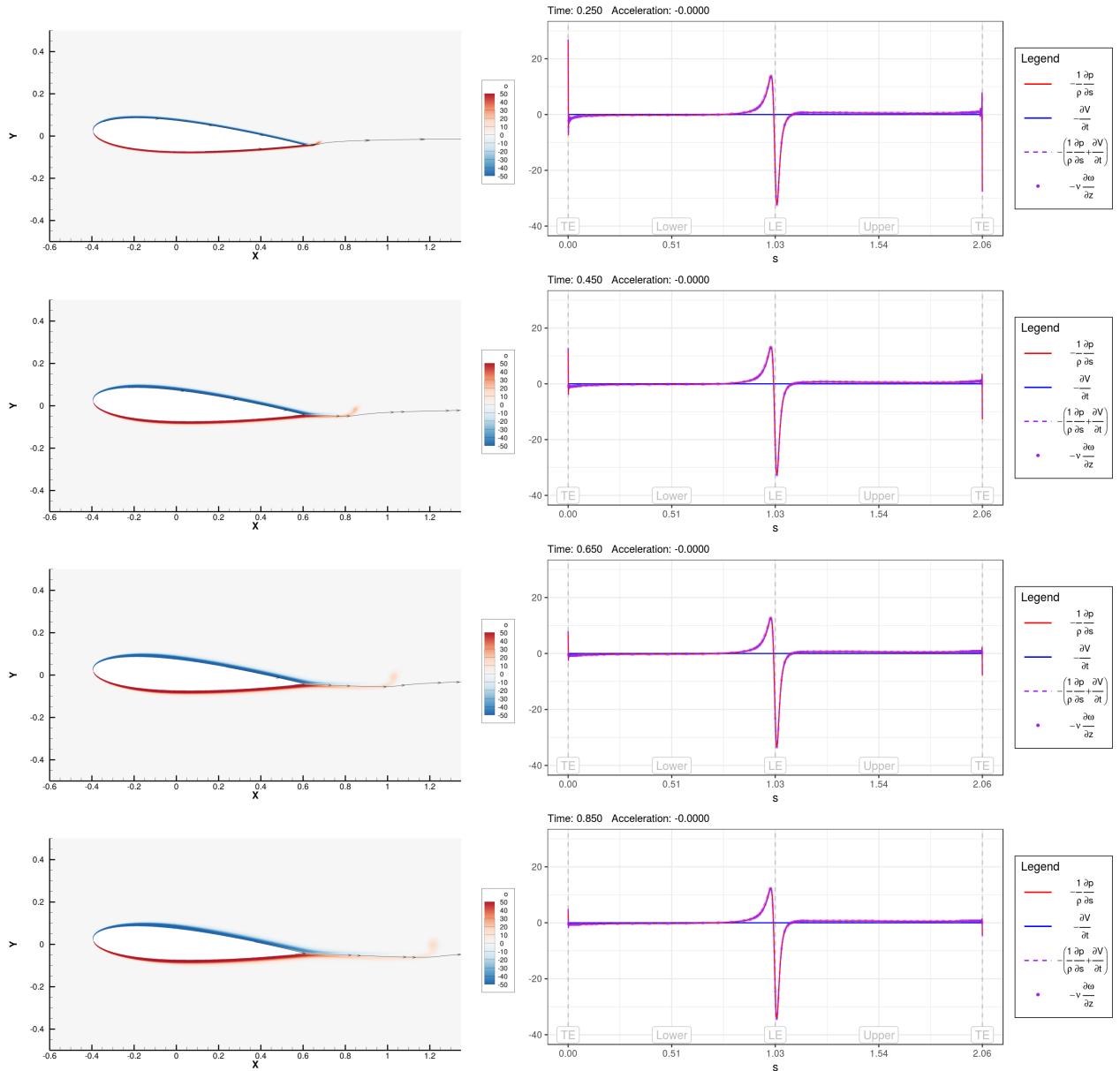


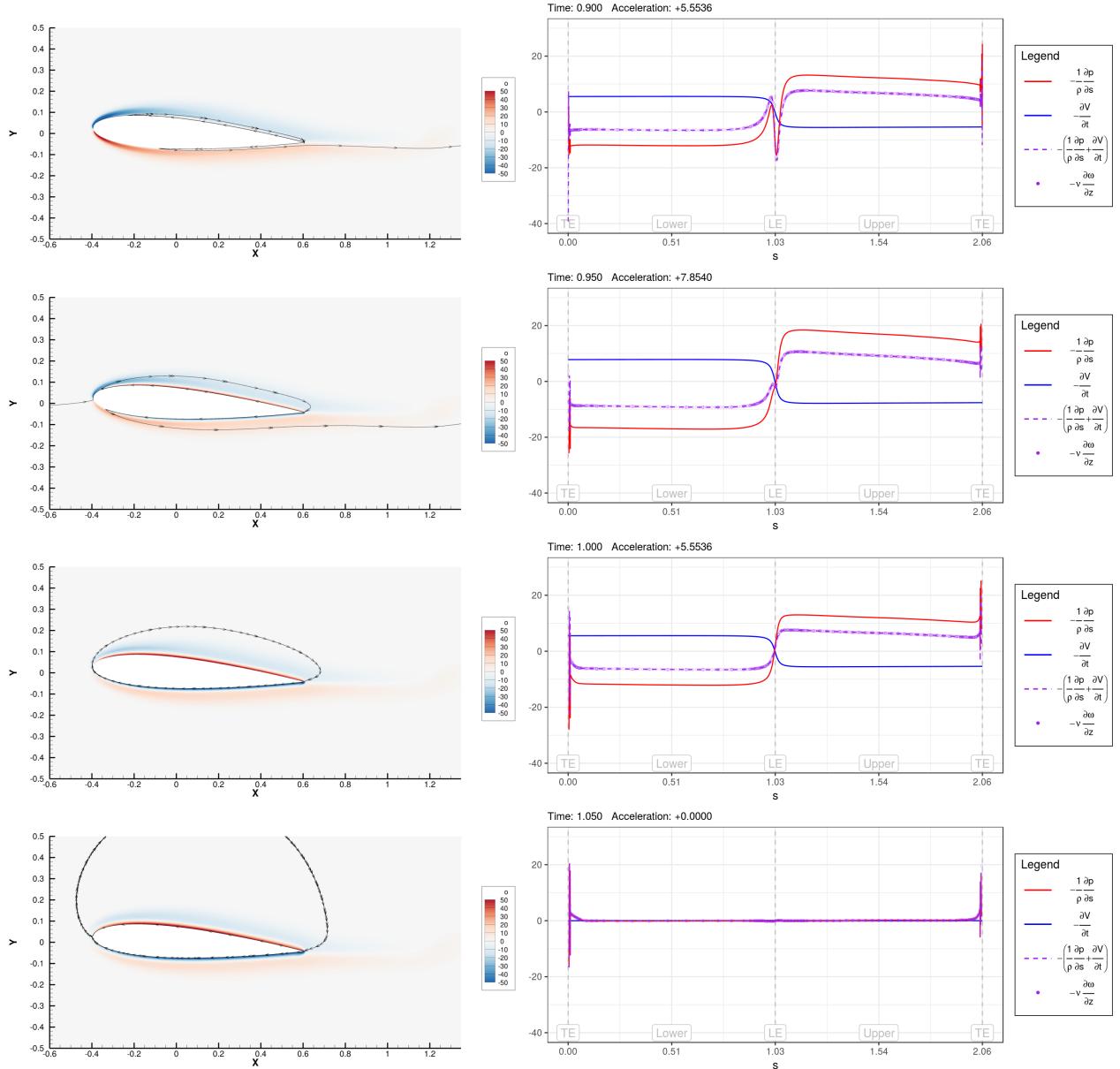
Figure 3.2: Integral of BVF versus time $0 \leq t \leq 2.5$ for $Re = 10,000$

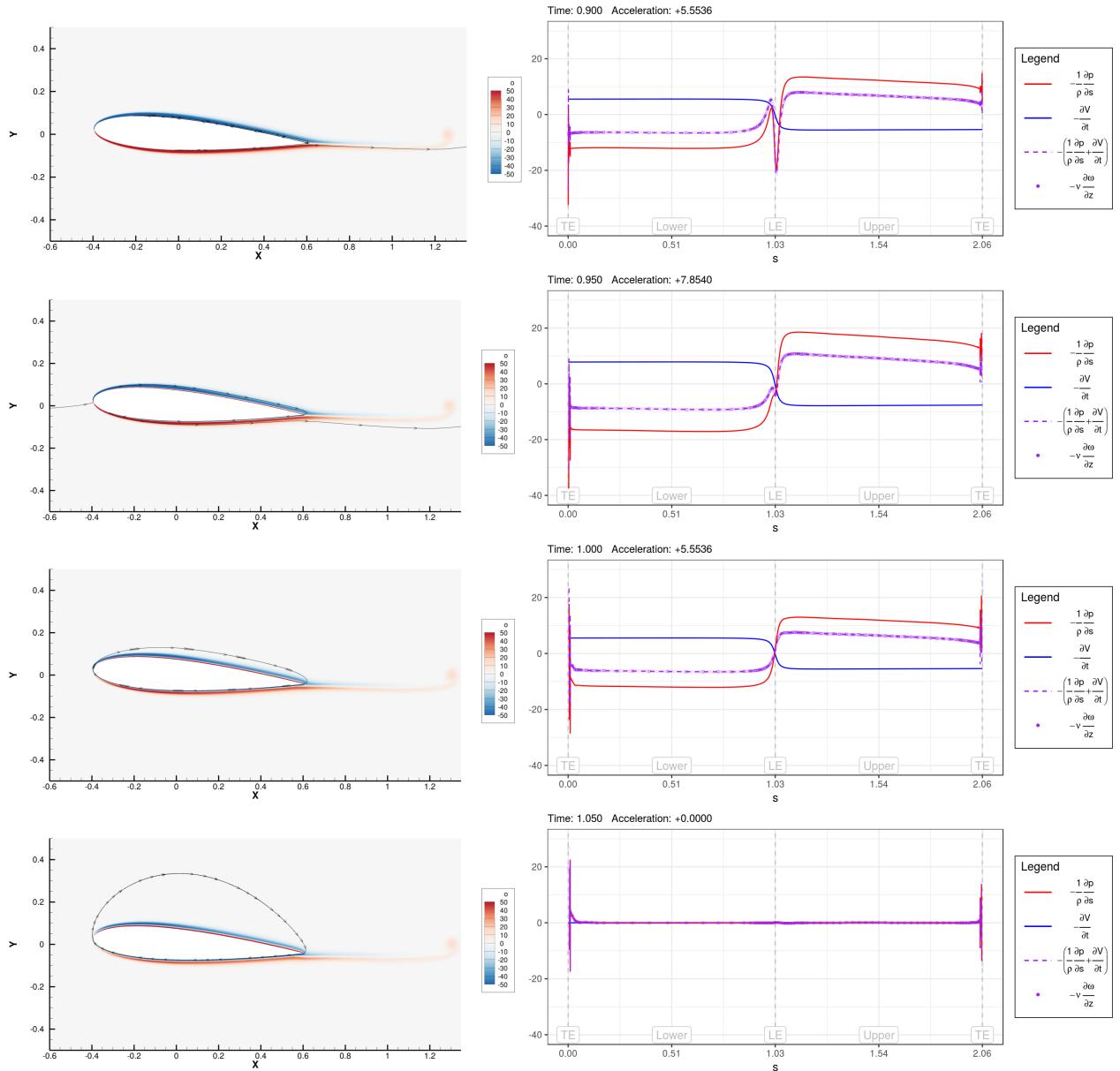
Figure 3.3: Acceleration stage for $Re = 1,000$. Top to bottom: $t = 0.05, 0.10, 0.15, 0.20$

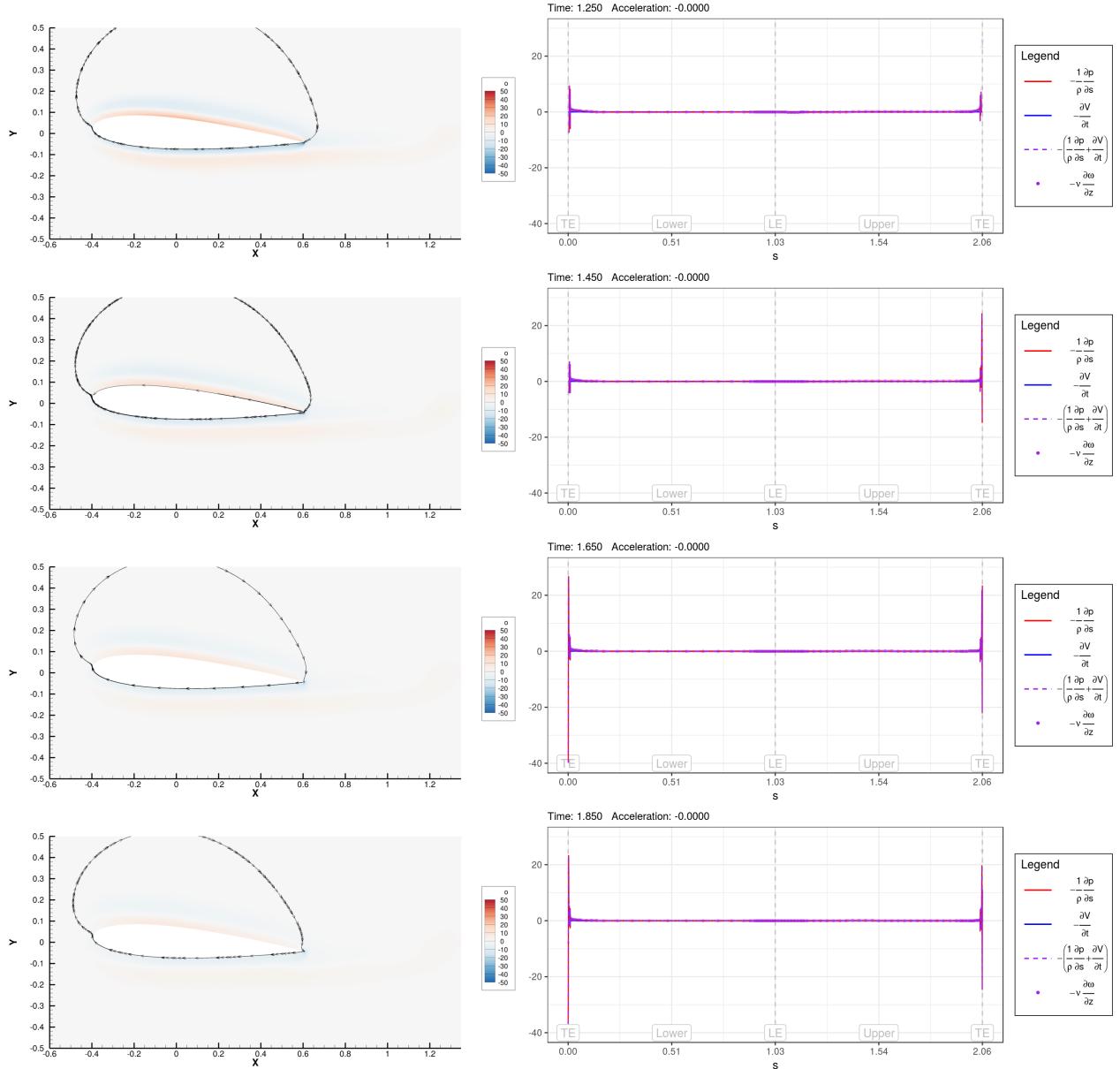
Figure 3.4: Acceleration stage for $Re = 10,000$. Top to bottom: $t = 0.05, 0.10, 0.15, 0.20$

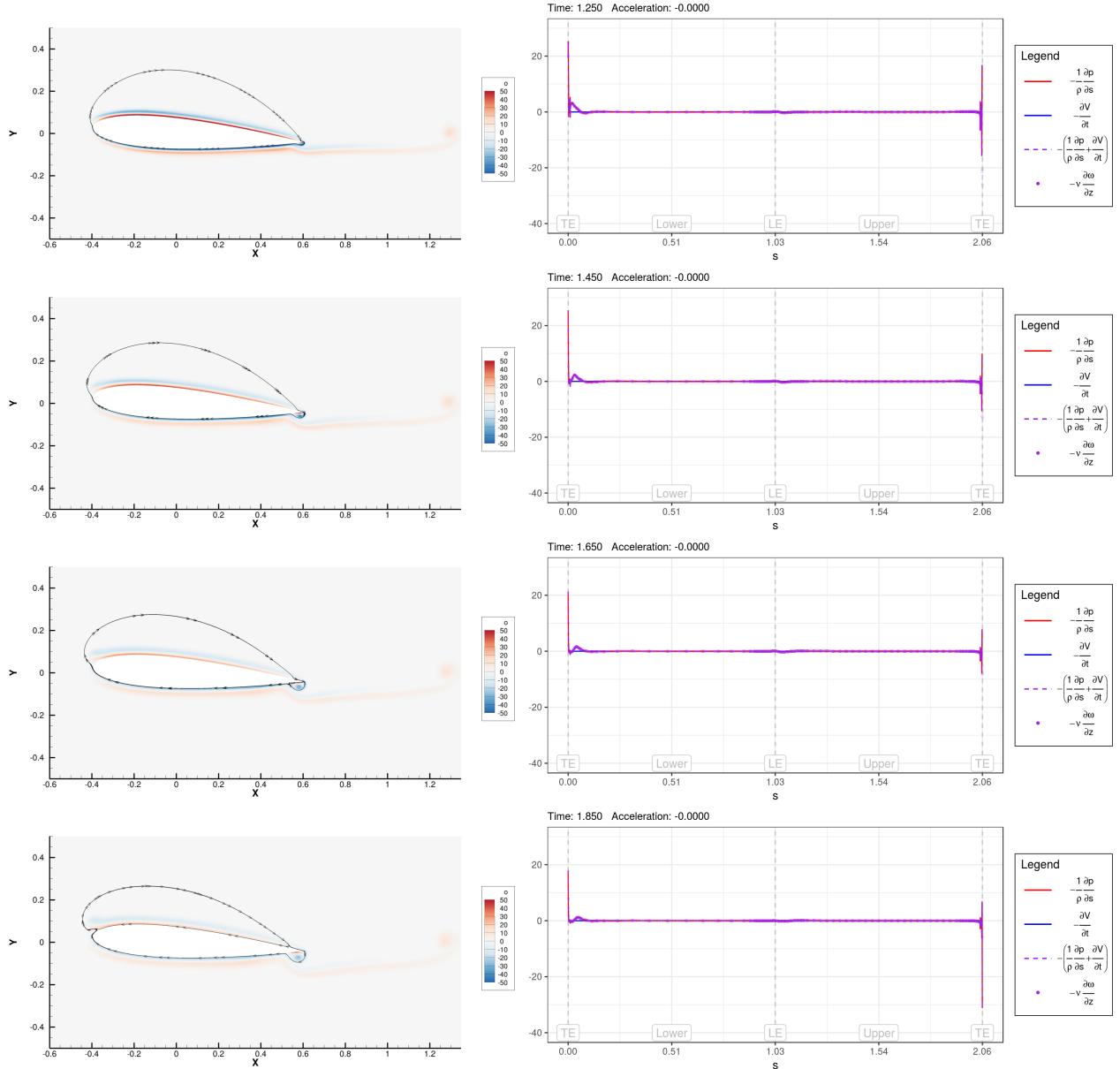
Figure 3.5: Steady-state stage for $Re = 1,000$. Top to bottom: $t = 0.25, 0.45, 0.65, 0.85$

Figure 3.6: Steady-state stage for $Re = 10,000$. Top to bottom: $t = 0.25, 0.45, 0.65, 0.85$

Figure 3.7: Deceleration stage for $Re = 1,000$. Top to bottom: $t = 0.90, 0.95, 1.00, 1.05$

Figure 3.8: Deceleration stage for $Re = 10,000$. Top to bottom: $t = 0.90, 0.95, 1.00, 1.05$

Figure 3.9: Stopped stage for $Re = 1,000$. Top to bottom: $t = 1.25, 1.45, 1.65, 1.85$

Figure 3.10: Stopped stage for $Re = 10,000$. Top to bottom: $t = 1.25, 1.45, 1.65, 1.85$

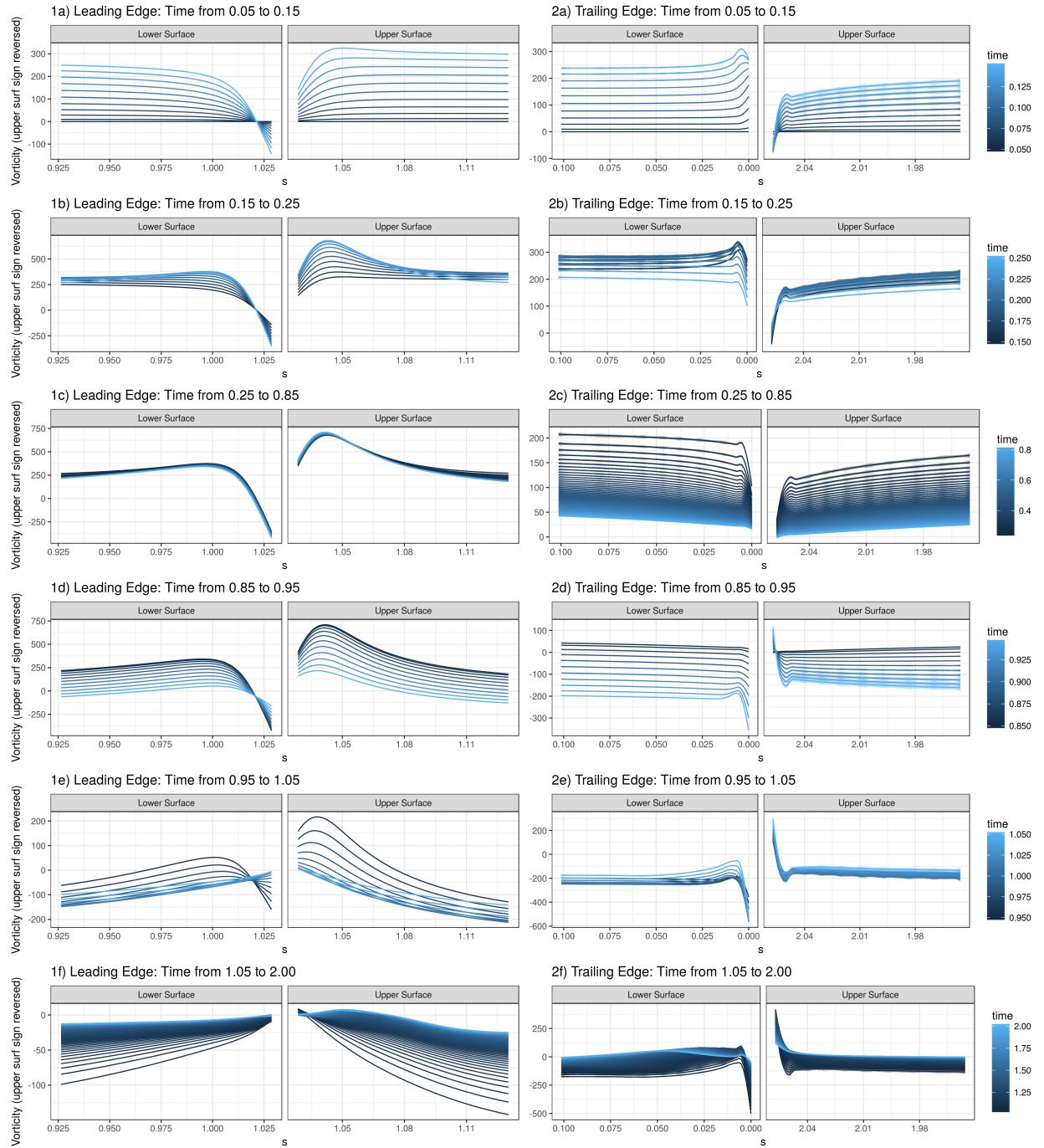


Figure 3.11: Vorticity on the airfoil surface for the first 10% of the chord and last 10% of the chord for $Re = 10,000$

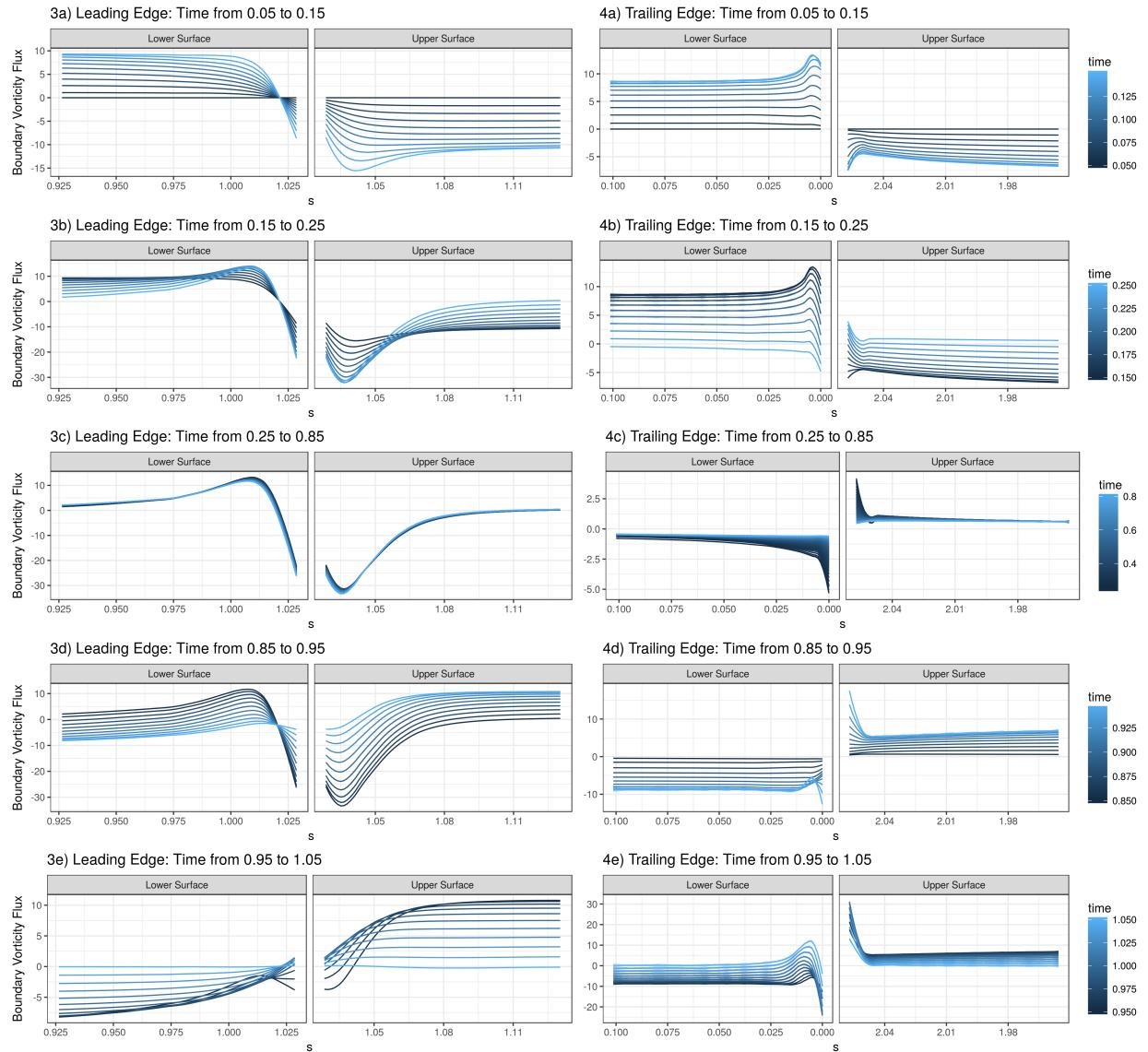


Figure 3.12: Vorticity production on the airfoil surface for the first 10% of the chord and last 10% of the chord for $Re = 10,000$

Vorticity Generation Mechanisms

The figures on the preceding pages illustrate the vorticity distribution around the airfoil during each of the stages and the associated vorticity generation mechanisms. Blackburn and Henderson [3] noted that the two generation mechanisms are in competition and this can be seen in both the acceleration and deceleration stages. In both the acceleration and deceleration stages, the sign of the acceleration mechanism term $\frac{dV}{dt}$ is opposite to the boundary vorticity flux term, σ . This indicates that the surface-acceleration vorticity generation opposes the pressure gradient vorticity generation but is not strong enough to override which is similar to the results of Blackburn and Henderson [3].

As the pressure gradient vorticity generation can be understood through roll-up of shear layers, it was hypothesised that reducing the viscosity would lead a change in the acceleration and pressure gradient generation mechanisms. However, comparing figure 3.3 with figure 3.4 and figure 3.7 with 3.8 there is little difference between the vorticity generation curves (apart from near the trailing edges) even though the higher Reynolds number simulation has much thinner vorticity regions.

When there is zero acceleration, seen in figures 3.5, 3.6, 3.7 and 3.8, the BVF σ is approximately constant in both time and viscosity. Since σ can be integrated around the surface to determine the lift force (ignoring viscosity) in equation 1.9 this is unsurprising as the lift force is not expected to greatly vary. For $Re = 10,000$,

3.2 Establishment of the Kutta Condition

Starting Phase

The initial objectives of this project, investigating the Kutta condition was an objective. However, even at high polynomial orders, low time steps and low Reynolds numbers, it was not possible to recreate the visualisations by Zhu et al. [24]. Figure 3.11 shows the vorticity on the upper and lower surfaces for the leading and trailing edges. 1b and 2b correspond to the starting phase and there is a change in sign on vorticity on the upper surface at the trailing edge as suggested by Zhu et al. However, the corresponding flow visualisation, figure 3.4, shows that the flow does not leave the trailing edge “smoothly” during the starting flow. Rather, throughout the acceleration phase, the stagnation point appears to stay on the upper surface rather than move to the trailing edge.

In their paper, Zhu et al. [24] identified the critical event at $t = 0.029$ s (instantaneous $Re(t) = 4554$, steady state $Re = 100,000$) for an acceleration profile $\sin\left(\frac{\pi t}{2}\right)$, $t < 1.0$ s. The boundary vorticity distribution for $t = 0.020$ s is also shown with a vortex bubble visible. However, the visualisations from Semtex DNS at a similar time length and Reynolds scale ($t = 0.144$ s, instantaneous $Re(t) = 4554$, steady state $Re = 10,000$) does not show any sign of a separation bubble (figure 3.13). The positive vorticity region merely appears to move downstream with the white, irrotational sliver rotating from approximately 90° to the x -axis to parallel with the x -axis. This is consistent with the figure 3.11 showing a small, single change in sign of vorticity not a double change in sign for a vortex bubble.

One explanation could be due to the solution not being fully converged or insufficient mesh density at the trailing edge as vortex bubble cannot be captured if it wholly exists between two mesh node points on the surface. The vortex generation shown in 4a and 4b of figure 3.12 is similar to the results of Zhu et al. [24] showing a peak in the BVF at the trailing edge.

Stopping Phase

During the stopping phase, figures 3.7 and 3.8 show positive vorticity on the upper surface and negative vorticity on the lower surface as the shear layers come to rest. This new vorticity is of opposite sign to the existing vorticity which is displaced instead of being reinforced. The vorticity generation demonstrates this as the BVF is of opposite sign to the existing vorticity.

Even after the deceleration has finished, the vorticity field is still evolving. In figures 3.9 and 3.10, the trailing vortex is just beginning to form and the pressure generation mechanism is seen to be causing the generation of positive vorticity on the lower trailing edge, displacing the negative vorticity of the

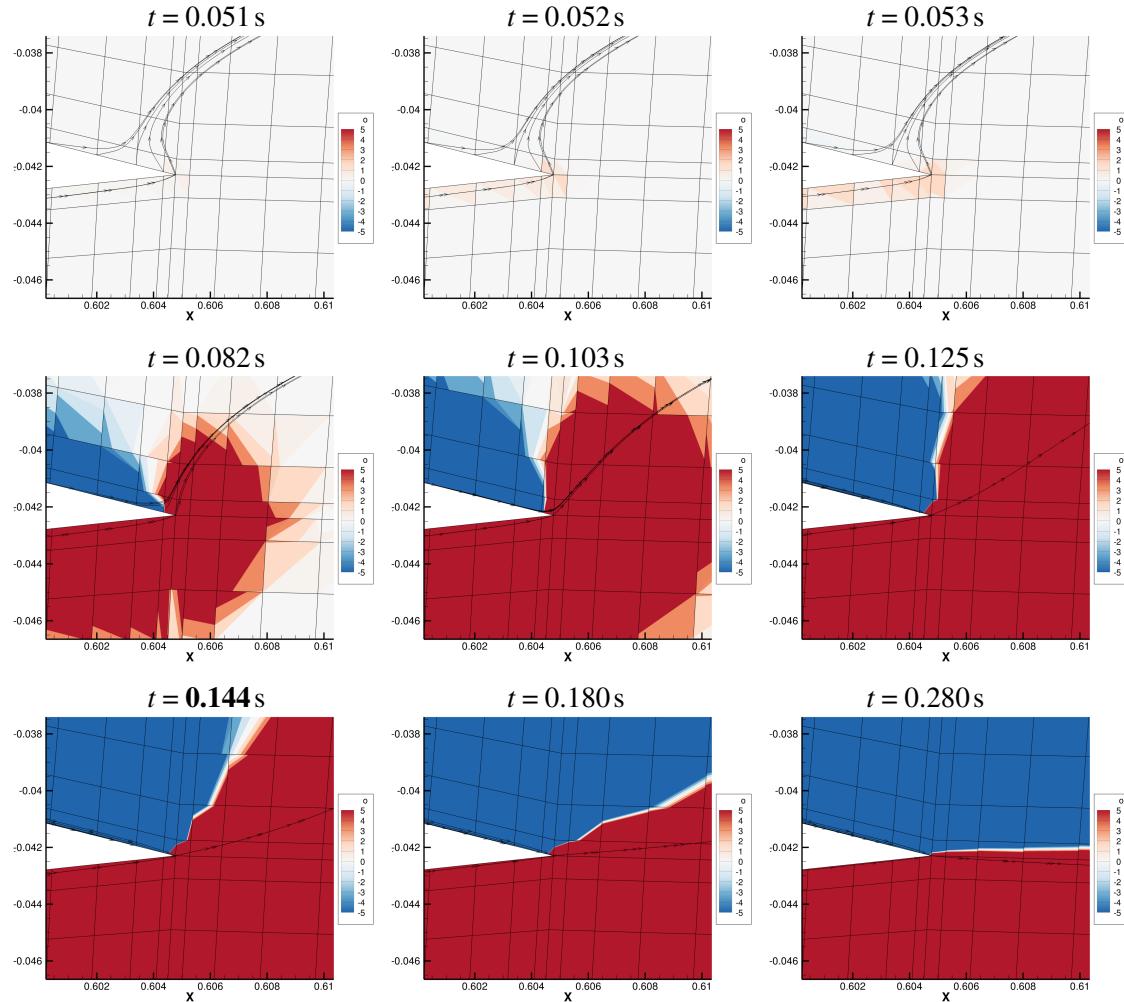


Figure 3.13: Vorticity distribution around the trailing edge during the starting flow at various time steps. Left to right: $t = 0.051, 0.052, 0.053, 0.082, 0.103, 0.125, 0.144, 0.180, 0.280$ s. Note that no interpolation has been performed and the black grid corresponds to points in the original mesh.

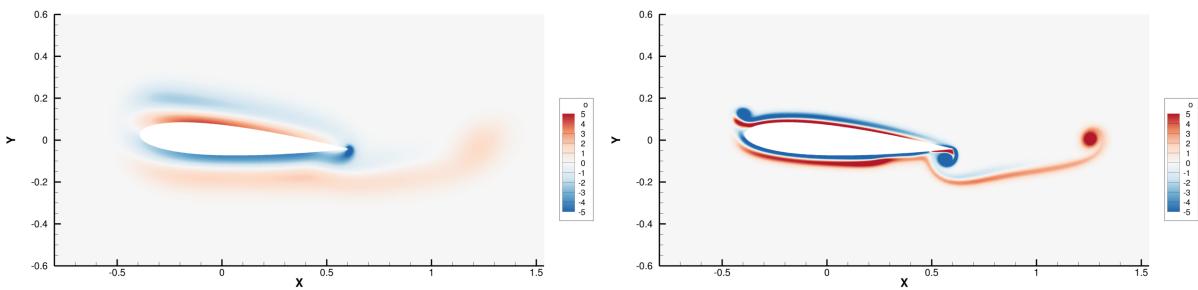


Figure 3.14: Vorticity field at $t = 2.5$ s for $Re = 1,000$ (left) and $Re = 10,000$ (right)

stopping vortex. For $Re = 10,000$, leading edge vortices are formed as well, but these are not seen in $Re = 1,000$ even after an additional 1.5 s (figure 3.14). This could be due to the vorticity diffusing away faster at lower viscosity instead of forming concentrated vortices.

3.3 Circulation of Airfoil and Shed Vortices

Circulation Regions

In order to assess the circulation of shed vortices, regions around the airfoil were defined shown in figure 3.15. The four regions are the elements adjacent but not behind the airfoil, AF, elements behind the trailing edge, E, elements on the upper surface, NW and elements on the lower surface, SW. In the typical description of the circulation bound to an airfoil, the sum of the Γ_{AF} , Γ_{NW} and Γ_{SW} are taken. In theory, this is equal and opposite to the circulation of the starting vortex in E.

$$\Gamma = \Gamma_{AF} + \Gamma_{NW} + \Gamma_{SW} \equiv -\Gamma_E \quad (3.1)$$

The primary issue with both figure 3.16 and figure 3.17 is that the total circulation is not zero for all of time. This indicates that vorticity must be leaving the solution domain, which can only be fixed by increasing the solution size as the circulation was determined to be converged already. A clear difference is seen between 3.16 and figure 3.17 in the circulation of the airfoil, Γ_{AF} . This is due to the higher Reynolds number simulation having thinner vorticity layers seen in previous figures.

Starting Phase and Steady State

As the airfoil initially gains velocity, Γ_{AF} rapidly increases, attributed to the large positive vorticity and BVF at the trailing edge of the airfoil. This is in contrast to the notion the bound circulation, Γ , must be negative for the total circulation to be conserved (note that $\Gamma_E > \Gamma_{Total}$ so Γ should be negative). The negative circulation is seen to be in the NW region and is the largest in magnitude of all the circulations.

At higher Reynolds numbers, the vorticity is contained to a narrow region. From figures 3.3 and 3.4 it was seen that $\sigma = \nu \frac{\partial \omega}{\partial z}$ is approximately constant for both Reynolds number simulations. Thus, $\frac{\partial \omega}{\partial z} \propto \frac{1}{\nu}$ which indicates lower vorticity gradients for lower Reynolds numbers which is consistent. This is observed through a smaller values of Γ_{NW} and Γ_{SW} at higher Reynolds numbers (more vorticity in the elements on the surface). For $Re = 1,000$ there is a clear dip in Γ_{AF} during the steady state, but this is believed to be due to the definition of circulation regions rather than a flow field phenomena.

Stopping Phase and Stopped State

During the stopping phase, Γ_E is seen to be constant as the stopping vortex does move far from the trailing edge (figures 3.7 and 3.8). Thus, Γ_E is approximately constant and so using $\Gamma \equiv -\Gamma_E$ would lead to an incorrect value of lift using $L = \rho V \Gamma$. The circulation Γ_{NW} and Γ_{SW} are seen to decrease during the stopping and stopped state due to vorticity of opposite sign being produced on the airfoil surface.

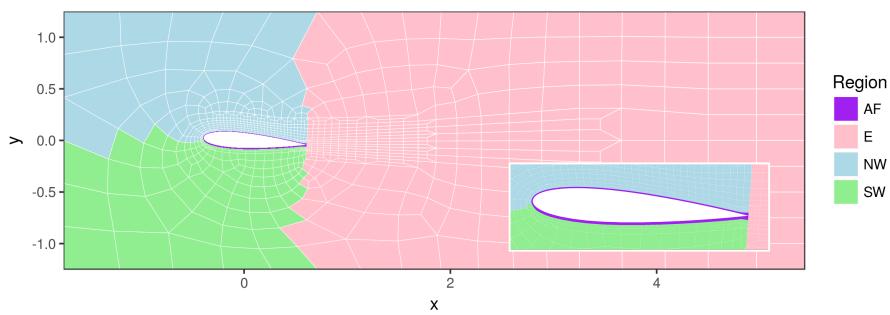


Figure 3.15: Definition of circulation regions. AF indicates the airfoil, E indicates “east”, NW indicates “north west” and SW indicates “south west” of the trailing edge.

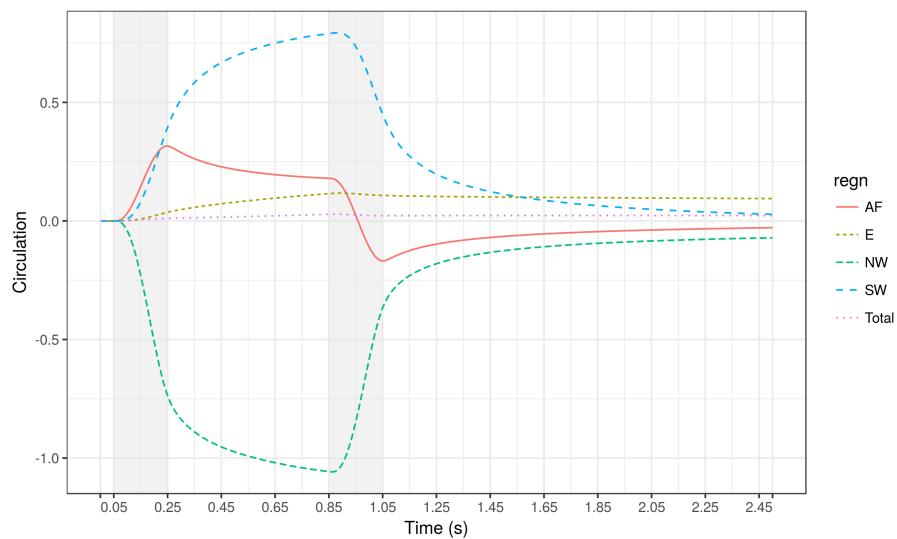


Figure 3.16: Circulation versus time $0 \leq t \leq 2.5$ for $\text{Re} = 1,000$

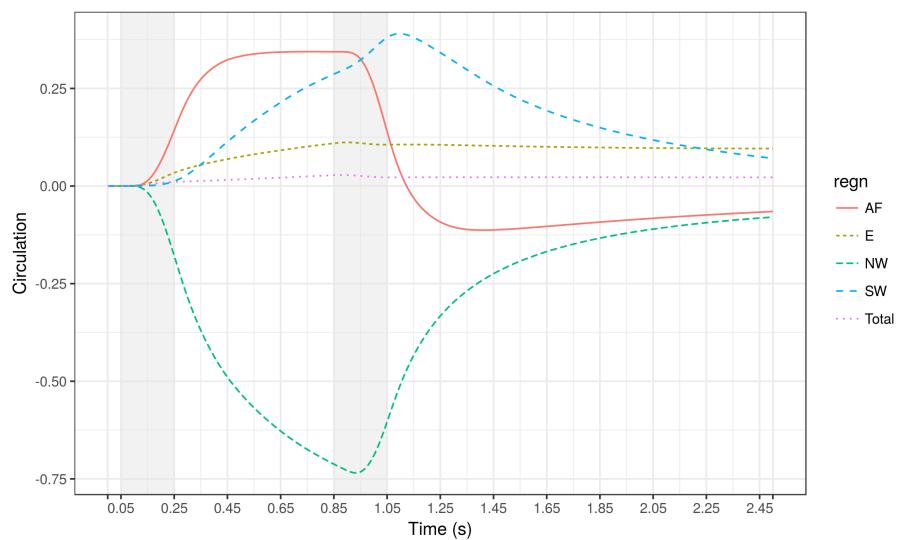


Figure 3.17: Circulation versus time $0 \leq t \leq 2.5$ for $\text{Re} = 10,000$

4 Comparison to Unsteady Potential Theory

4.1 Lift, Drag and Added-Mass Forces

Following Limacher, Morton and Wood's formulation 2 [12], equation (1.14), the vortex impulse was calculated. This is shown in figure 4.1 and it can be seen while the general shapes are the same there are big differences between the two due to the thickness of vorticity layers. For instance, oy is smaller for $Re = 10,000$ because narrower vorticity regions mean smaller values of y in the integrand of $oy = \int \omega \times y dA$. Taking time derivatives, the force due to vortical impulse can be found

$$\vec{P}_v = \int_V (\vec{x} \times \vec{\omega}) dV, \quad (oy, ox) \equiv \left(\int \omega y dA, - \int \omega x dA \right) \quad (4.1)$$

$$L_v = \frac{d}{dt}(ox), \quad D_v = \frac{d}{dt}(oy) \quad (4.2)$$

It is clear that figure 4.2 indicates that the vortex impulse does not, alone, provide enough information to explain the lift. Notably, during the steady state $0.25 \leq t \leq 0.85$ there is no added mass force so the discrepancy must be explained with other means, for example using Limacher, Morton and Wood's formulation 1 [12] which involves more terms. The "wiggles" during steady-state are also of concern as no other results have indicated large fluctuations in forces.

On the other hand, figure 4.3 indicates that the vortex impulse does a great deal to explain the drag force. The difference between the two can be determined as due to added mass terms in equation 1.14 which can be expressed as a coefficient of the acceleration, $F_{am} = -m \cdot a(t)$. The result for lift, figure 4.4, is nonsensical as expected. The result from drag, however, is promising as it is approximately constant. According to equation (1.14), the value should be V_b which is the area enclosed by the NACA0012 airfoil, 0.0822. The values shown in figure 4.5 unfortunately are larger. This also indicates that maybe formulation 1 [12] should be used or the domain not sufficiently large enough for circulation to be conserved, or something else entirely.

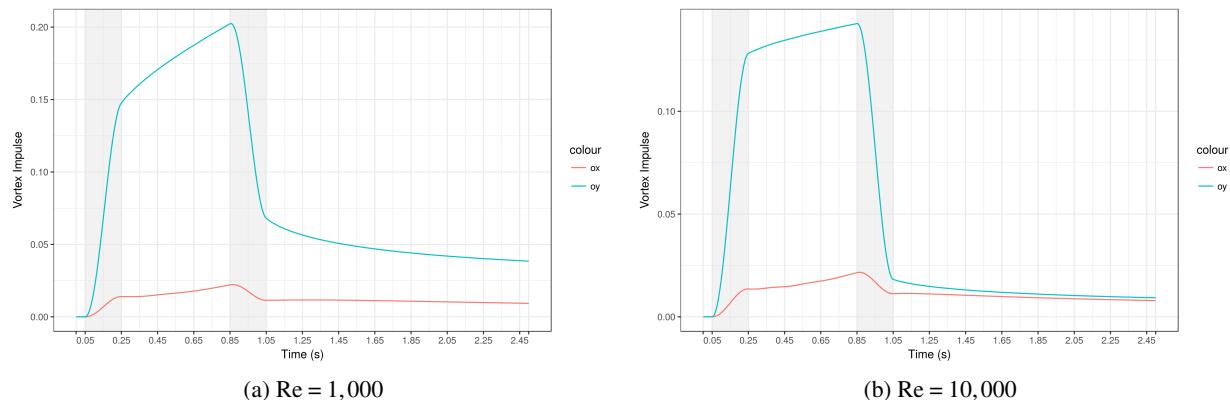


Figure 4.1: Vortex impulse vs time, $0 \leq t \leq 2.5$. ox indicates $-\int \omega x dA$ and oy indicates $\int \omega y dA$.

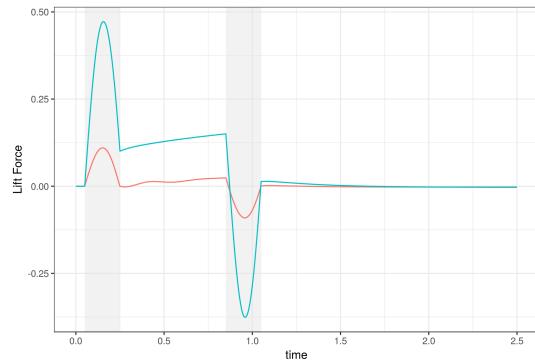
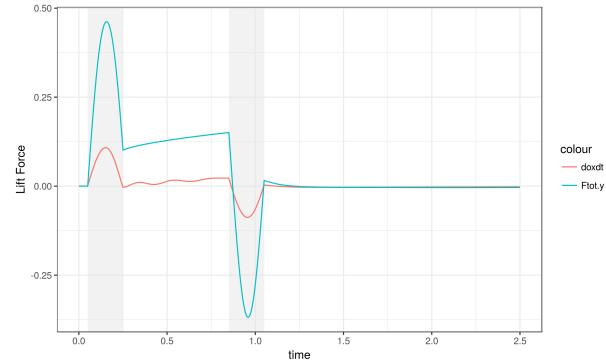
(a) $Re = 1,000$ (b) $Re = 10,000$

Figure 4.2: Lift versus time, $0 \leq t \leq 2.5$. doxdt indicates $\frac{d}{dt} (-\int \omega x dA)$ and Ftot.y is the total of the pressure and viscous force in the y direction from DNS.

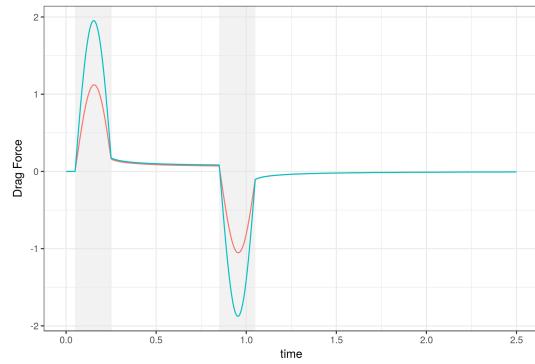
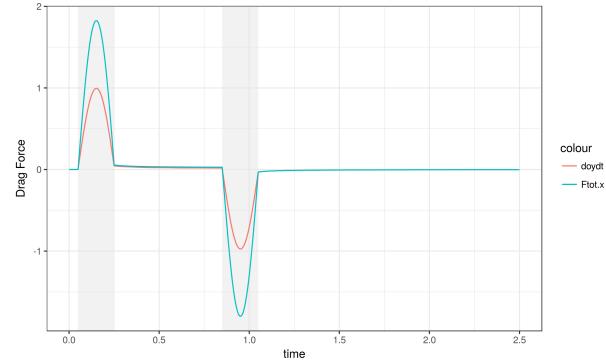
(a) $Re = 1,000$ (b) $Re = 10,000$

Figure 4.3: Drag versus time, $0 \leq t \leq 2.5$. doydt indicates $\frac{d}{dt} (-\int \omega y dA)$ and Ftot.x is the total of the pressure and viscous force in the x direction from DNS.

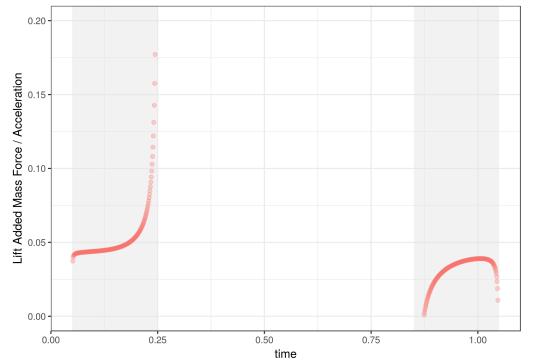
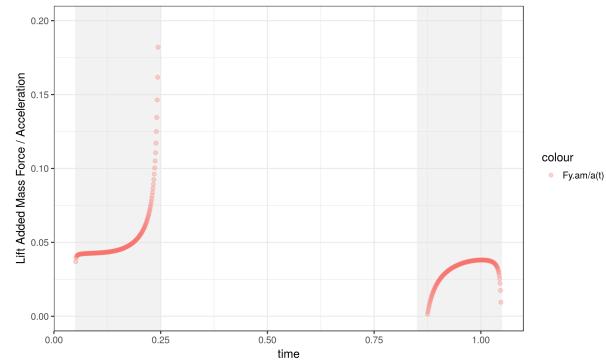
(a) $Re = 1,000$ (b) $Re = 10,000$

Figure 4.4: Added mass lift force coefficient, $0 \leq t \leq 2.5$, $\frac{\text{Ftot.y}-\text{doxdt}}{a(t)}$

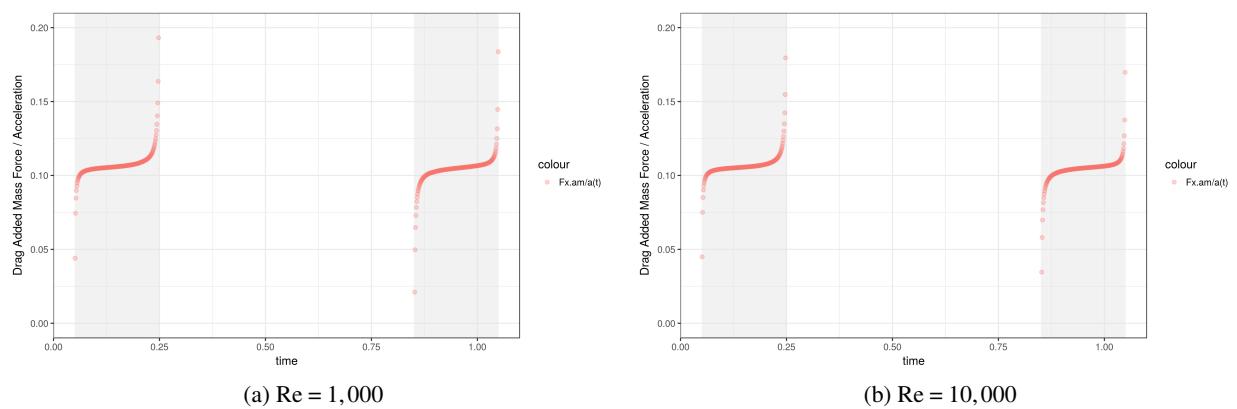


Figure 4.5: Added mass drag force coefficient, $0 \leq t \leq 2.5$, $\frac{\text{Ftot.x-doydt}}{a(t)}$

4.2 Unsteady Vortex Sheet Methods

Thin Airfoil Theory

Figure 1.17 compares the quasi-steady and apparent mass terms of equation (1.17) with the result from DNS ($F_{tot,y}$). Immediately it is clear that apparent mass in Kármán and Sears' formulation [10] is the major contributor to the large lift force. This is somewhat contradictory to the results in figure 4.2, if the vortex impulse method is taken to be true. Limacher, Morton and Wood do show their general formula collapses into the formulation by Kármán and Sears, so [12]

At the end of the stopping phase, the DNS lift is approximately one half of the quasi-steady lift which is in agreement with Saffman's statement that the initial lift is one-half of the final steady-state lift [16]. Even without the wake-effect term, it appears that equation (1.17) well describes the lift force.

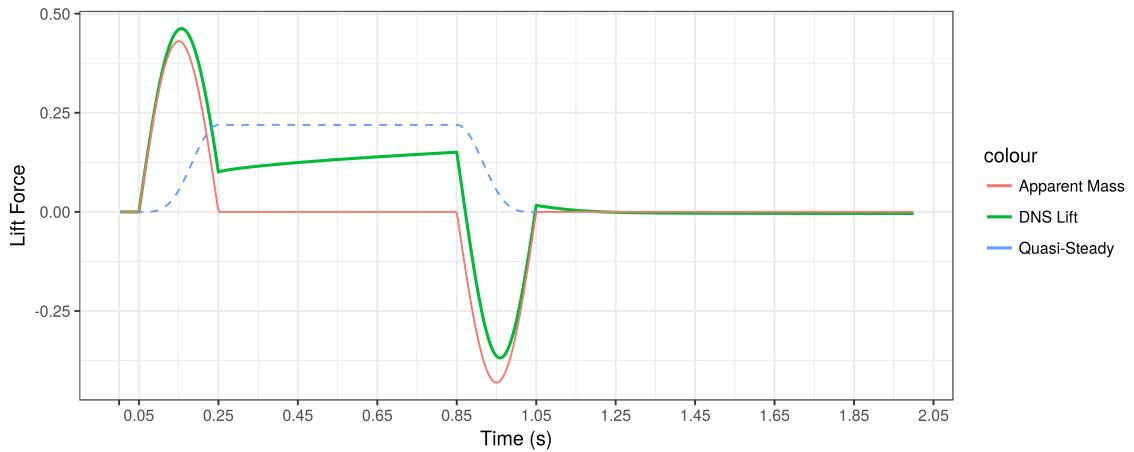


Figure 4.6: Unsteady lift estimation using equation (1.17)

Vortex Panel Method

In order to address the problems with determining the apparent lift, a linearly varying strength vortex panel method was formulated following the classical method outlined in Katz and Plotkin [8] and Kuethe and Chow [9]. To account for the unsteady nature of the problem, the unsteady Bernoulli equation is required to determine pressure forces. This means the velocity potential is needed which standard panel codes do not calculate. Results from Chattot and Hafez [5] is used to ensure the potentials are calculated correctly.

$$p = -\rho \left[\frac{\partial \Phi}{\partial t} + \frac{1}{2} |\nabla \Phi|^2 \right] \quad (4.3)$$

The wake effect problem is tackled in the same manner as Giesing [7] and Wu et al. [21] by allocating a wake panel at the trailing edge and using point vortices. The benchmark for the wake problem is the theoretical work of Wagner [19] and Theodorsen [17].

Unfortunately, due to time restrictions and effort required to test and debug such a program, the unsteady vortex panel method did not progress past an unsteady formulation (presented in the appendix) and validation of code for steady state problems. It is hoped that work will continue in the future and will help resolve the added mass force problems by calculating Φ for formulation 1 of Limacher, Morton and Wood [12] and model the wake roll up ignored by Kármán and Sears' [10].

Part III

Closing Remarks

5 Conclusion

The first object of this project was to accurately calculate the vorticity generation mechanisms and boundary vorticity flux. This was achieved by modifying the source code of Semtex and writing custom scripts to handle the outputs. A single NACA0012 airfoil of unit chord and angle of attack 4 degrees was used at two Reynolds numbers of $Re = 1,000$ and $Re = 10,000$. Convergence of the results was determined by considering the partial derivatives as well as primary variables u , v and p .

A concerted effort to replicate the work of Zhu et al. [24] and Xu [22] was made. The resolution of the grid was increased at the trailing edge, small time steps were used and the polynomial order increased. However, bubbles of alternate signed vorticity were not found during the establishment of the Kutta condition. This problem could be addressed by further increasing the resolution at trailing edge, at the expense of computational time. The boundary vorticity flux, however, was consistent with the findings of Zhu et al.

The lift and drag during the starting and stopping phases was analysed in an attempt to explain their large values. While the drag force was well explained using (no slip) formulation 2 by Limacher, Morton and Wood [12], the lift force was not. The lift force was well explained by applying thin airfoil theory by Kármán and Sears [10] but the result for the added mass force term was not consistent with the vortex impulse calculations in formulation 2.

In an attempt to reconcile these problems, work was started on an unsteady linearly varying strength panel method that could approximate the velocity potential required for formulation 1 by Limacher, Morton and Wood [12] and the wake roll up problem not present in unsteady thin airfoil theory by Kármán and Sears [10]. Unfortunately, this solver was not completed in time as it was not part of the original direction and objectives.

References

- [1] Agromayor, R., Rúa, J. and Kristoffersen, R., Simulation of Starting and Stopping Vortices of an Airfoil, in *Proceedings of the 58th Conference on Simulation and Modelling (SIMS 58) Reykjavik, Iceland, September 25th – 27th, 2017*, Linkoping University Electronic Press, 2017.
- [2] Anderson, J., *Fundamentals of Aerodynamics*, Aeronautical and Aerospace Engineering Series, McGraw-Hill, 2010, 5 edition.
- [3] Blackburn, H. M. and Henderson, R. D., A study of two-dimensional flow past an oscillating cylinder, *Journal of Fluid Mechanics*, **385**, 1999, 255–286.
- [4] Blackburn, H. M. and Sherwin, S. J., Formulation of a Galerkin spectral element–Fourier method for three-dimensional incompressible flows in cylindrical geometries, *Journal of Computational Physics*, **197**, 2004, 759–778.
- [5] Chattot, J.-J. and Hafez, M., *Theoretical and Applied Aerodynamics*, Springer-Verlag GmbH, 2015.
- [6] Fornberg, B., Generation of finite difference formulas on arbitrarily spaced grids, *Mathematics of Computation*, **51**, 1988, 699–699.
- [7] Giesing, J. P., Nonlinear two-dimensional unsteady potential flow with lift, *Journal of Aircraft*, **5**, 1968, 135–143.
- [8] Katz, J. and Plotkin, A., *Low-Speed Aerodynamics (Cambridge Aerospace Series)*, Cambridge University Press, 2001.
- [9] Kuethe, A. M. and Chow, C.-Y., *Foundations of Aerodynamics*, John Wiley & Sons, 1997.
- [10] Kármán, T. V. and Sears, W. R., Airfoil Theory for Non-Uniform Motion, *Journal of the Aeronautical Sciences*, **5**, 1938, 379–390.
- [11] Lighthill, J., *An Informal Introduction to Theoretical Fluid Mechanics*, Oxford University Press, 1986.
- [12] Limacher, E., Morton, C. and Wood, D., Generalized derivation of the added-mass and circulatory forces for viscous flows, *Physical Review Fluids*, **3**.
- [13] Liu, T., Wang, S., Zhang, X. and He, G., Unsteady Thin-Airfoil Theory Revisited: Application of a Simple Lift Formula, *AIAA Journal*, **53**, 2015, 1492–1502.
- [14] Ludwig Prandtl, O. K. G. T., *Applied hydro- and aeromechanics*, New York ; London : McGraw-Hill Book Company, inc, 1934, 1st ed edition, plates: p. [277]-306.
- [15] Morton, B. R., The generation and decay of vorticity, *Geophysical & Astrophysical Fluid Dynamics*, **28**, 1984, 277–308.
- [16] Saffman, P. G., *Vortex Dynamics (Cambridge Monographs on Mechanics)*, Cambridge University Press, 1993.
- [17] Theodorsen, T., General Theory of Aerodynamic Instability and the Mechanism of Flutter, *NACA*, **496**.
- [18] Vincent, M. and Blackburn, H. M., Simulation of starting/stopping vortices for a lifting aerofoil, in *Proceedings of the 19th Australasian Fluid Mechanics Conference (AFMC)*, editors H. Chowdhury and F. Alam, RMIT University, 2014, 1–4, 1–4.

- [19] Wagner, H., Über die Entstehung des dynamischen Auftriebes von Tragflügeln, *ZAMM - Zeitschrift für Angewandte Mathematik und Mechanik*, **5**, 1925, 17–35.
- [20] Willert, C. and Kompenhans, J., PIV Analysis of Ludwig Prandtl's Historic Flow Visualization Films, *arXiv preprint*.
- [21] Wu, F. B., Zeng, N. D., Zhang, L. and Wu, D. M., Numerical solutions for a two-dimensional airfoil undergoing unsteady motion, *Journal of Marine Science and Application*, **3**, 2004, 7–11.
- [22] Xu, L., Numerical study of viscous starting flow past wedges, *Journal of Fluid Mechanics*, **801**, 2016, 150–165.
- [23] Zhou, M.-D., Ma, H.-Y. and Wu, J.-Z., *Vorticity and Vortex Dynamics*, Springer Berlin Heidelberg, 2007.
- [24] Zhu, J. Y., Liu, T. S., Liu, L. Q., Zou, S. F. and Wu, J. Z., Causal mechanisms in airfoil-circulation formation, *Physics of Fluids*, **27**, 2015, 123601.

Part IV

Appendix

A Proofs and Derivations

A.1 Vorticity Generation

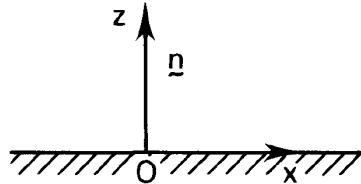


Figure A.1: Coordinate system on the surface of a wall. *Source: Morton 1984, pg 283 [15]*

The Navier-Stokes equation (A.1) can be reduced to the case of a wall moving in its own plane for an infinitesimally small region on the airfoil as described by Morton [15]. The boundary conditions for the Navier-Stokes equation were found by using $\vec{V} = \{V(t), 0\}$ where $V(t)$ is the motion of the body and $z = 0$ to indicate the rigid boundary at the wall. Thus, no slip condition gives $\vec{v} = (V(t), 0) \forall x, t$ for the fluid at the wall and there is no spatial variation of the boundary motion as shown in equation (A.2). Substitution into the Navier-Stokes equation at the boundary yields Equation 5a which can be simplified to equations (A.3b) and (A.3c) (Morton, 1984).

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = -\frac{1}{\rho} \nabla \vec{p} + \frac{1}{Re} \nabla^2 \vec{u} - \vec{a}, \quad \nabla \cdot \vec{u} = 0 \quad (\text{A.1})$$

$$\left[\left(\frac{\partial}{\partial x}, \frac{\partial^2}{\partial x^2} \right) \cdot (u, w) \right]_0 = 0 \quad (\text{A.2})$$

$$\left[\frac{\partial^2}{\partial z^2} (u, w) \right]_0 = \left\{ 0, \frac{1}{\mu} \frac{\partial \vec{p}}{\partial x} + \frac{1}{\nu} \frac{dV(t)}{dt}, \frac{1}{\mu} \right\}_0 \quad (\text{A.3a})$$

$$\nu \left(\frac{\partial \omega}{\partial z} \right)_0 = \left\{ 0, - \left[-\frac{1}{\rho} \frac{\partial \vec{p}}{\partial x} + \frac{dV(t)}{dt} \right], 0 \right\}_0 \quad (\text{A.3b})$$

$$-\nu \vec{n} \cdot \nabla \vec{\omega} = -\vec{n} \times (\nabla \vec{P} + \vec{a}(t)) \quad (\text{A.3c})$$

A.2 Unsteady Thin-Airfoil Theory

Kármán and Sears [10] begin their formulation by considering a vortex pair $\pm\Gamma$ on the x -axis at coordinates x_1 and x_2 . The magnitude of momentum is $\rho\Gamma(x_2-x_1)$ with a line of action at $x=\frac{1}{2}(x_2+x_1)$. The total lift per unit span can be determined as

$$L = -\rho \frac{d}{dt} \sum \Gamma_i x_i \quad (\text{A.4})$$

In order to determine the effect of vorticity being shed into the wake, the conformal mapping is used between a flat plate, $-1 \leq x \leq 1$ in the z plane and a unit circle in the z' plane. The effect of a point vortex of strength $-\Gamma'$ at $x = \xi$ is determined by solving the problem in the z' plane and transforming the solution back into z to give the tangential velocity, v_θ , vorticity distribution over the airfoil, $\gamma(x)$ and total circulation of the airfoil, Γ .

$$v_\theta = \frac{\Gamma'}{2\pi} \left[\frac{\sqrt{\xi^2-1}}{\xi-\cos(\theta)} - \sqrt{\frac{\xi+1}{\xi-1}} \right] = -\frac{\Gamma'}{2\pi} \frac{1-\cos(\theta)}{\xi-\cos(\theta)} \sqrt{\frac{\xi+1}{\xi-1}} \quad (\text{A.5})$$

$$\gamma(x) = \frac{1}{\pi} \frac{\Gamma'}{\xi-x} \sqrt{\frac{1-x}{1+x}} \sqrt{\frac{\xi+1}{\xi-1}} \quad (\text{A.6})$$

$$\Gamma = \int_{-1}^1 \gamma(x) dx = \Gamma' \left(\sqrt{\frac{\xi+1}{\xi-1}} - 1 \right) \quad (\text{A.7})$$

The vorticity bound to the airfoil is denoted by $\gamma(x)$ and that in the wake by $\gamma(\xi)$. The vorticity $\gamma(x)$ is composed of two parts:

- (a) the vorticity, $\gamma_0(x)$, which would be produced, according to thin airfoil theory, by the motion of the airfoil or the given velocity distribution (gust) in the air, if the wake had no effect. $\gamma_0(x)$ is called the “quasi-steady” vorticity distribution;
- (b) the vorticity, $\gamma_1(x)$, which is induced by the wake

A.3 Unsteady Panel Method

Steady Panel Method Derivation

This derivation follows the work of Katz & Plotkin [8] and Kuethe & Chow [9].

The source Φ and induced velocity (u, w) at a point (x, z) for various vortex distributions is shown below.

- Constant Strength Vortex Distribution

In general,

$$\Phi_0 = -\frac{\gamma_0}{2\pi} \int_{x_1}^{x_2} \tan^{-1} \left(\frac{z}{x-x_0} \right) dx_0 = -\frac{\gamma_0}{2\pi} \left[(x-x_1)\theta_1 - (x-x_2)\theta_2 + \frac{z}{2} \ln \left(\frac{r_1^2}{r_2^2} \right) \right] \quad (\text{A.8a})$$

$$u_0 = \frac{\gamma_0}{2\pi} \int_{x_1}^{x_2} \frac{z}{(x-x_0)^2 + z^2} dx_0 = \frac{\gamma_0}{2\pi} (\theta_2 - \theta_1) \quad (\text{A.8b})$$

$$w_0 = -\frac{\gamma_0}{2\pi} \int_{x_1}^{x_2} \frac{x-x_0}{(x-x_0)+z^2} dx_0 = -\frac{\gamma_0}{2\pi} \ln \left(\frac{r_1}{r_2} \right) \quad (\text{A.8c})$$

For the case where (x, z) lies on the element and where x is the midpoint, $x_m = \frac{x_1+x_2}{2}$,

$$\Phi_0(x, \pm 0) = \pm \frac{\gamma_0}{2} (x - x_2) \quad \Phi_0(x_m, \pm 0) = \pm \frac{\gamma_0}{4} (x_1 - x_2) \quad (\text{A.9a})$$

$$u_0(x, \pm 0) = \pm \frac{\gamma_0}{2} \quad u_0(x_m, \pm 0) = \pm \frac{\gamma_0}{2} \quad (\text{A.9b})$$

$$w_0(x, \pm 0) = \frac{\gamma_0}{2\pi} \ln \left(\frac{r_2}{r_1} \right) \quad w_0(x_m, \pm 0) = 0 \quad (\text{A.9c})$$

- Linear Vortex Distribution

In general,

$$\Phi_1 = -\frac{\gamma_1}{2\pi} \int_{x_1}^{x_2} x_0 \tan^{-1} \left(\frac{z}{x-x_0} \right) dx_0 = -\frac{\gamma_1}{2\pi} \left[\frac{xz}{2} \ln \left(\frac{r_1^2}{r_2^2} \right) + \frac{z}{2} (x_1 - x_2) + \frac{x^2 - x_1^2 - z^2}{2} \theta_1 - \frac{x^2 - x_2^2 - z^2}{2} \theta_2 \right] \quad (\text{A.10a})$$

$$u_1 = \frac{\gamma_1}{2\pi} \int_{x_1}^{x_2} \frac{x_0 z}{(x-x_0)^2 + z^2} dx_0 = -\frac{\gamma_1}{2\pi} \left[z \ln \left(\frac{r_1}{r_2} \right) - x(\theta_2 - \theta_1) \right] \quad (\text{A.10b})$$

$$w_1 = -\frac{\gamma_1}{2\pi} \int_{x_1}^{x_2} \frac{x_0 (x-x_0)}{(x-x_0)+z^2} dx_0 = -\frac{\gamma_1}{2\pi} \left[x \ln \left(\frac{r_1}{r_2} \right) + (x_1 - x_2) + z(\theta_2 - \theta_1) \right] \quad (\text{A.10c})$$

For the case where (x, z) lies on the element and where x is the midpoint, $x_m = \frac{x_1+x_2}{2}$,

$$\Phi_1(x, \pm 0) = \pm \frac{\gamma_1}{4} (x^2 - x_2^2) \quad \Phi_1(x_m, \pm 0) = \pm \frac{\gamma_1}{16} (x_1^2 + 2x_1 x_2 - 3x_2^2) \quad (\text{A.11a})$$

$$u_1(x, \pm 0) = \pm \frac{\gamma_1}{2} x \quad u_1(x_m, \pm 0) = \pm \frac{\gamma_1}{4} (x_1 + x_2) \quad (\text{A.11b})$$

$$w_1(x, \pm 0) = \frac{\gamma_1}{2\pi} \left[x \ln \left(\frac{|x-x_1|}{|x-x_2|} \right) + (x_1 - x_2) \right] \quad w_1(x_m, \pm 0) = -\frac{\gamma_1}{2\pi} (x_1 - x_2) \quad (\text{A.11c})$$

For a linearly varying vortex strength panel method, the sum of potential flows should be used for $\gamma = \gamma_0 + \gamma_1(x - x_1)$.

- Panel coordinate system in terms of γ_0 (constant) and γ_1 (linear)

$$\Phi_p = \Phi_0 + \Phi_1 = -\frac{\gamma_0}{2\pi} \left[(x - x_1)\theta_1 - (x - x_2)\theta_2 + \frac{z}{2} \ln \left(\frac{r_1^2}{r_2^2} \right) \right] - \frac{\gamma_1}{2\pi} \left[\frac{xz}{2} \ln \left(\frac{r_1^2}{r_2^2} \right) + \frac{z}{2} (x_1 - x_2) + \frac{x^2 - x_1^2 - z^2}{2} \theta_1 - \frac{x^2 - x_2^2 - z^2}{2} \theta_2 \right] \quad (\text{A.12a})$$

$$u_p = u_0 + u_1 = \frac{\gamma_0}{2\pi} (\theta_2 - \theta_1) - \frac{\gamma_1}{2\pi} \left[z \ln \left(\frac{r_1}{r_2} \right) - x(\theta_2 - \theta_1) \right] \quad (\text{A.12b})$$

$$w_p = w_0 + w_1 = -\frac{\gamma_0}{2\pi} \ln \left(\frac{r_1}{r_2} \right) - \frac{\gamma_1}{2\pi} \left[x \ln \left(\frac{r_1}{r_2} \right) + (x_1 - x_2) + z(\theta_2 - \theta_1) \right] \quad (\text{A.12c})$$

- Panel coordinate system in terms of γ_a and γ_b where

$$\gamma_a = \gamma_0 \quad \text{and} \quad \gamma_b = \gamma_0 + \gamma_1(x_2 - x_1) \quad (\text{A.13a})$$

$$\gamma_0 = \gamma_a \quad \text{and} \quad \gamma_1 = \frac{\gamma_b - \gamma_a}{x_2 - x_1} \quad (\text{A.13b})$$

Rewriting in term of γ_a and γ_b yields

$$\Phi_p = -\frac{\gamma_a}{2\pi} \left[(x - x_1)\theta_1 - (x - x_2)\theta_2 + \frac{z}{2} \ln \left(\frac{r_1^2}{r_2^2} \right) \right] - \frac{\gamma_b - \gamma_a}{2\pi(x_2 - x_1)} \left[\frac{xz}{2} \ln \left(\frac{r_1^2}{r_2^2} \right) + \frac{z}{2} (x_1 - x_2) + \frac{x^2 - x_1^2 - z^2}{2} \theta_1 - \frac{x^2 - x_2^2 - z^2}{2} \theta_2 \right] \quad (\text{A.14a})$$

$$u_p = \frac{\gamma_a}{2\pi} (\theta_2 - \theta_1) - \frac{\gamma_b - \gamma_a}{2\pi(x_2 - x_1)} \left[z \ln \left(\frac{r_1}{r_2} \right) - x(\theta_2 - \theta_1) \right] \quad (\text{A.14b})$$

$$w_p = -\frac{\gamma_a}{2\pi} \ln \left(\frac{r_1}{r_2} \right) - \frac{\gamma_b - \gamma_a}{2\pi(x_2 - x_1)} \left[x \ln \left(\frac{r_1}{r_2} \right) + (x_1 - x_2) + z(\theta_2 - \theta_1) \right] \quad (\text{A.14c})$$

- Each of these expressions can now be written as functions of γ_a and γ_b

$$\Phi_p(\gamma_a) = -\frac{z}{4\pi} \left[\ln \left(\frac{r_1^2}{r_2^2} \right) \left(1 - \frac{x}{x_2 - x_1} \right) + 1 \right] - \frac{1}{2\pi} \left[\theta_1 \left((x - x_1) - \frac{x^2 - x_1^2 - z^2}{2(x_2 - x_1)} \right) - \theta_2 \left((x - x_2) - \frac{x^2 - x_2^2 - z^2}{2(x_2 - x_1)} \right) \right] \quad (\text{A.15a})$$

$$\Phi_p(\gamma_b) = -\frac{z}{4\pi} \left[\ln \left(\frac{r_1^2}{r_2^2} \right) \frac{x}{x_2 - x_1} - 1 \right] - \frac{1}{2\pi} \left[\theta_1 \left(\frac{x^2 - x_1^2 - z^2}{2(x_2 - x_1)} \right) - \theta_2 \left(\frac{x^2 - x_2^2 - z^2}{2(x_2 - x_1)} \right) \right] \quad (\text{A.15b})$$

$$u_p(\gamma_a) = \frac{1}{2\pi} \left[(\theta_2 - \theta_1) + \frac{z}{x_2 - x_1} \ln \left(\frac{r_1}{r_2} \right) - \frac{x}{x_2 - x_1} (\theta_2 - \theta_1) \right] \quad (\text{A.16a})$$

$$u_p(\gamma_b) = -\frac{1}{2\pi} \left[\frac{z}{x_2 - x_1} \ln \left(\frac{r_1}{r_2} \right) - \frac{x}{x_2 - x_1} (\theta_2 - \theta_1) \right] \quad (\text{A.16b})$$

$$w_p(\gamma_a) = -\frac{1}{2\pi} \left[\left(1 - \frac{x}{x_2 - x_1} \right) \ln \left(\frac{r_2}{r_1} \right) + 1 - \frac{z}{x_2 - x_1} (\theta_2 - \theta_1) \right] \quad (\text{A.17a})$$

$$w_p(\gamma_b) = -\frac{1}{2\pi} \left[\frac{x}{x_2 - x_1} \ln \left(\frac{r_1}{r_2} \right) - 1 + \frac{z}{x_2 - x_1} (\theta_2 - \theta_1) \right] \quad (\text{A.17b})$$

- In the local panel coordinates, define $(x_1, y_1) = (0, 0)$.
i.e. use the transformation

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \right) \quad (\text{A.18})$$

Thus, the equations can be significantly simplified to

$$\Phi_p(\gamma_a) = -\frac{z}{4\pi x_2} \left[(x_2 - x) \ln \left(\frac{r_1^2}{r_2^2} \right) + x_2 \right] - \frac{1}{4\pi x_2} \left[(-x^2 + 2xx_2 + z^2)(\theta_1 - \theta_2) + x_2^2 \theta_2 \right] \quad (\text{A.19a})$$

$$\Phi_p(\gamma_b) = -\frac{z}{4\pi x_2} \left[x \ln \left(\frac{r_1^2}{r_2^2} \right) - x_2 \right] - \frac{1}{4\pi x_2} \left[(x^2 - z^2)(\theta_1 - \theta_2) + x_2^2 \theta_2 \right] \quad (\text{A.19b})$$

$$u_p(\gamma_a) = \frac{1}{2\pi x_2} \left[-z \ln \left(\frac{r_2}{r_1} \right) + (x_2 - x)(\theta_2 - \theta_1) \right] \quad (\text{A.20a})$$

$$u_p(\gamma_b) = \frac{1}{2\pi x_2} \left[z \ln \left(\frac{r_2}{r_1} \right) + x(\theta_2 - \theta_1) \right] \quad (\text{A.20b})$$

$$w_p(\gamma_a) = \frac{1}{2\pi x_2} \left[(x_2 - x) \ln \left(\frac{r_2}{r_1} \right) - x_2 + z(\theta_2 - \theta_1) \right] \quad (\text{A.21a})$$

$$w_p(\gamma_b) = \frac{1}{2\pi x_2} \left[x \ln \left(\frac{r_2}{r_1} \right) + x_2 - z(\theta_2 - \theta_1) \right] \quad (\text{A.21b})$$

Steady Panel Method Validation Shed Vortices

- Two-Dimensional Point Vortex

$$\Phi_{\bullet} = -\frac{\Gamma_{\bullet}}{2\pi} \tan^{-1} \left(\frac{z - z_0}{x - x_0} \right) \quad (\text{A.22a})$$

$$u_{\bullet} = \frac{\Gamma_{\bullet}}{2\pi} \frac{z - z_0}{(x - x_0)^2 + (z - z_0)^2} \quad (\text{A.22b})$$

$$w_{\bullet} = -\frac{\Gamma_{\bullet}}{2\pi} \frac{x - x_0}{(x - x_0)^2 + (z - z_0)^2} \quad (\text{A.22c})$$

Unsteady Panel Method Formulation

1. Constants for all time steps following the formulation in Katz and Plotkin [8].
 - (a) Determine panel collocation points $(x, y)_i$, unit normal vectors \vec{n}_i , unit tangent \vec{t}_i and panel lengths c_i .
 - (b) Determine the influence coefficients K_{ij} using the standard method for a linearly varying strength vortex sheet panel method.

$$K_{i,j} = (u, w)_{i,j} \cdot \vec{n}_i = [(u_b, w_b)_{i,j-1} + (u_a, w_a)_{i,j}] \cdot \vec{n}_i, \quad j \neq 1, N+1 \quad (\text{A.23a})$$

$$K_{i,1} = (u_a, w_a)_{i,1} \cdot \vec{n}_i \quad \text{and} \quad K_{i,N+1} = (u_b, w_b)_{i,N+1} \cdot \vec{n}_i \quad (\text{A.23b})$$

where $(u_a, w_a) = (u_p(\gamma_a), w_p(\gamma_a))$ and $(u_b, w_b) = (u_p(\gamma_b), w_p(\gamma_b))$ as defined in (A.20a), (A.21a), (A.20b) and (A.21b)

- There are $N+1$ unknowns, $\gamma_1, \gamma_2, \dots, \gamma_{N+1}$, and N equations for $K_{i,j}$ panels $1, 2, \dots, N$.
- 2. At the start of time $t = t_k$, denoted by $t = t_k^+$, the strength of the point vortices $\Gamma_1, \Gamma_2, \dots, \Gamma_W$ are known from the previous time steps $t = 1, 2, \dots, k-1$ since $k = W$. A constant strength vortex panel, γ_c , is introduced. The length of the panel, r_w , orientation, θ_w , and the influence coefficients $L_{i,k}$ must be determined.

The formulation by Wu et al. [21] and Giesing [7] is followed.

- (a) The strength of the panel γ_c is determined by considering Wu et al. [21] to arrive at $\gamma_1 + \gamma_{N+1} = \gamma_c$. Kelvin's theorem is used to get to $\sum (C_j \gamma_j) + \sum (\Gamma_k) + C_w \gamma_c = 0$. This is shown in matrix form below, however it is clear that the solution depends on the C_w .

$$\begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 & -1 \\ C_1 & \cdots & C_{N+1} & 1 & \cdots & 1 & C_W \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_{N+1} \\ \Gamma_1 \\ \vdots \\ \Gamma_W \\ \gamma_c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (\text{A.24})$$

$$C_j = \frac{1}{2} (c_{j-1} + c_j), \quad j \neq 1, N+1 \quad (\text{A.25a})$$

$$C_1 = \frac{1}{2} c_1 \quad \text{and} \quad C_{N+1} = \frac{1}{2} c_{N+1} \quad (\text{A.25b})$$

- (b) To lie on the trailing edge streamline, the panel should be parallel to the flow coming off the trailing edge and with a distance, C_w , equal to $\sqrt{u^2 + v^2} \times (t_{k+1} - t_k)$

$$\begin{bmatrix} C_w \cos(\theta_w) \\ C_w \sin(\theta_w) \end{bmatrix} = \begin{bmatrix} u \cdot (t_{k+1} - t_k) \\ v \cdot (t_{k+1} - t_k) \end{bmatrix}$$

$$= \begin{bmatrix} K_{u,1} & \cdots & K_{u,N+1} & L_{u,1} & \cdots & L_{u,W} & M_u \\ K_{w,1} & \cdots & K_{w,N+1} & L_{w,1} & \cdots & L_{w,W} & M_w \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_{N+1} \\ \Gamma_1 \\ \vdots \\ \Gamma_W \\ \gamma_c \end{bmatrix} \times (t_{k+1} - t_k) \quad (\text{A.26})$$

$$K_{u,j} = (u_b)_{i,j-1} + (u_a)_{i,j}, \quad j \neq 1, N+1 \quad (\text{A.27a})$$

$$K_{u,1} = (u_a)_{i,1} \quad \text{and} \quad K_{u,N+1} = (u_b)_{i,N+1} \quad (\text{A.27b})$$

$$K_{w,j} = (w_b)_{i,j-1} + (w_a)_{i,j}, \quad j \neq 1, N+1 \quad (\text{A.28a})$$

$$K_{w,1} = (w_a)_{i,1} \quad \text{and} \quad K_{w,N+1} = (w_b)_{i,N+1} \quad (\text{A.28b})$$

It is immediately clear that $L_{u,k}, L_{w,k}, M_u, M_w$ depend on the location of $\Gamma_1, \Gamma_2, \dots, \Gamma_W$ and γ_c which have not been determined yet. Thus, an iterative process is required.

- (c) The movement of the vortices can be found as by considering the induced velocity at the location of the
- For the shed panel, there are 3 unknowns γ_c, C_w, θ_w and 4 equations
 - For the location of the point vortices, there are $2W$ unknowns and $2W$ equations
 - Thus, the complete formulation is

$$\begin{bmatrix} K_{1,1} & K_{1,2} & \cdots & K_{1,N} & K_{1,N+1} & L_{1,1} & \cdots & L_{1,W} & M_1 \\ K_{2,1} & K_{2,2} & \cdots & K_{2,N} & K_{2,N+1} & L_{2,1} & \cdots & L_{2,W} & M_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ K_{N,1} & K_{N,2} & \cdots & K_{N,N} & K_{N,N+1} & L_{N,1} & \cdots & L_{N,W} & M_N \\ 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & -1 \\ C_1 & C_2 & \cdots & C_N & C_{N+1} & 1 & \cdots & 1 & C_W \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_N \\ \gamma_{N+1} \\ \Gamma_1 \\ \vdots \\ \Gamma_W \\ \gamma_c \end{bmatrix} = \begin{bmatrix} -\vec{u} \cdot \vec{n}_1 \\ -\vec{u} \cdot \vec{n}_2 \\ \vdots \\ -\vec{u} \cdot \vec{n}_N \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (\text{A.29})$$

$$M_i = (u, w)_i \cdot \vec{n}_i$$

3. At end of the period $(t_{k+1} - t_k)$, denoted by t_{k+1}^- , the panel sheet is replaced by a point vortex. This is to reduce the computation time requirement and avoid the need to calculate the change in vorticity due to stretching and compressing of elements.

B Post-Processing Scripts

B.1 Semtex Files

Add Vorticity Fields

Four additional fields were required in order to determine the values of $\omega, \frac{\partial\omega}{\partial x}, \frac{\partial\omega}{\partial y}, \frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}$. This was achieved by adding five additional AuxField.

Listing 1: Exerts of modifications to addfield.cpp

```

1 // Add new flag case for command line
2 case 'G': flag[VORTGEN]      = true; break;
3 // Declare VortGen
4 vector<AuxField*>          VortGen;
5 vector<real_t*>             VortGenData;
6 // Handle flags from command line
7 if (need[VORTGEN]) {
8     VortGen.resize (5);
9     VortGenData.resize (5);
10    for (i = 0; i < 5; i++) {
11        VortGenData[i] = new real_t [allocSize];
12        VortGen[i]     = new AuxField (VortGenData[i],nz,elmt,'k'+i);
13        addField[iAdd++] = VortGen[i];
14    }
15 }
16 // while (getDump(D, file))
17 // for(i = 0; i < allocSize; i++)
18 //     if (need[VORTGEN]) { // Only for 2 dimensions
19 //         tensor3::vorticity (tensor, vort);
20 //         VortGenData[4][i] = vort[2];           // o = omega
21 //     }
22 // finish looping over each point i
23 if (need[VORTGEN]) { // Only for 2 dimensions
24     (*VortGen[0] = *VortGen[4]).gradient(0); // k = do/dx
25     (*VortGen[1] = *VortGen[4]).gradient(1); // l = do/dy
26     (*VortGen[2] = *pressure) .gradient(0); // m = dp/dx
27     (*VortGen[3] = *pressure) .gradient(1); // n = dp/dy
28 }
29 // finish getDump

```

Wall Gradients

Listing 2: Exerts of modifications to wallmesh.cpp

```

1 // static void printwalls (...) 
2   std::cout.precision(16);
3   cout << \
4     setw(24) << "x" << setw(24) << "y" << \
5     setw(24) << "nx" << setw(24) << "ny" << setw(24) << "area" << \
6     setw(10) << "k" << setw(10) << "s" << endl;
7   for (i = 0; i < Nedge; i++) {
8     if (strstr (bman -> groupInfo (edge[i] -> group), "wall")) {
9       k = edge[i] -> elmt;
10      s = edge[i] -> side;
11      elmt[k] -> sideGeom(s, &xs[0], &ys[0], &nxs[0], &nys[0], &area[0]);
12      for (j = 0; j < np; j++) {
13        cout << \
14          setw(24) << xs[j] << setw(24) << ys[j] << \
15          setw(24) << nxs[j] << setw(24) << nys[j] << \
16          setw(24) << area[j] << \
17          setw(10) << k + 1 << setw(10) << s + 1 << endl;
18    }
19  }

```

Session File

Listing 3: Header of session file for Re = 10,000

```

1 <USER>
2   u = 0.0
3   v = 0.0
4   p = 0.0
5 </USER>
6
7 <FIELDS>
8   u v p
9 </FIELDS>
10
11 <TOKENS>
12   KINVIS = 1./10000.
13
14   D_T = 0.0001
15   N_STEP = 25000
16   N_TIME = 2
17
18   N_P = 10
19   N_Z = 1
20   LZ = 1.0
21   BETA = TWOPi/LZ
22
23   IO_CFL = 50
24   IO_FLD = 10
25   IO_HIS = 1
26
27   AVERAGE = 0
28   CHKPOINT = 0
29   ITERATIVE = 1
30   TBCS = 1
31 </TOKENS>

```

```

32
33 <FORCE>
34   MOD_A_X = 1
35   MOD_A_Y = 0
36   MOD_ALPHA_X = -((5*PI/2*sin(5*PI*(t-0.05)))*heav(t-0.05)+(-5*PI/2*sin(5*
37     PI*(t-0.05)))*heav(t-0.25)+(-5*PI/2*sin(5*PI*(t-0.05)))*heav(t-0.85)
38     +(5*PI/2*sin(5*PI*(t-0.05)))*heav(t-1.05))
39   MOD_ALPHA_Y = 0
40 </FORCE>
41
42 <GROUPS NUMBER=5>
43   1   o    outflow
44   2   p    wall
45   3   w    group_name
46   4   s    group_name
47   5   v    inflow
48 </GROUPS>
49
50 <BCS NUMBER=5>
51   1   o    3
52     <N>      u = 0      </N>
53     <N>      v = 0      </N>
54     <D>      p = 0      </D>
55   2   p    3
56     <D>      u = 0      </D>
57     <D>      v = 0      </D>
58     <H>      p = 0      </H>
59   3   w    3
60     <N>      u = 0      </N>
61     <D>      v = 0      </D>
62     <H>      p = 0      </H>
63   4   s    3
64     <N>      u = 0      </N>
65     <D>      v = 0      </D>
66     <H>      p = 0      </H>
67   5   v    3
68     <D>      u = (1/2-1/2*cos(5*PI*(t-0.05)))*heav(t-0.05)
69       +(1/2-1/2*cos(5*PI*(t-0.05)+PI))*heav(t-0.25)
       +(-1/2-1/2*cos(5*PI*(t-0.05)+PI))*heav(t-0.85)
       +(-1/2-1/2*cos(5*PI*(t-0.05)))*heav(t-1.05)  </D>
       <D>      v = 0      </D>
       <H>      p = 0      </H>
70 </BCS>

```

B.2 Bash Scripts

Convergence Study

This script takes all *.sesh files and copies them into a /results folder. Semtex is called to perform DNS and Semtex utilities used to generate useful output. Important outputs include:

- addvortfield to create $\omega, \frac{\partial \omega}{\partial x}, \frac{\partial \omega}{\partial y}, \frac{\partial p}{\partial x}, \frac{\partial p}{\partial y}$
- wallgrad to generate the wall surface and normals
- convert and csplit to output separate ASCII text files of the field dump for each time step

Listing 4: Bash script to perform DNS for convergence study

```

1 # Create required folders (if necessary)
2 mkdir -p results
3 mkdir -p images
4 # Create array of N_P to loop over
5 N_P=( 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 )
6 # Run dns on each N_P case
7 for i in ${N_P[*]}; do
8     # Set N_P in session file and move to new unique folder
9     name=$(printf "%02d" $i)
10    echo N_P = $name
11    mkdir -p results/NACA0012-N_P$name
12    sed "s/N_P = X/N_P      = $i/g" \
13        naca0012 >      results/NACA0012-N_P$name/naca0012-N_P$name.sesh
14    cp bndry_prf.dat results/NACA0012-N_P$name/bndry_prf.dat
15    cd results/NACA0012-N_P$name
16    # Run dns on session file
17    for sessionfile in *.sesh; do
18        # Move setup files into unique folder
19        f="${sessionfile%.*}"                                &&
20        cp $sessionfile $f                                    &&
21        # Run semtex
22        meshpr      $f > $f.msh                         &&
23        meshpr -i $f > $f.mshi                         &&
24        massmat     $f > $f.mass                         &&
25        enumerate   $f > $f.num                          &&
26        dns $f | grep "Divergence Energy:"             &&
27        # Run semtex utilities
28        addvortfield -G -s $f $f.fld > $f.Gfld    &&
29        convert     $f.Gfld > $f.flldump           &&
30        wallgrad   $f $f.msh > $f.wallgrad         &&
31        # Create ASCII dump for post processing
32        csplit -z "$f.flldump" /Session/ '{*'}' >/dev/null &&
33        for i in [xx]*; do
34            mv $i "$f-${i##*xx}.dump"; done                &&
35        echo $sessionfile complete
36        # Finish
37        rm $f.msh $f.num $f.fld $f.Gfld $f.mdl $f.flldump $f.his
38    done
39    cd ../..
40 done

```

Element-by-element Integration

This script takes all *.sesh files and copies them into a /results folder. Semtex is called to perform DNS and Semtex utilities used to generate useful output. Important outputs include:

- massmat is used to create the system mass matrix for all points (element-by-element)
- integral has been modified to output the integration per element

Listing 5: Bash script to integrate element-by-element across the domain

```

1 # Create required folders (if necessary)
2 mkdir -p results
3 mkdir -p images
4 # Run dns on each session file
5 for sessionfile in *.sesh; do
6     # Move setup files into results
7     f="${sessionfile%.*}"          &&
8     cp $sessionfile results/$sessionfile  &&
9     cp bndry_prf.dat results/bndry_prf.dat  &&
10    cd results                      &&
11    cp $sessionfile $f                &&
12    # Run semtex
13    meshpr   $f > $f.msh           &&
14    meshpr -i $f > $f.mshi          &&
15    massmat   $f > $f.mass          &&
16    enumerate $f > $f.num           &&
17    dns $f | grep "Divergence Energy:"  &&
18    # Run semtex utilities
19    addvortfield -G -s $f $f.fld > $f.Gfld  &&
20    convert $f.Gfld   > $f.flldump  &&
21    compare $f $f.fld > /dev/null  &&
22    wallmesh $f $f.msh > $f.wallmsh  &&
23    wallgrad $f $f.msh > $f.wallgrad  &&
24    # Create ASCII dump for post processing
25    csplit -z "$f.flldump" /Session/ '{*' } >/dev/null  &&
26    for i in [xx]*; do
27        mv $i "$f-${i##*xx}.dump"; done          &&
28    # Create integral ASCII dump to compare to massmat
29    integral $f $f.Gfld > $f.integral      &&
30    csplit -z "$f.integral" /timestep/ '{*' } >/dev/null  &&
31    for i in [xx]*; do
32        mv $i "$f-${i##*xx}.integral"; done      &&
33    # Finish
34    echo "$sessionfile complete"
35    cd ..
36 done

```

Generating Tecplot Output

This script takes all *.sesh files and copies them into a /results folder. Semtex is called to perform DNS and Semtex utilities used to generate useful output. Tecplot *plt outputs are placed in a /tecplot folder. Important outputs include:

- time=\$(...) reads in the time from the ASCII dump file
- sem2tec converts the ASCII dump to a (non-interpolated) Tecplot *dat format
- preplot converts the *dat file to a binary *plt file

Note that preplot packaged with semtex came from 1993 and does not have the SOLUTIONTIME token. Thus, the preplot program packaged with Tecplot (in /bin) was used instead.

Listing 6: Bash script to generate Tecplot files for each time step

```

1 # Create required folders (if necessary)
2 mkdir -p results
3 mkdir -p images
4 # Run dns on each session file
5 for sessionfile in *.sesh; do
6     # Move setup files into results
7     f="${sessionfile%.*}"          &&
8     cp $sessionfile results/$sessionfile  &&
9     cp bndry_prf.dat results/bndry_prf.dat  &&
10    cd results                      &&
11    cp $sessionfile $f                &&
12    # Run semtex
13    meshpr   $f > $f.msh           &&
14    meshpr -i $f > $f.mshi          &&
15    massmat   $f > $f.mass          &&
16    enumerate $f > $f.num           &&
17    dns $f | grep "Divergence Energy:"  &&
18    # Run semtex utilities
19    addvortfield -G -s $f $f.fld > $f.Gfld  &&
20    convert $f.Gfld > $f.flldump      &&
21    compare $f $f.fld > /dev/null      &&
22    wallmesh $f $f.msh > $f.wallmsh    &&
23    wallgrad $f $f.msh > $f.wallgrad    &&
24    # Create ASCII dump for post processing
25    csplit -z "$f.flldump" /Session/ '{*' } >/dev/null  &&
26    for i in [xx]*; do
27        dump="$f-${i#xx}"          &&
28        mv $i $dump.dump          &&
29        # Add SOLUTIONTIME and call preplot (tecplot360 2018 R1)
30        time=$(sed -n '5p' $dump.dump | grep -Eo "[0-9]+\.[0-9]+")  &&
31        echo $time                &&
32        sem2tec -n 0 -o $dump.dat -m $f.msh $dump.dump          &&
33        sed -i "/POINT/s/$/, SOLUTIONTIME=$time/" $dump.dat      &&
34        preplot $dump.dat > /dev/null          &&
35        rm $dump.dat            &&
36        mv $dump.plt ..../tecplot/$dump.plt      &&
37        done                     &&
38        echo $sessionfile complete
39        cd ..
40    done

```

B.3 R Scripts

Some selected R scripts are presented below to give an overview of the post processing methodology.

Listing 7: R script to batch process session and dump files

```

1 ######
2 # Analysis of Whole Time Domain
3 # Alwin Wang
4 -----
5
6 ---- Set Up
7 # Use rstudioapi to get saved location of this file
8 setwd(dirname(rstudioapi::getActiveDocumentContext()$path))
9 ---- * Scripts
10 # Source Required Scripts
11 srcpath = ".../R/"
12 source(paste0(srcpath, "src_library-manager.R"))
13 source(paste0(srcpath, "src_batch-functions.R"))
14 source(paste0(srcpath, "src_helper-functions.R"))
15
16 ---- Batch Process
17 # Batch list to process
18 batchfolder = "results"
19 df_batch <- ListSesh(batchfolder, c("airfoil"))
20 # List of unique airfoils & N_P
21 li_airfoil <- split (df_batch, df_batch$airfoil)
22 li_airfoil <- lapply(li_airfoil, function(df) df[1,])
23 outp_airfoil <- pblapply(li_airfoil, BatchLoadAirfoil, srcpath)
24 # List of meshes
25 tokenwords = list("N_P")
26 df_mesh <- df_batch %>%
27   rowwise() %>%
28   do(data.frame(
29     ., LoadSeshTokenWords(.seshpath, tokenwords),
30     stringsAsFactors = FALSE)) %>%
31   filter(!is.na(tokenvalue)) %>%
32   mutate(
33     ID = paste0(airfoil))
34 li_mesh <- split (df_mesh, df_mesh$ID)
35 li_mesh <- lapply(li_mesh, function(x) x[1,])
36 outp_mesh <- pblapply(li_mesh, BatchLoadMesh, outp_airfoil, srcpath)
37 # List of dumps
38 df_dump <- df_mesh %>%
39   rowwise() %>%
40   do(data.frame(., dumpfile = ListDump(.folder, .seshname),
41     stringsAsFactors = FALSE)) %>%
42   mutate(dumpppath = paste0(folder, dumpfile))
43 li_dump <- split(df_dump, df_dump$dumppath)
44 # Use cluster
45 cl <- makeCluster(detectCores() - 1)
46 outp_dump <- pblapply(li_dump, BatchLoadDump, outp_mesh,
47   plot = c("none"),
48   outp = "wall", srcpath,
49   cl = cl)
50 stopCluster(cl)
51 save(outp_dump, file="outp_dump.RData")
52
53 ---- Batch Calc Output

```

```

54 outp <- list()
55 # Wall
56 outp$wall <- lapply(outp_dump, function(dump) cbind(
57   dump$wall,
58   select(dump$data_plot, airfoil, seshname, tokenvalue, ID, time, kinvis, a) ))
59 outp$wall <- bind_rows(outp$wall)
60 # BVFa
61 outp$BVFa <- lapply(outp_dump, function(dump) cbind(
62   dump$bvfa,
63   select(dump$data_plot, airfoil, seshname, time, kinvis, a) ))
64 outp$BVFa <- bind_rows(outp$BVFa)
65 # Circulation
66 outp$circ <- lapply(outp_dump, function(dump) cbind(
67   rbind(select(dump$inte, regn, o),
68         data.frame(regn = "W", o = sum(dump$inte[c(1,3,4),2]))),
69   select(dump$data_plot, airfoil, seshname, time, kinvis, a) ))
70 outp$circ <- bind_rows(outp$circ)
71 # Vortex Impulse
72 outp$impl <- lapply(outp_dump, function(dump) cbind(
73   select(dump$inte, regn, ox, oy),
74   select(dump$data_plot, airfoil, seshname, time, kinvis, a) ))
75 outp$impl <- bind_rows(outp$impl)
76 outp$impl <- outp$impl %>%
77   arrange(time)
78 # Vortex Force
79 outp$forc <- outp$impl %>%
80   filter(regn == "Total") %>%
81   mutate(doxdt = (lead(ox)-ox)/(lead(time)-time),
82         doydt = (lead(oy)-oy)/(lead(time)-time) )
83 # Lift and Drag Force
84 outp$LD <- read.table(
85   paste0(df_batch$seshpath, ".flx"), skip = 3 )
86 colnames(outp$LD) <- c(
87   "step", "time",
88   "Fpre.x", "Fvis.x", "Ftot.x",
89   "Fpre.y", "Fvis.y", "Ftot.y",
90   "Fpre.z", "Fvis.z", "Ftot.z" )

```

Listing 8: R script calculations of the session, mesh and batch files

```

1 ######
2 # Batch Functions
3 # Alwin Wang
4 -----
5
6 #### Airfoil Calculation
7 # Determine things like spline distance once per unique airfoil
8 BatchLoadAirfoil <- function(data_airfoil, srcpath = "") {
9   source(paste0(srcpath, "src_library-manager.R"))
10  source(paste0(srcpath, "src_helper-functions.R"))
11  #--- * Boundary Data
12  # Not sure this is actually used anywhere...
13  bndry <- LoadBndry(data_airfoil$folder)
14  #--- * Wall Mesh Data
15  wallmesh <- LoadWallGrad(data_airfoil$seshpath)
16  long_wall <- AirfoilLongWall(wallmesh)
17  long_wall <- AirfoilSpline(long_wall)
18  long_wall <- AirfoilNorm (long_wall)
19  #--- > Airfoil Calc Output
20  list_airfoil <- list(
21    airfoil = data_airfoil,
22    bndry = bndry,
23    long_wall = long_wall)
24  rm(data_airfoil, bndry, long_wall, wallmesh)
25  # Output
26  return(list_airfoil)
27 }
28
29 #### Session and Mesh Calculation
30 BatchLoadMesh <- function(data_mesh, outp_airfoil, srcpath = "") {
31   source(paste0(srcpath, "src_library-manager.R"))
32   source(paste0(srcpath, "src_helper-functions.R"))
33   list_airfoil <- outp_airfoil[[data_mesh$airfoil]]
34   #--- * Airfoil Data
35   long <- list()
36   #--- * Session Data
37   session <- LoadSeshFileKeywords(data_mesh$seshpath)
38   long$sesh <- LongSesh(session)
39   #--- * Mesh Data
40   long$mesh <- LoadMesh(data_mesh$seshpath)
41   long$mesh <- LongMesh(long$mesh, long$sesh)
42   #--- * Mass Data
43   long$mass <- LoadMass(data_mesh$seshpath)
44   long$mesh <- LongMass(long$mesh, long$mass)
45   #--- * Wall Data
46   long$wall <- list_airfoil$long_wall
47   long$wall <- LongWall(long$wall, long$mesh)
48   #--- * Local Data
49   long$mesh <- LocalMesh(long$mesh, long$wall)
50   long$enum <- LocalEnum(long$mesh, long$wall)
51   #--- > Sesh & Mesh Calc Output
52   list_mesh <- list(
53     wall = long$wall,
54     mesh = long$mesh,
55     enum = long$enum)
56   rm(data_mesh, long, session)

```

```

57  # output
58  return(list_mesh)
59 }
60
61 ----- Dump File Calculation -----
62 BatchLoadDump <- function(data_dump, outp_mesh, plot = "none",
63                           outp = "wall", srcpath = "", addscr = NULL) {
64   source(paste0(srcpath, "src_library-manager.R"))
65   source(paste0(srcpath, "src_helper-functions.R"))
66   source(paste0(srcpath, "src_plotting-functions.R"))
67   list_mesh <- outp_mesh[[data_dump$ID]]
68
69   ----- * Dump Data -----
70   dump      <- LoadGradFieldDump(data_dump$folder, data_dump$dumpfile)
71   dump$dump <- DumpMesh(list_mesh$mesh, dump$dump)
72   dump$wall <- DumpWall(list_mesh$wall, dump$dump)
73
74   ----- * Acceleration Data -----
75   # Note that tangent direction based on spline calc
76   LoadSeshBCEq(data_dump$seshpath, "MOD_ALPHA_X")
77   dump$a     <- BC_mod_alpha_x(dump$time)
78   dump$wall <- DumpAccel(dump$a, dump$wall)
79
80   ----- * Pressure Data -----
81   dump$wall <- DumpPres(dump$wall, interp = FALSE)
82
83   ----- * Vorticity Data -----
84   dump$wall <- DumpVortOnly(dump$wall, dump$kinvis)
85
86   ----- * Dump Calc Output -----
87   data_plot <- bind_rows(dump[c("time", "kinvis", "a")])
88   data_plot <- cbind(data_dump, data_plot) %>%
89     mutate(plotname = paste0(
90       airfoil, "-v", sprintf("%0.4f", kinvis), "-t",
91       sprintf("%06.4f", time)))
92
93   ----- * Integral Output -----
94   if (outp == "wall") {
95     list_dump <- c(
96       data_plot = list(data_plot),
97       dump = dump[c("wall")],
98       bvfa = dump[c("bvfa")],
99       flow = dump[c("flow")],
100      inte = dump[c("inte")])
101     names(list_dump) <- c("data_plot", "wall",
102                           "bvfa", "flow", "inte")
103   } else if (outp == "node") {
104     dump$node <- filter(dump$dump, node)
105     list_dump <- c(
106       data_plot = list(data_plot),
107       dump = dump[c("wall")],
108       node = dump[c("node")],
109       bvfa = dump[c("bvfa")],
110       flow = dump[c("flow")],
111       inte = dump[c("inte")])
112     names(list_dump) <- c("data_plot", "wall", "node",
113                           "bvfa", "flow", "inte")
114   }
115 }
```

```

114 } else if (outp == "te") {
115   xvec = c( 0.595,  0.615)
116   yvec = c(-0.046, -0.030)
117   xveci = xvec + c(-0.1*(xvec[2]-xvec[1]), 0.1*(xvec[2]-xvec[1]))
118   yveci = yvec + c(-0.1*(yvec[2]-yvec[1]), 0.1*(yvec[2]-yvec[1]))
119   dump$te <- filter(dump$dump, x >= xveci[1], x <= xveci[2],
120                      y >= yveci[1], y <= yveci[2]) %>%
121   select(x, y, u, v, p, o) %>%
122   unique()
123   list_dump <- c(
124     data_plot = list(data_plot),
125     dump = dump[c("wall")],
126     node = dump[c("te" )],
127     bvfa = dump[c("bvfa")],
128     flow = dump[c("flow")],
129     inte = dump[c("inte")])
130   names(list_dump) <- c("data_plot", "wall", "te",
131                         "bvfa", "flow", "inte")
132 }
133
134 #--- > Produce Plots if Required -----#
135 # Last entry of plot needs to be saveplot
136 if      ("none" %in% plot) {
137 } else {
138   saveplot = plot[1]
139   if      ("airfoil"  %in% plot) {
140     # Plot the airfoil
141     plot_af = PlotAirfoil(
142       dump$dump, dump$wall, data_plot, scalearr = c(10, 4), saveplot) }
143   if ("TEstream"  %in% plot) {
144     # Plot the trailing edge
145     plot_te = PlotTE(
146       dump$dump, dump$wall, data_plot, scalearr = c(10, 4), saveplot) }
147   if ("LEstream"  %in% plot) {
148     # Plot the leading edge
149     plot_le = PlotLE(
150       dump$dump, dump$wall, data_plot, scalearr = c(10, 4), saveplot) }
151   }
152 }
153
154 #--- > Run additional scripts if called -----#
155 if (!is.null(addscr))
156   for (i in 1:length(addscr)) source(addscr[i])
157
158 rm(data_dump, data_plot, dump)
159 # output
160 return(list_dump)
161 }
```