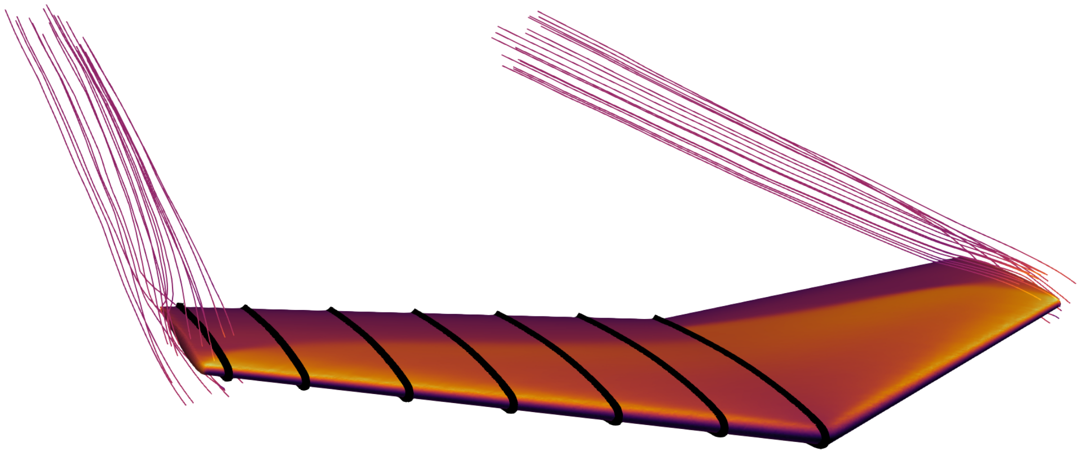
BLASTER

Boundary Layer Adjoint Solver for Transonic External high Reynolds number flows

Theory Manual

Paul Dechamps

Department of Aerospace & Mechanical Engineering
University of Liège



Abstract

The present document provides details of the mathematical model used in BLASTER, a boundary layer adjoint solver for transonic external high Reynolds number flows.

BLASTER is developed by Paul Dechamps at the University of Liège, Belgium, under the supervision of Prof. V. Terrapon and Prof. G. Dimitriadis and with help of Dr. A. Crovato and A. Bilocq since 2022.

Contents

1 Model Description

This section presents the viscous-inviscid interaction model used in BLASTER. After that, the boundary layer equations are derived and the adjoint equations are presented.

1.1 Viscous-Inviscid Interaction Model

Viscous-inviscid interaction (VII) in BLASTER is done by considering that viscous effects are only relevant in a thin shear layer adjacent to the body surface, as depicted in Figure 1.1. The flow is divided into two regions: the inviscid region, where an inviscid flow model is used, and the viscous region, where viscosity is taken into account to model the flow. A coupling strategy is used to correctly represent the flow at the interface. Figure 1.2 represents schematically the algorithm used in BLASTER to solve the VII problem. First, the inviscid solution is obtained and the mach number M_e, the density ρ_e and the velocity u_e at the edge of the boundary layer are used as input to solve the boundary layer equations implemented in the viscous solver. The viscous solution allows to compute the so-called blowing velocity V_e, used as a boundary condition in the inviscid calculation, to account for the presence of the boundary layer in the body vicinity.

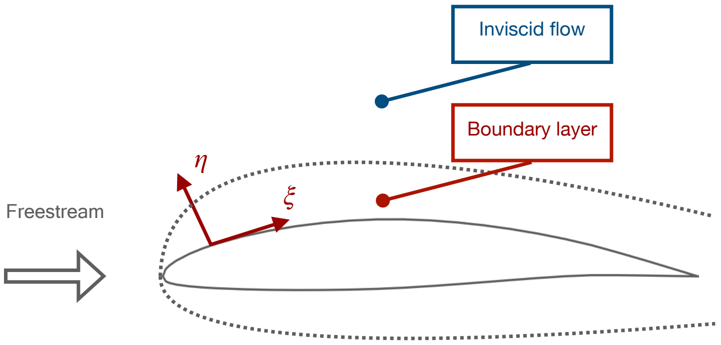


Figure 1.1: Viscous-inviscid interaction model. Viscous effects are only relevant in the thin boundary layer adjacent to the body surface. ξ and η are the tangential and normal body coordinates, respectively.

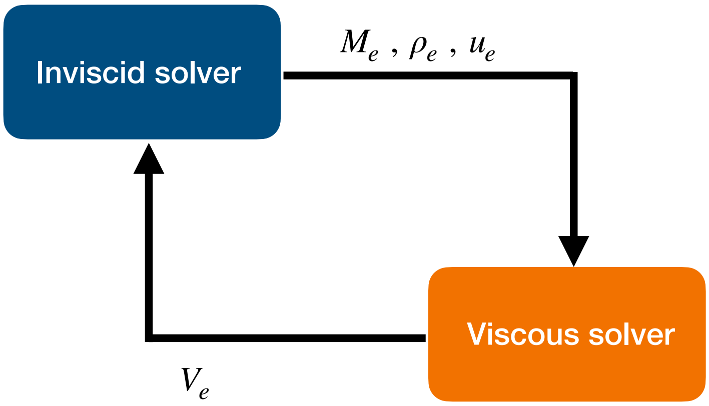


Figure 1.2: Viscous-inviscid interaction algorithm. M_e, ρ_e and u_e are the mach number, the density and the velocity at the edge of the boundary layer, respectively. V_e is the blowing velocity.

BLASTER was originally developed to work with DART [1], an inviscid flow solver using the compressible full-potential equation to represent the flow. But the philosophy was to provide a general framework that could be used with any inviscid flow solver with only few modifications of the latter. The inviscid flow solver is used in BLASTER as an independent module, which can be easily replaced by another inviscid model. Interfaces with the Euler solver in SU2 [2] or panel methods HSPM and UPM [3] have been made, and the general VII algorithm has shown to be very flexible and adapt to various already existing inviscid solvers.

1.2 Boundary Layer Equations

The computation of the flow in the boundary layer in the viscous solver is done by considering a two-dimensional coordinate system attached to the body, (ξ,η) with ξ pointing in the downstream direction and η normal to the solid wall surface as shown in Fig. 1.1. The boundary layer equations implemented in BLASTER are a pseudo-unsteady formulation of the integral boundary layer equations derived by Drela [4] and Nishida [5].

1.2.1 Base equations

The momentum and kinetic energy shape parameter equations are written as

$$\frac{H}{u_e}\frac{\partial\theta}{\partial t}+\frac{\theta}{u_e}\frac{\partial H}{\partial t}+\frac{\theta H}{u_e^2}\frac{\partial u_e}{\partial t}=\frac{\partial\theta}{\partial\xi}+(2+H-M_e^2)\frac{\theta}{u_e}\frac{\partial u_e}{\partial\xi}-\frac{c_{f,l}}{2}, \tag{1.1}$$

$$\begin{aligned}&\frac{1+H(1-H^*)}{u_e}\frac{\partial\theta}{\partial t}+\frac{\theta(1-H^*)}{u_e}\frac{\partial H}{\partial t}+\frac{\theta(2-H^*H)}{u_e^2}\frac{\partial u_e}{\partial t}\\&=\theta\frac{dH^*}{d\xi}+(2H^{**}+H^*(1-H))\frac{\theta}{u_e}\frac{\partial u_e}{\partial\xi}-2c_{d,l}+H^*\frac{c_{f,l}}{2}.\end{aligned} \tag{1.2}$$

where H is the shape factor of the boundary layer, θ is the momentum thickness, $c_{f,l}$ is the local skin friction coefficient, H^* is the kinetic energy shape parameter, H^{**} is the density shape parameter, and $c_{d,l}$ is the local dissipation coefficient. Equations (1.1) and (1.2) are the fundamental equations of the pseudo-unsteady boundary layer system used in the present work, they constitute a combination of compressible steady terms and incompressible unsteady terms. As such, they cannot be used to model unsteady compressible flow. However, they can be used to represent steady-state compressible transonic flows. The unsteady terms are only included in order to circumvent Goldstein's singularity [6] and the transient section of the computed response is discarded.

1.2.2 Shear-lag equation

The shear-lag equation is used to model turbulent flow out of equilibrium, when turbulent production and dissipation are not equals. In the pseudo-unsteady form, the equation is

$$\begin{aligned}\frac{\delta}{U_s u_e C_\tau}\frac{\partial C_\tau}{\partial t}+\frac{2\delta}{U_s u_e^2}\frac{\partial u_e}{\partial t}&=\left(\frac{2\delta}{C_\tau}\right)\frac{\partial C_\tau}{\partial\xi}-5.6\left(C_{\tau_{\mathrm{EQ}}}-C_\tau\omega\right)\\&\quad-2\delta\left(\frac{4}{3H\theta}\left(\frac{c_{f_a}}{2}-\left(\frac{H_k-1}{6.7H_k\omega}\right)^2\right)-\frac{1}{u_e}\frac{\partial u_e}{\partial\xi}\right),\end{aligned} \tag{1.3}$$

where ω is a dissipation factor, $\omega = 1$ on the airfoil and $\omega = 0.9$ in the wake. $\delta^* = \theta H$ is the displacement thickness and C_τ is the shear-stress coefficient, defined as

$$C_\tau = \frac{1}{u_e^2}\left(-\overline{u'v'}\right)_{\max}. \tag{1.4}$$

$\overline{u'v'}$ are the Reynolds stresses, $C_{\tau_{EQ}}$ is the equilibrium shear-stress coefficient, H_k is the kinematic shape parameter. Finally, δ is the boundary layer thickness and U_s is the equivalent normalized wall slip velocity, both computed using the empirical correlations [4] shown in Appendix 3.

1.2.3 Transition model

In order to capture the laminar to turbulent transition in the boundary layer, a pseudo-unsteady formulation of the e^N method [7, 8] is used,

$$\frac{\partial N}{\partial t} + \frac{\partial N}{\partial \xi} = \left(\frac{\mathrm{d}N}{\mathrm{d}\mathrm{Re}_\theta}\frac{\mathrm{d}\mathrm{Re}_\theta}{\mathrm{d}\xi}\right)_{\text{attached}} + A_{\text{separated}}. \tag{1.5}$$

where N is the amplification ratio whose value is prescribed at transition according to experiments,

$$N(\xi) = \ln\frac{A}{A_{\text{crit}}}, \tag{1.6}$$

where $\frac{A}{A_{\text{crit}}}$ is the ratio between the amplification rate of the perturbation wave triggering transition and a critical amplification rate, Re_θ is the Reynolds number based on the momentum thickness θ, The slope $dN/d\xi$ and $A_{\text{separated}}$, which modifies the envelope amplification rate in separated flows, are evaluated following [4, 5]. The critical amplification ratio N_{crit} can be computed following Drela [9],

$$N_{\text{crit}} = -8.43 - 2.4\ln(T_u). \tag{1.7}$$

where

$$T_u \equiv \frac{u'}{U} \tag{1.8}$$

is the turbulence intensity. A commonly used value is $N_{\text{crit}} = 9$ which corresponds to a free-stream turbulence intensity of 0.07%. When $N \geq N_{\text{crit}}$, the flow becomes turbulent.

1.2.4 Coupling strategy

The coupling strategy considered in this work is based on the quasi-simultaneous methodology introduced by Veldman [10]. An approximation of the inviscid flow from thin airfoil theory is solve with the boundary layer system and is written as

$$\left(\frac{\partial u_e}{\partial t} - \tilde{c}\frac{\partial(\theta H)}{\partial t} + \frac{\partial u_e}{\partial \xi} - \tilde{c}\frac{\partial(\theta H)}{\partial \xi}\right)^{(\text{new})} + \left(-\frac{\partial u_e}{\partial \xi} + \tilde{c}\frac{\partial(\theta H)}{\partial \xi}\right)^{(\text{old})} = 0. \tag{1.9}$$

where subscript $(\cdot)^{(\text{new})}$ denotes the current iteration and $(\cdot)^{(\text{old})}$ the previous iteration. This equation is used to avoid Goldstein singularity and to correctly represent the flow at the interface between the inviscid and viscous regions.

1.3 Two-dimensional solution procedure

The computation around static airfoils is done by using the inviscid quantities defined at the nodes on the body surface to solve the boundary layer systems, equations (1.1), (1.2), (1.3), (1.5) and (1.9). The traditional space marching is used and each station in treated individually starting from the stagnation point. The stagnation point is located where u_e is minimum and a boundary condition is imposed there,

$$\theta = \sqrt{\frac{0.075}{\mathrm{Re}\,\frac{du_e}{d\xi}}} \tag{1.10}$$

$$H = 2.23 \tag{1.11}$$

$$C_\tau = 0 \tag{1.12}$$

and laminar closure relations given in Appendix 3 are used at that point. For the downstream stations, a quasi Newton method is used to solve the system. Initial conditions are set to the values obtained at the previous station, where the solution is already converged. Nonphysical initial conditions in the turbulent shear layer are discarded when the shear-stress coefficient C_τ is negative on the upstream point. In that case, $C_\tau(t=0) = 0.03$. The quasi-Newton method yields

$$(J + \frac{I}{\Delta t})\Delta U = -F, \tag{1.13}$$

where $U = [\theta, H, N, u_e, C_\tau]$ is the solution vector, F is the residual vector of equations (1.1), (1.2), (1.3), (1.5) and (1.9), $J = \frac{\partial F}{\partial U}$ is the Jacobian matrix of the system, I is an identity matrix and Δt is the time step. The residuals are evaluated by considering the system written in the form

$$A\frac{\partial U}{\partial t} = B, \tag{1.14}$$

where

$$A = \begin{pmatrix} \frac{H}{u_e} & \frac{\theta}{u_e} & 0 & \frac{\theta H}{u_e^2} & 0 \\ \frac{1+H(1-H^*)}{u_e} & \frac{\theta(1-H^*)}{u_e} & 0 & \frac{\theta(2-H^*H)}{u_e^2} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -\tilde{c}H & -\tilde{c}\theta & 0 & 1 & 0 \\ 0 & 0 & 0 & \frac{2\delta}{U_s u_e^2} & \frac{\delta}{U_s u_e C_\tau} \end{pmatrix} \tag{1.15}$$

and

$$B = \begin{pmatrix} \frac{\partial\theta}{\partial\xi} + (2 + H - M_e^2)\frac{\theta}{u_e}\frac{\partial u_e}{\partial\xi} - \frac{c_{f,l}}{2} \\ \theta\frac{dH^*}{d\xi} + (2H^{**} + H^*(1-H))\frac{\theta}{u_e}\frac{\partial u_e}{\partial\xi} - 2c_{d,l} + H^*\frac{c_{f,l}}{2} \\ \left(\frac{\mathrm{d}N}{\mathrm{d}\mathrm{Re}_\theta}\frac{\mathrm{d}\mathrm{Re}_\theta}{\mathrm{d}\xi}\right)_{\text{attached}} + A_{\text{separated}} \\ \left(\frac{\partial u_e}{\partial\xi} - \tilde{c}\frac{\partial(\theta H)}{\partial\xi}\right)^{\text{(new)}} + \left(-\frac{\partial ue}{\partial\xi} + \tilde{c}\frac{\partial(\theta H)}{\partial\xi}\right)^{\text{(old)}} \\ \left(\frac{2\delta}{C_\tau}\right)\frac{\partial C_\tau}{\partial\xi} - 5.6\left(C_{\tau_{\text{EQ}}} - C_\tau\omega\right) - 2\delta\left(\frac{4}{3H\theta}\left(\frac{c_{fa}}{2} - \left(\frac{H_k-1}{6.7H_k\omega}\right)^2\right) - \frac{1}{u_e}\frac{\partial u_e}{\partial\xi}\right) \end{pmatrix} \tag{1.16}$$

The residual vector is computed as $F = -A^{-1}B$. The corresponding analytical expression for the matrix inversion has been implemented in BLASTER. The Jacobian matrix is computed using a first order finite difference of the residuals,

$$J = \frac{F(U+h) - F(U)}{h} \tag{1.17}$$

where h is the step used for finite difference, 10^{-8} by default. The initial time step is obtained from a CFL number given by the user as,

$$\Delta t = \mathsf{CFL}\frac{\Delta \xi}{u_e} \tag{1.18}$$

where $\Delta\xi$ is the grid spacing in the ξ direction. The CFL is adjusted every time step to accelerate the convergence of the Newton method according to

$$\mathsf{CFL}^{n+1} = \max\left(\mathsf{CFL}^{n}\left[\frac{F(U^0)}{F(U^n)}\right]^{0.7}, 0.1\right) \tag{1.19}$$

where subscript n denotes the time step number. The solution is considered converged when the residual has decreased by a given number of orders of magnitude, which leads to the following condition

$$||F(U^n) + \Delta U/\Delta t|| \leq \varepsilon ||F(U^0)||, \tag{1.20}$$

where ε is the convergence criterion, set to 10^{-8} by default. At each pseudo-time iteration, nonphysical solutions are discarded by preventing the the momentum thickness θ to become negative and the shape factor H to become smaller than 1. This translates into the following conditions,

$$\theta = \max(\theta, 10^{-6}) \tag{1.21}$$

$$H = \max(H, 1.0005) \tag{1.22}$$

At the first wake point, the following boundary condition is imposed,

$$\theta_w = \frac{\theta_{\mathsf{Last\ up}} + \theta_{\mathsf{Last\ lw}}}{2} \tag{1.23}$$

$$H_w = \frac{\theta_{\mathsf{Last\ up}} H_{\mathsf{Last\ up}} + \theta_{\mathsf{Last\ lw}} H_{\mathsf{Last\ lw}}}{\theta_w} \tag{1.24}$$

$$N_w = N_{\mathsf{crit}} \tag{1.25}$$

$$C_{\tau_w} = \frac{\theta_{\mathsf{Last\ up}} C_{\tau,\mathsf{Last\ up}} + \theta_{\mathsf{Last\ lw}} C_{\tau,\mathsf{Last\ lw}}}{\theta_w} \tag{1.26}$$

where Last up denotes the last station on the upper side and Last lw the last station on the lower side. The same equations then the one used on the airfoil are used in the wake by omitting the wall contribution ($c_{f,l} = 0$).

The solution is used to compute the boundary layer displacement thickness,

$$\delta^* = \theta H \tag{1.27}$$

on each node on the surface and in the wake and the blowing velocity is computed on the elements as

$$V_e = \frac{\partial(\rho u_e \delta^*)}{\partial \xi}, \tag{1.28}$$

evaluated using finite difference. The blowing velocity is used as a boundary condition in the inviscid solver to account for the presence of the boundary layer in the body vicinity. The implementation of the boundary condition depends on the inviscid solver.

1.4 Three-dimensional solution procedure

For computations of viscous flows around wings, the viscous equations are solved in two-dimensional boundary layer stations, defined at different spanwise locations, as depicted in Figure 1.3 plane attached to the wing surface. The boundary layer equations are solved for each wing section independently. The inviscid solution, computed on the unstructured mesh on the wing surface, is interpolated to the boundary layer stations to provide the boundary conditions for the boundary layer equations. The interaction between the different stations is not taken into account and the flow is considered two-dimensional on each station. The viscous solution is then used to compute the blowing velocity, which is interpolated back to the wing surface to provide the boundary condition for the inviscid solver. The process is repeated until convergence is achieved.

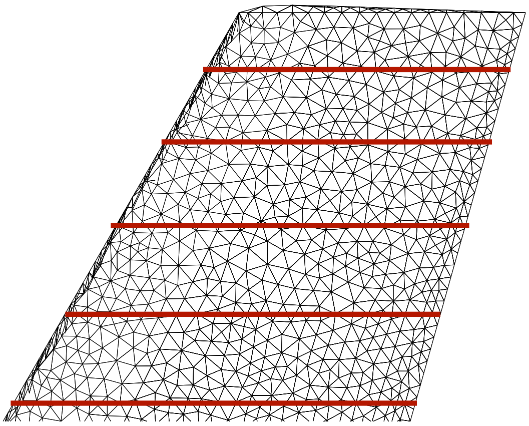


Figure 1.3: Three-dimensional viscous flow computation. The boundary layer equations are solved in two-dimensional planes attached to the wing surface.

The overall algorithm is depicted in Figure 1.4.

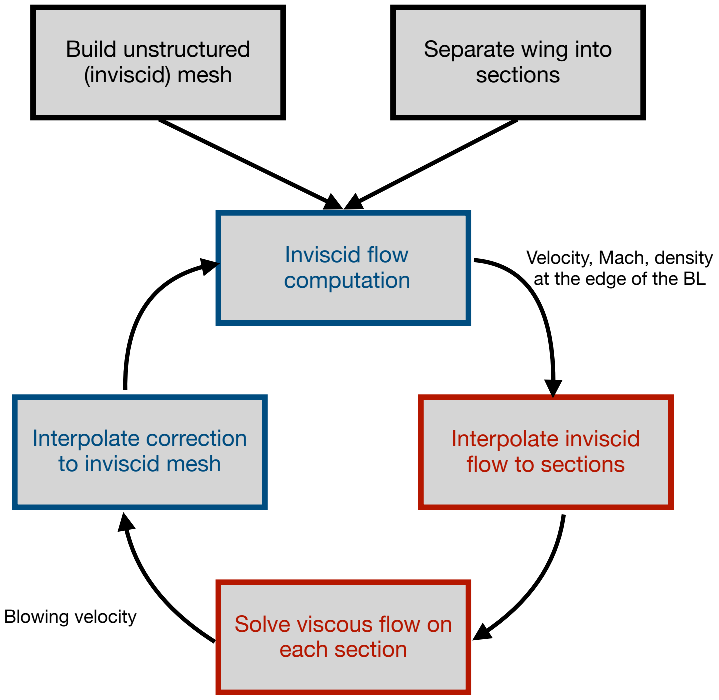

Build unstructured (inviscid) mesh
Separate wing into sections
Inviscid flow computation
Velocity, Mach, density at the edge of the BL
Interpolate inviscid flow to sections
Solve viscous flow on each section
Blowing velocity
Interpolate correction to inviscid mesh

Figure 1.4: Three-dimensional viscous flow computation algorithm.

The interpolation is currently done by using the RBF interpolator implemented in SciPy [11].

2 Adjoint method

This section presents the adjoint method implemented in BLASTER.

2.1 Adjoint formulation of the coupled problem in two-dimensions

The adjoint methodology for two-dimensional configurations is developed to be used in a gradient-based optimization scheme. The VII methdology is detailed in Figure 2.1. The figure presents the inviscid and the viscous solver with the corresponding transfer quantities.

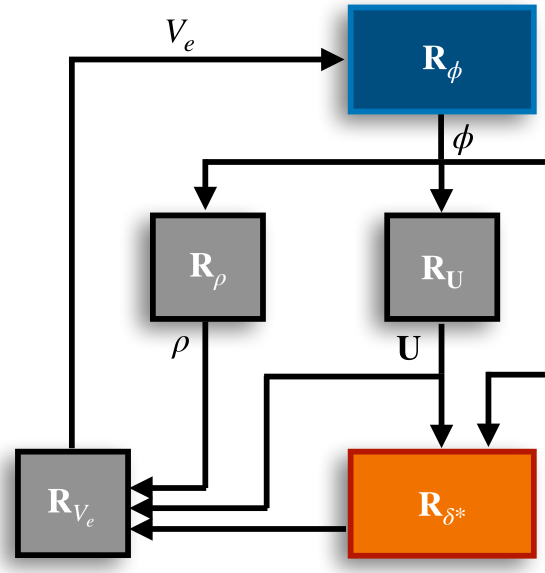


Figure 2.1: Decomposition of the VII problem. In blue, the inviscid flow solver. In red, the viscous flow solver. Grey boxes represent the transfer quantities between the two solvers.

For an aerodynamic optimization, the problem is formulated as follows,

$$\min_{\alpha} F(\phi, \delta^*, \mathbf{x}; \alpha) \tag{2.1}$$

$$\text{s.t. } \mathbf{R}_\phi = 0, \tag{2.2}$$

$$\mathbf{R}_\rho = 0, \tag{2.3}$$

$$\mathbf{R}_\mathbf{U} = 0, \tag{2.4}$$

$$\mathbf{R}_M = 0, \tag{2.5}$$

$$\mathbf{R}_{\delta^*} = 0, \tag{2.6}$$

$$\mathbf{R}_{V_e} = 0, \tag{2.7}$$

$$\mathbf{R}_\mathbf{x} = 0, \tag{2.8}$$

where ϕ is the vector of the potential variables, α is the angle of attack, α is the angle of attack, $\mathbf{x}$ is the mesh coordinates vector, F is the objective function (lift or drag coefficient) and $\mathbf{R}$ are the residuals of the equations. The augmented Lagrangian is defined as,

$$\mathscr{L} = \mathbf{R}_\phi \lambda_\phi + \mathbf{R}_\rho \lambda_\rho + \mathbf{R}_\mathrm{U} \lambda_\mathrm{U} + \mathbf{R}_M \lambda_M + \mathbf{R}_{\delta^*} \lambda_{\delta^*} + \mathbf{R}_{V_e} \lambda_{V_e} + \mathbf{R}_\mathbf{x} \lambda_\mathbf{x} \tag{2.9}$$

where λ are the Lagrange multipliers. Equation (2.9) is differentiated such that $\delta\mathscr{L} = 0$, or

$$\begin{cases}
\partial_\phi F + \partial_\phi \mathbf{R}_\phi \lambda_\phi + \partial_\phi \mathbf{R}_\rho \lambda_\rho + \partial_\phi \mathbf{R}_\mathrm{U} \lambda_\mathrm{U} + \partial_\phi \mathbf{R}_M \lambda_M + \partial_\phi \mathbf{R}_{\delta^*} \lambda_{\delta^*} + \partial_\phi \mathbf{R}_{V_e} \lambda_{V_e} + \partial_\phi \mathbf{R}_\mathbf{x} \lambda_\mathbf{x} & = 0 \\
\partial_\rho F + \partial_\rho \mathbf{R}_\phi \lambda_\phi + \partial_\rho \mathbf{R}_\rho \lambda_\rho + \partial_\rho \mathbf{R}_\mathrm{U} \lambda_\mathrm{U} + \partial_\rho \mathbf{R}_M \lambda_M + \partial_\rho \mathbf{R}_{\delta^*} \lambda_{\delta^*} + \partial_\rho \mathbf{R}_{V_e} \lambda_{V_e} + \partial_\rho \mathbf{R}_\mathbf{x} \lambda_\mathbf{x} & = 0 \\
\partial_\mathrm{U} F + \partial_\mathrm{U} \mathbf{R}_\phi \lambda_\phi + \partial_\mathrm{U} \mathbf{R}_\rho \lambda_\rho + \partial_\mathrm{U} \mathbf{R}_\mathrm{U} \lambda_\mathrm{U} + \partial_\mathrm{U} \mathbf{R}_M \lambda_M + \partial_\mathrm{U} \mathbf{R}_{\delta^*} \lambda_{\delta^*} + \partial_\mathrm{U} \mathbf{R}_{V_e} \lambda_{V_e} + \partial_\mathrm{U} \mathbf{R}_\mathbf{x} \lambda_\mathbf{x} & = 0 \\
\partial_M F + \partial_M \mathbf{R}_\phi \lambda_\phi + \partial_M \mathbf{R}_\rho \lambda_\rho + \partial_M \mathbf{R}_\mathrm{U} \lambda_\mathrm{U} + \partial_M \mathbf{R}_M \lambda_M + \partial_M \mathbf{R}_{\delta^*} \lambda_{\delta^*} + \partial_M \mathbf{R}_{V_e} \lambda_{V_e} + \partial_M \mathbf{R}_\mathbf{x} \lambda_\mathbf{x} & = 0 \\
\partial_{\delta^*} F + \partial_{\delta^*} \mathbf{R}_\phi \lambda_\phi + \partial_{\delta^*} \mathbf{R}_\rho \lambda_\rho + \partial_{\delta^*} \mathbf{R}_\mathrm{U} \lambda_\mathrm{U} + \partial_{\delta^*} \mathbf{R}_M \lambda_M + \partial_{\delta^*} \mathbf{R}_{\delta^*} \lambda_{\delta^*} + \partial_{\delta^*} \mathbf{R}_{V_e} \lambda_{V_e} + \partial_{\delta^*} \mathbf{R}_\mathbf{x} \lambda_\mathbf{x} & = 0 \\
\partial_{V_e} F + \partial_{V_e} \mathbf{R}_\phi \lambda_\phi + \partial_{V_e} \mathbf{R}_\rho \lambda_\rho + \partial_{V_e} \mathbf{R}_\mathrm{U} \lambda_\mathrm{U} + \partial_{V_e} \mathbf{R}_M \lambda_M + \partial_{V_e} \mathbf{R}_{\delta^*} \lambda_{\delta^*} + \partial_{V_e} \mathbf{R}_{V_e} \lambda_{V_e} + \partial_{V_e} \mathbf{R}_\mathbf{x} \lambda_\mathbf{x} & = 0 \\
\partial_\mathbf{x} F + \partial_\mathbf{x} \mathbf{R}_\phi \lambda_\phi + \partial_\mathbf{x} \mathbf{R}_\rho \lambda_\rho + \partial_\mathbf{x} \mathbf{R}_\mathrm{U} \lambda_\mathrm{U} + \partial_\mathbf{x} \mathbf{R}_M \lambda_M + \partial_\mathbf{x} \mathbf{R}_{\delta^*} \lambda_{\delta^*} + \partial_\mathbf{x} \mathbf{R}_{V_e} \lambda_{V_e} + \partial_\mathbf{x} \mathbf{R}_\mathbf{x} \lambda_\mathbf{x} & = 0 \\
\partial_\alpha F + \partial_\alpha \mathbf{R}_\phi \lambda_\phi + \partial_\alpha \mathbf{R}_\rho \lambda_\rho + \partial_\alpha \mathbf{R}_\mathrm{U} \lambda_\mathrm{U} + \partial_\alpha \mathbf{R}_M \lambda_M + \partial_\alpha \mathbf{R}_{\delta^*} \lambda_{\delta^*} + \partial_\alpha \mathbf{R}_{V_e} \lambda_{V_e} + \partial_\alpha \mathbf{R}_\mathbf{x} \lambda_\mathbf{x} & = 0 \\
\mathbf{R}_\phi = 0 \\
\mathbf{R}_\rho = 0 \\
\mathbf{R}_\mathrm{U} = 0 \\
\mathbf{R}_M = 0 \\
\mathbf{R}_{\delta^*} = 0 \\
\mathbf{R}_{V_e} = 0
\end{cases} \tag{2.10}$$

The total gradient of F, the solution must respect equations (2.2) to (2.7) and the linear adjoint equations in the general case are given by

$$\begin{bmatrix}
\partial_\phi \mathbf{R}_\phi^T & \partial_\phi \mathbf{R}_\rho^T & \partial_\phi \mathbf{R}_\mathrm{U}^T & \partial_\phi \mathbf{R}_M^T & \partial_\phi \mathbf{R}_{\delta^*}^T & \partial_\phi \mathbf{R}_{V_e}^T & \partial_\phi \mathbf{R}_\mathbf{x}^T \\
\partial_\rho \mathbf{R}_\phi^T & \partial_\rho \mathbf{R}_\rho^T & \partial_\rho \mathbf{R}_\mathrm{U}^T & \partial_\rho \mathbf{R}_M^T & \partial_\rho \mathbf{R}_{\delta^*}^T & \partial_\rho \mathbf{R}_{V_e}^T & \partial_\rho \mathbf{R}_\mathbf{x}^T \\
\partial_\mathrm{U} \mathbf{R}_\phi^T & \partial_\mathrm{U} \mathbf{R}_\rho^T & \partial_\mathrm{U} \mathbf{R}_\mathrm{U}^T & \partial_\mathrm{U} \mathbf{R}_M^T & \partial_\mathrm{U} \mathbf{R}_{\delta^*}^T & \partial_\mathrm{U} \mathbf{R}_{V_e}^T & \partial_\mathrm{U} \mathbf{R}_\mathbf{x}^T \\
\partial_M \mathbf{R}_\phi^T & \partial_M \mathbf{R}_\rho^T & \partial_M \mathbf{R}_\mathrm{U}^T & \partial_M \mathbf{R}_M^T & \partial_M \mathbf{R}_{\delta^*}^T & \partial_M \mathbf{R}_{V_e}^T & \partial_M \mathbf{R}_\mathbf{x}^T \\
\partial_{\delta^*} \mathbf{R}_\phi^T & \partial_{\delta^*} \mathbf{R}_\rho^T & \partial_{\delta^*} \mathbf{R}_\mathrm{U}^T & \partial_{\delta^*} \mathbf{R}_M^T & \partial_{\delta^*} \mathbf{R}_{\delta^*}^T & \partial_{\delta^*} \mathbf{R}_{V_e}^T & \partial_{\delta^*} \mathbf{R}_\mathbf{x}^T \\
\partial_{V_e} \mathbf{R}_\phi^T & \partial_{V_e} \mathbf{R}_\rho^T & \partial_{V_e} \mathbf{R}_\mathrm{U}^T & \partial_{V_e} \mathbf{R}_M^T & \partial_{V_e} \mathbf{R}_{\delta^*}^T & \partial_{V_e} \mathbf{R}_{V_e}^T & \partial_{V_e} \mathbf{R}_\mathbf{x}^T \\
\partial_\mathbf{x} \mathbf{R}_\phi^T & \partial_\mathbf{x} \mathbf{R}_\rho^T & \partial_\mathbf{x} \mathbf{R}_\mathrm{U}^T & \partial_\mathbf{x} \mathbf{R}_M^T & \partial_\mathbf{x} \mathbf{R}_{\delta^*}^T & \partial_\mathbf{x} \mathbf{R}_{V_e}^T & \partial_\mathbf{x} \mathbf{R}_\mathbf{x}^T
\end{bmatrix}
\begin{bmatrix} \lambda_\phi \\ \lambda_\rho \\ \lambda_\mathrm{U} \\ \lambda_M \\ \lambda_{\delta^*} \\ \lambda_{V_e} \\ \lambda_\mathbf{x} \end{bmatrix}
= -
\begin{bmatrix} \partial_\phi F^T \\ \partial_\rho F^T \\ \partial_\mathrm{U} F^T \\ \partial_M F^T \\ \partial_{\delta^*} F^T \\ \partial_{V_e} F^T \\ \partial_\mathbf{x} F^T \end{bmatrix}. \tag{2.11}$$

Looking at Figure 2.1, the following dependencies are identified,

$$\mathbf{R}_\phi = \mathbf{R}_\phi(\phi, V_e, \alpha, \mathbf{x}) \tag{2.12}$$

$$\mathbf{R}_\rho = \mathbf{R}_\rho(\phi, \rho, \mathbf{x}) \tag{2.13}$$

$$\mathbf{R}_\mathbf{U} = \mathbf{R}_\mathbf{U}(\phi, \mathbf{U}, \mathbf{x}) \tag{2.14}$$

$$\mathbf{R}_M = \mathbf{R}_M(\phi, M, \mathbf{x}) \tag{2.15}$$

$$\mathbf{R}_{\delta^*} = \mathbf{R}_{\delta^*}(\mathbf{U}, M, \delta^*, \mathbf{x}) \tag{2.16}$$

$$\mathbf{R}_{V_e} = \mathbf{R}_{V_e}(\rho, \mathbf{U}, \delta^*, \mathbf{x}) \tag{2.17}$$

$$\mathbf{R}_\mathbf{x} = \mathbf{R}_\mathbf{x}(\mathbf{x}). \tag{2.18}$$

Equation (2.18) only holds because the optimization is purely aerodynamic and the mesh is only impacted by mesh deformation and not by the flow itself. For aerostructural optimization problems, the mesh also depends on variables of the problem (i.e. surface displacement computed by the structural solver). Which allows to identify the derivatives that are zero. For the functionals c_l and c_d, we have the following dependencies,

$$c_l = c_l(\phi, \mathbf{x}) \tag{2.19}$$

$$c_d = c_{d,p}(\phi, \mathbf{x}) + c_{d,f}(\mathbf{U}, M, \delta^*, \mathbf{x}). \tag{2.20}$$

where c_l is the lift coefficient, c_d is the drag coefficient, $c_{d,p}$ is the pressure drag coefficient computed by the inviscid solver and $c_{d,f}$ is the friction drag coefficient computed by the viscous solver. Taking dependencies (2.12) to (2.20) into account, the adjoint system becomes,

$$\begin{bmatrix}
\partial_\phi \mathbf{R}_\phi^T & \partial_\phi \mathbf{R}_\rho^T & \partial_\phi \mathbf{R}_\mathbf{U}^T & \partial_\phi \mathbf{R}_M^T & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \partial_\rho \mathbf{R}_\rho^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \partial_\rho \mathbf{R}_{V_e}^T & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \partial_\mathbf{U} \mathbf{R}_\mathbf{U}^T & \mathbf{0} & \partial_\mathbf{U} \mathbf{R}_{\delta^*}^T & \partial_\mathbf{U} \mathbf{R}_{V_e}^T & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \partial_M \mathbf{R}_M^T & \partial_M \mathbf{R}_{\delta^*}^T & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \partial_{\delta^*} \mathbf{R}_{\delta^*}^T & \partial_{\delta^*} \mathbf{R}_{V_e}^T & \mathbf{0} \\
\partial_{V_e} \mathbf{R}_\phi^T & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \partial_{V_e} \mathbf{R}_{V_e}^T & \mathbf{0} \\
\partial_\mathbf{x} \mathbf{R}_\phi^T & \partial_\mathbf{x} \mathbf{R}_\rho^T & \partial_\mathbf{x} \mathbf{R}_\mathbf{U}^T & \partial_\mathbf{x} \mathbf{R}_M^T & \partial_\mathbf{x} \mathbf{R}_{\delta^*}^T & \partial_\mathbf{x} \mathbf{R}_{V_e}^T & \partial_\mathbf{x} \mathbf{R}_\mathbf{x}^T
\end{bmatrix}
\begin{bmatrix}
\lambda_\phi \\ \lambda_\rho \\ \lambda_\mathbf{U} \\ \lambda_M \\ \lambda_{\delta^*} \\ \lambda_{V_e} \\ \lambda_\mathbf{x}
\end{bmatrix}
= -
\begin{bmatrix}
\partial_\phi F \\ \mathbf{0} \\ \partial_\mathbf{U} F \\ \partial_M F \\ \partial_{\delta^*} F \\ \mathbf{0} \\ \partial_\mathbf{x} F
\end{bmatrix}. \tag{2.21}$$

Notice that the last equation of the system (2.21) is independent of the others because of relation (2.18) and the equation can be solved separately.

Finally, using the equation of the augmented Lagrangian differentiated with respect to the angle of attack α in system (2.9), it comes

$$d_\alpha F = \partial_\alpha F^T - \partial_\alpha \mathbf{R}_\phi^T \lambda_\phi \tag{2.22}$$

because only $\mathbf{R}_\phi$ depends on α.

From the last equation of system (2.21), we have

$$d_{\mathbf{x}}F = \lambda_{\mathbf{x}} = \partial_x \mathbf{R}_{\mathbf{x}}^{-T} \big(\partial_x F^T - \partial_{\mathbf{x}} \mathbf{R}_{\phi}^T \lambda_{\phi} - \partial_{\mathbf{x}} \mathbf{R}_{\rho}^T \lambda_{\rho} \\ - \partial_{\mathbf{x}} \mathbf{R}_{\mathbf{U}}^T \lambda_{\mathbf{U}} - \partial_{\mathbf{x}} \mathbf{R}_{M}^T \lambda_{M} - \partial_{\mathbf{x}} \mathbf{R}_{\delta^*}^T \lambda_{\delta^*} - \partial_{\mathbf{x}} \mathbf{R}_{V_e}^T \lambda_{V_e} \big), \tag{2.23}$$

which is the gradient of the objective function with respect to the mesh coordinates [12]. Since the mesh deformation on the surface is imposed and the volume deformation is computed, $\partial_{\mathbf{x}}\mathbf{R}_{\mathbf{x}}$ is an identity matrix for the surface node and $d_{\mathbf{x}}F = \lambda_{\mathbf{x}}$ for these nodes. Therefore, equation (2.23) is used to compute the gradient of the objective function with respect to the mesh coordinates.

2.2 Partial gradients

In system (2.21), 32 partial derivatives have to be determined to obtain the partial gradients of the lift and drag coefficients with respect to the angle of attack and the mesh coordinates. Because the viscous solver involves closure relations which depends on the flow regime and the values of the parameters, there exist multiple configurations of the system of equations. Developing analytical jacobian matrices for the different cases would require considering at least $2^{13} = 8192$ different combinations. Therefore, it is not straightforward to implement an analytical expression of the jacobian matrix for the viscous solver. Alternatives exist to compute sensitivities of the viscous variables with respect to the input flow conditions such as finite difference, complex-step or automatic differentiation. Automatic differentiation is probably the best choice regarding the computational cost of the gradient evaluations but may lead to a high memory footprint. Considering the complex-step approach would lead to a very intrusive implementation as it would require that the solver can handle complex numbers. This would lead to an increase in computational cost, even when using the solver for analysis only. This is why finite difference is currently chosen to compute the partial gradients of the viscous solver. The main downside of finite difference is the truncation error that can be introduced in the gradient computation. The following derivatives are computed using finite difference,

$$\partial_M \mathbf{R}_{\delta^*},\ \partial_{\mathbf{U}} \mathbf{R}_{\delta^*},\ \partial_{\delta^*} \mathbf{R}_{\delta^*},\ \partial_{\mathbf{x}} \mathbf{R}_{\delta^*},\ \partial_{\mathbf{U}} c_{d,f},\ \partial_M c_{d,f},\ \partial_{\delta^*} c_{d,f},\ \partial_{\mathbf{x}} c_{d,f}.$$

Because the viscous calculation requires a small amount of computational time, computing these derivatives through finite difference in done in less than 10 seconds on accurate grids ($\sim$ 500 surface nodes). Note that resorting to automatic differentiation may be needed in three-dimensional cases whereby the number of surface nodes is considerably larger. All other derivatives are computed analytically. $\partial_{\phi} R_{\phi}$ is the inviscid flow jacobian computed by taking the derivative of equation (2.2). The derivatives $\partial_{\phi} R_{\rho}$, $\partial_{\phi} R_M$ and $\partial_{\phi} R_{\mathbf{U}}$ are obtained by differentiating equations (2.3), (2.5) and (2.4), respectively. The expressions of the different quantities are given in residual form in Crovato [13]. Derivatives are given on one element. For the velocity, which derives from the potential function (i.e $\mathbf{U} = \nabla\phi$),

$$\partial_{\phi_j} \mathbf{U} = \partial_{\nabla \phi_k} \mathbf{U} \partial_{\phi_j} \nabla \phi_k = \partial_{\phi_j} \nabla \phi_k = \nabla N_j = J_e^{-1} \nabla^{\xi} N_j, \tag{2.24}$$

where J_e is the jacobian of the change of variable between the element local coordinates ξ and the global coordinates. The gradients of the Mach number and the density are given by

$$\partial_{\phi_j}M = \partial_{\nabla\phi_k}M \ \partial_{\phi_j}\nabla\phi_k = \partial_{\nabla\phi_k}M \ \partial_{\phi_j}\mathbf{U} = \frac{|\nabla\phi|^2}{M}J_e^{-1}\nabla^\xi N_j, \tag{2.25}$$

and by

$$\partial_{\phi_j}\rho = \partial_{\nabla\phi_k}\rho \ \partial_{\phi_j}\nabla\phi_k = \partial_{\nabla\phi}\rho \ \partial_{\phi_j}\mathbf{U} = -M^2\rho^{2-\gamma}\nabla\phi^T J_e^{-1}\nabla^\xi N_j. \tag{2.26}$$

The gradients of $\mathbf{U}$, M and ρ with respect to the mesh coordinates are obtained similarly. They are given by

$$\begin{aligned}\partial_{x_j}\mathbf{U} = \partial_{\nabla\phi_k}\mathbf{U}\partial_{x_j}\nabla\phi_k &= \frac{\partial(J_e^{-1}\nabla^\xi N_j\phi_j)}{\partial x_j}\\ &= \left(-J_e^{-1}\frac{\partial J_e}{\partial x_j}J_e^{-1}\right)\nabla^\xi\phi = \left(-J_e^{-1}\frac{\partial J_e}{\partial x_j}\right)\nabla^x\phi,\end{aligned} \tag{2.27}$$

$$\begin{aligned}\partial_{x_j}M = \partial_{\nabla\phi_k}M \ \partial_{x_j}\nabla\phi_k &= \frac{|\nabla\phi|^2}{M}\frac{\partial(J_e^{-1}\nabla^\xi N_j\phi_j)}{\partial x_j}\\ &= \frac{|\nabla\phi|^2}{M}\left(-J_e^{-1}\frac{\partial J_e}{\partial x_j}J_e^{-1}\right)\nabla^\xi\phi = \frac{|\nabla\phi|^2}{M}\left(-J_e^{-1}\frac{\partial J_e}{\partial x_j}\right)\nabla^x\phi,\end{aligned} \tag{2.28}$$

$$\begin{aligned}\partial_{x_j}\rho = \partial_{\nabla\phi_k}\rho \ \partial_{x_j}\nabla\phi_k &= -M^2\rho^{2-\gamma}\nabla\phi^T\frac{\partial(J_e^{-1}\nabla^\xi N_j\phi_j)}{\partial x_j}\\ &= -M^2\rho^{2-\gamma}\nabla\phi^T\left(-J_e^{-1}\frac{\partial J_e}{\partial x_j}J_e^{-1}\right)\nabla^\xi\phi = -M^2\rho^{2-\gamma}\nabla\phi^T\left(-J_e^{-1}\frac{\partial J_e}{\partial x_j}\right)\nabla^x\phi.\end{aligned} \tag{2.29}$$

The derivatives of the blowing velocity $\partial_\rho\mathbf{R}_{V_e}$, $\partial_\mathbf{U}\mathbf{R}_{V_e}$, $\partial_{\delta^*}\mathbf{R}_{V_e}$ and $\partial_\mathbf{x}\mathbf{R}_{V_e}$ are computed by differentiating equation (1.28). In residual form for one station, this equation writes,

$$(R_{V_e})_{i-1/2} = (V_e)_{i-1/2} - \frac{\rho_i u_{e,i}\delta_i^* - \rho_{i-1}u_{e,i-1}\delta_{i-1}^*}{\rho_i(\xi_i-\xi_{i-1})}. \tag{2.30}$$

The derivative of the blowing velocity with respect to the density is given by

$$\begin{aligned}\partial_{\rho_j}(R_{V_e})_{i-1/2} =& \delta'_{i-1/2,j}\left(-\frac{1}{\rho_i^2}\left(\frac{\rho_i u_{e_i}\delta_i^* - \rho_{i-1}u_{e,i-1}\delta_{i-1}^*}{\xi_i-\xi_{i-1}}\right) + \frac{u_{e_i}\delta_i^* - u_{e,i-1}\delta_{i-1}^*}{\rho_i(\xi_i-\xi_{i-1})}\right)\\ &+ \delta'_{i-1/2,j-1}\left(\frac{\rho_{i-1}u_{e,i-1}\delta_{i-1}^*}{\rho_i(\xi_i-\xi_{i-1})}\right),\end{aligned} \tag{2.31}$$

where δ' is Kronecker's delta. The derivatives of the blowing velocity with respect to the displacement thickness is given by

$$\begin{aligned}\partial_{\delta_j^*}(R_{V_e})_{i-1/2} =& \delta'_{i-1/2,j}\left(\frac{u_{e_i}}{\xi_i-\xi_{i-1}}\right)\\ &+ \delta'_{i-1/2,j-1}\left(\frac{-u_{e_{i-1}}}{\rho_i(\xi_i-\xi_{i-1})}\right).\end{aligned} \tag{2.32}$$

Finally, the derivatives of the blowing velocity with respect to the velocity is given by

$$\partial_{u_j^k}(R_{V_e})_{i-1/2} = \partial_{u_{e_j}}(R_{V_e})_{i-1/2}\partial_{u_j^k}u_{e_j}, \tag{2.33}$$

where

$$\partial_{u_{e_j}}(R_{V_e})_{i-1/2} = \left[\delta'_{i-1/2,j}\left(\frac{\delta_i^*}{\xi_i - \xi_{i-1}}\right) + \delta'_{i-1/2,j-1}\left(\frac{\rho_{i-1}\delta_{i-1}^*}{\rho_i(\xi_i - \xi_{i-1})}\right)\right] \tag{2.34}$$

and

$$\partial_{u_j^k}u_{e_j} = \frac{u_j^k}{\sqrt{(u_j^0)^2 + (u_j^1)^2}}. \tag{2.35}$$

3 Closure models

Closure models required to solve the integral boundary layer system are presented here. Given the non-interest in the flow unsteadiness, the empirical correlations do not account for time dependent effects. Most of the correlations are taken from [4] and [5]. Some correlations are taken from the source code of `XFOIL` v6.99 [14].

The parameters to be modeled are H^*, H^{**}, c_f and c_d and, following [4], the following depencies are assumed,

$$\begin{aligned} c_f &= F\left(H_k, M_e, \mathrm{Re}_\theta\right); \\ c_d &= F\left(H_k, M_e, \mathrm{Re}_\theta\right); \\ H^* &= F\left(H_k, M_e, \mathrm{Re}_\theta\right); \\ H^{\star\star} &= F\left(H_k, M_e\right). \end{aligned} \tag{3.1}$$

The kinematic shape parameter, H_k, is derived after [15] as

$$H_k = \frac{H - 0.290M_e^2}{1 + 0.113M_e^2}. \tag{3.2}$$

while Re_θ is the Reynolds number based on the momentum thickness θ,

$$\mathrm{Re}_\theta = \frac{\rho_e u_e \theta}{\mu_e}, \tag{3.3}$$

the different closure terms are presented depending on the flow regime.

3.1 Laminar boundary layer

In a steady incompressible laminar flow, the velocity profile is similar to the Falkner-Skan profile, thanks to the self-similarity assumption (see [16]). The kinematic energy shape parameter is expressed locally and differently whether the flow is attached or detached,

$$H^* = \begin{cases} 1.528 + 0.0111\frac{(H_k-4.35)^2}{H_k+1} - 0.0278\frac{(H_k-4.35)^3}{H_k+1} & \\ \quad -0.0002\left[(H_k-4.35)H_k\right]^2, & H_k < 4.35 \\ 1.528 + 0.015\frac{(H_k-4.35)^2}{H_k} & H_k \geq 4.35 \end{cases} \tag{3.4}$$

For compressible flow, a correction on H^* is required,

$$H^* = \frac{H^* + 0.028M_e^2}{1 + 0.014M_e^2} \tag{3.5}$$

The density shape parameter, H^{**} is a corrected kinematic shape parameter only defined for a compressible flow. It is obtained using the expression

$$H^{**} = \left(\frac{0.064}{H_k - 0.8} + 0.251\right) M_e^2. \tag{3.6}$$

A normalized local friction coefficient, defined as

$$\overline{c_f} = c_f \frac{\mathrm{Re}_\theta}{2}, \tag{3.7}$$

can be obtained using

$$\overline{c_f} = \begin{cases} \frac{1}{2}\left[-0.07 + 0.0727 \frac{(5.5-H_k)^3}{H_k+1}\right], & H_k < 5.5 \\ \frac{1}{2}\left[-0.07 + 0.015\left(1 - \frac{1}{H_k-4.5}\right)^2\right]. & H_k \geq 5.5 \end{cases} \tag{3.8}$$

The same procedure is applied to the dissipation coefficient. Successively,

$$\overline{c_d} = 2\,\mathrm{Re}_\theta \frac{c_d}{H^*} \tag{3.9}$$

is defined and

$$\overline{c_d} = \begin{cases} 0.207 + 0.00205\,(4-H_k)^{5.5}, & H_k < 4 \\ 0.207 - 0.0016 \frac{(H_k-4)^2}{1+0.02(H_k-4)^2}, & H_k \geq 4 \end{cases} \tag{3.10}$$

without compressibility corrections on c_f or c_d.

3.2 Turbulent boundary layer

In the case of a turbulent boundary layer, the self-similarity assumption is not valid. [5] suggests to start from an expression of the skin friction coefficient.

$$F_c c_f = \frac{0.3 e^{-1.33 H_k}}{\left(\frac{\ln Re_\theta}{2.3026}\right)^{-1.74-0.31H}} + 0.00011\left[\tanh\left(4 - \frac{H_k}{0.875}\right) - 1\right] \tag{3.11}$$

where $F_c = \sqrt{1+0.2M_e^2}$. This particular expression of F_c is given by [4] and is used to compute subsonic and transonic flow solutions. The kinetic energy shape parameter is defined as

$$H^* = \begin{cases} 1.5 + \frac{4}{Re_\theta} + \left(0.5 - \frac{4}{Re_\theta}\right)\left(\frac{H_0-H_k)}{H_0-1}\right)^2\left(\frac{1.5}{H_k+0.5}\right), & H_k < H_0 \\ 1.5 + \frac{4}{Re_\theta} + (H_k - H_0)^2\left[\frac{0.007 \ln Re_\theta}{\left(H_k - H_0 + \frac{4}{\ln Re_\theta}\right)} + \frac{0.015}{H_k}\right], & H_k \geq H_0 \end{cases} \tag{3.12}$$

with

$$H_0 = \begin{cases} 3 + \frac{400}{Re_0} & \mathrm{Re}_\theta \geq 400 \\ 4. & \mathrm{Re}_\theta < 400 \end{cases} \tag{3.13}$$

The density shape parameter is defined as

$$H^{**} = \left(\frac{0.064}{H_k - 0.8} + 0.251\right) M_e^2. \tag{3.14}$$

Lastly, a correlation for the dissipation coefficient is required. The work of [17], among others, shows that if turbulent production and dissipation are in near equilibrium, c_d depends only locally on the turbulent shear layer's parameters. This equilibrium is usually assumed in most algebraic turbulence models. However, [18] brought experimental evidence of significant upstream history effects on Reynolds stresses for flows demonstrating an adverse pressure gradient. In such scenario, it is clear that an empirical correlation for the dissipation coefficient can not be obtained easily. Following [19] and [20], the method consists in separating the contributions to the dissipation coefficient between the local wall layer contribution and the upstream effect. Each being defined by a velocity and a stress scale. Introducing the equivalent normalized wall slip velocity,

$$U_s = \frac{H^*}{2}\left(1-4\left(\frac{H_k-1}{3H}\right)\right), \tag{3.15}$$

the dissipation coefficient is expressed as

$$c_d = \frac{c_f}{2}U_s + C_\tau(1-U_s). \tag{3.16}$$

The wall contribution from the wall layer, the first term of the right hand side, driven by c_f, is determined strictly by the local boundary layer. The second term of the right hand side is driven by the shear stress coefficient C_τ and is influenced by upstream conditions in the turbulent shear layer.

3.3 Empirical formula of the transition model

$$\frac{dN}{d\mathrm{Re}_\theta} = 0.028(H_k-1) - 0.0345\exp\left[-\left(\frac{3.87}{H_k-1}-2.52\right)^2\right] \tag{3.17}$$

and

$$\theta\frac{d\mathrm{Re}_\theta}{d\xi} = -0.05 + \left(\frac{2.7}{H_k-1}\right) - \left(\frac{5.5}{H_k-1}\right)^2 + \left(\frac{3}{H_k-1}\right)^3 + 0.1\exp\left(\frac{-20}{H_k-1}\right) \tag{3.18}$$

$$\theta A_{\text{separated}} = H_{\text{corr}}\left(0.086\tanh\left[1.2\left(\log_{10}\mathrm{Re}_\theta - 0.3 + 0.35\exp\left(-0.15(H_k-5)\right)\right)\right] - \left(\frac{0.25}{H_k-1}\right)^{1.5}\right) \tag{3.19}$$

where the correction term H_{corr} is obtained by

$$H_{\text{corr}} = \begin{cases} 0 & H_{\text{norm}} \le 0, \\ 3H_{\text{norm}}^2 - 2H_{\text{norm}}^3 & 0 < H_{\text{norm}} < 1, \\ 1 & H_{\text{norm}} \ge 1 \end{cases} \tag{3.20}$$

with

$$H_{\text{norm}} = \frac{H_k-3.5}{4-3.5} \tag{3.21}$$

References

[1] Adrien Crovato. *Steady transonic aerodynamic and aeroelastic modeling for preliminary aircraft design*. PhD thesis, Universite de Liege (Belgium), 2021.

[2] Thomas D Economon, Francisco Palacios, Sean R Copeland, Trent W Lukaczyk, and Juan J Alonso. Su2: An open-source suite for multiphysics simulation and design. *Aiaa Journal*, 54(3):828–846, 2016.

[3] Corentin Thomée et al. Master thesis and internship [br]-master's thesis: Comparison of steady and unsteady viscous-inviscid coupling strategies in blaster [br]-internship. 2024.

[4] Mark Drela and Michael B Giles. Viscous-inviscid analysis of transonic and low reynolds number airfoils. *AIAA journal*, 25(10):1347–1355, 1987.

[5] Brian Nishida and Mark Drela. Fully simultaneous coupling for three-dimensional viscous/inviscid flows. In *13th Applied Aerodynamics Conference*, page 1806, 1995.

[6] Sidney Goldstein. On laminar boundary-layer flow near a position of separation. *The Quarterly Journal of Mechanics and Applied Mathematics*, 1(1):43–69, 1948.

[7] JL Van Ingen. A suggested semi-empirical method for the calculation of the boundary layer transition region. *Technische Hogeschool Delft, Vliegtuigbouwkunde, Rapport VTH-74*, 1956.

[8] Apollo Milton Olin Smith. Transition, pressure gradient and stability theory. *Douglas Aircraft Co., Report ES 26388*, 1956.

[9] Mark Drela and Michael B Giles. Viscous-inviscid analysis of transonic and low reynolds number airfoils. *AIAA journal*, 25(10):1347–1355, 1987.

[10] Arthur EP Veldman. A simple interaction law for viscous–inviscid interaction. *Journal of Engineering Mathematics*, 65(4):367–383, 2009.

[11] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[12] Markus Widhalm, Joël Brezillon, Ilic Caslav, and Tobias Leicht. Investigation on adjoint based gradient computations for realistic 3d aero-optimization. 09 2010.

[13] Adrien Crovato. *Steady Transonic Aerodynamic and Aeroelastic Modeling for Preliminary Aircraft Design*. PhD thesis, University of Liège, October 2020.

[14] Mark Drela. Xfoil: An analysis and design system for low reynolds number airfoils. In *Low Reynolds number aerodynamics*, pages 1–12. Springer, 1989.

[15] David L Whitfield. Integral solution of compressible turbulent boundary layers using improved velocity profiles. Technical report, Arnold Engineering Development Center Arnold AFB TN, 1978.

[16] VM Falkneb and Sylvia W Skan. Lxxxv. solutions of the boundary-layer equations. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 12(80):865–896, 1931.

[17] JE Green, DJ Weeks, and JWF Brooman. Prediction of turbulent boundary layers and wakes in compressible flow by a lag-entrainment method. Technical report, AERONAUTICAL RESEARCH COUNCIL LONDON (ENGLAND), 1973.

[18] Perry Goldberg. Upstream history and apparent stress in turbulent boundary layers. Technical report, Cambridge, Mass.: Massachusetts Institute of Technology, Gas Turbine . . . , 1966.

[19] J Thomas. Integral boundary-layer models for turbulent separated flows. In *17th Fluid Dynamics, Plasma Dynamics, and Lasers Conference*, page 1615, 1984.

[20] JC LeBalleur. Strong matching method for computing transonic viscous flows including wakes and separations: Lifting airfoils. *La Rech. Aerospatiale, Bull. Bimestriel,(Paris)*, (1981):161–185, 1981.