Fluid Dynamics and Co-located Conferences
June 24-27, 2013, San Diego, CA
21st AIAA Computational Fluid Dynamics Conference

10.2514/6.2013-2943

Advances in the development of a High Order, Viscous-Inviscid Interaction Solver

D. Moro[*], N. C. Nguyen[†], J. Peraire[‡] and M. Drela[§]
Massachusetts Institute of Technology, Cambridge, MA 02139, USA

We present a high order viscous-inviscid interaction solver for aerodynamic flows. Our approach is based on a split formulation where the viscous and inviscid effects are solved in two different domains that overlap near solid walls or the wake centerline and are coupled using the equivalent mass transpiration proposed by Lighthill. Both the viscous and inviscid solvers are based on a high order Hybridized Discontinuous Galerkin scheme (HDG). In the case of the viscous solver, the mesh is extruded on the fly from the surface mesh using a normal scaling indicator δ, based on integrated boundary layer quantities, that is computed as part of the solution. Results will be presented to validate the coupled viscous-inviscid solver in a variety of cases using the Euler equations as well as the full potential equation as inviscid models.

I. Motivation

The growth in computer power over the last two decades has increased the number of CFD practitioners in the industry. Numerical simulation tools have been increasingly used as substitutes for expensive and cumbersome experiments, often reducing the length and cost of a design cycle dramatically.[1] In the meantime, researchers in the field have put significant effort into advancing the state of the art of the algorithms so that bigger problems can be solved in less time and with greater accuracy. From the point of view of the discretization, most of the work has focused on high order schemes in different variants: Discontinuous Galerkin (DG), Spectral Finite Differences (SD) and Spectral Finite Volumes (SV), etc. Despite their theoretical advantages, these schemes have not been widely adopted in the industry for several reasons. Amongst others:

- They are considered computationally expensive for the levels of accuracy of interest,[2] so their higher cost cannot be leveraged by a faster convergence rate.
- High order anisotropic meshes for cases involving turbulent boundary layers and shocks are hard to generate in an automated fashion and require significant amount of intervention in the process.
- They are deemed to suffer from robustness issues for certain model equations, especially when there is not enough resolution to capture relevant features.

The CFD community is well aware of all these shortcomings and is putting effort into alleviating them in one way or another.[2–4] And despite this, it is not clear whether these tools would easily find their way into all the stages of the design, or rather very detailed and specific cases.

To see why this might be the case, one has to reckon the fact that in certain stages of the design, an engineer is not looking for a full fledged Navier-Stokes result but rather a quick and reliable solution that has the right dependency on the parameters of the problem. This helps the engineer build his or her intuition and make decisions along the way instead of spending time fixing meshes and monitoring convergence. A very important example of this type of code would be TRANAIR, used throughout the Boeing company[1,5] for compressible external aerodynamics. In comparison with some state of the art codes, TRANAIR is

[*] PhD Candidate, AIAA Student Member.
[†] Research Scientist, AIAA Member.
[‡] H. N. Slater Professor of Aeronautics and Astronautics, AIAA Fellow.
[§] Terry J. Kohler Professor of Fluid Dynamics, AIAA Fellow.

rather outdated (full potential equation, linear finite element discretization), however, its automatic adaptive refinement capabilities combined with the robustness of its nonlinear solver make it a perfect example of a reliable design code. In this case, the user is willing to give away some of the physics to obtain a reasonable solution within a few minutes. The error associated with this solution will be bounded provided the code is used within its design envelope. A very similar idea is found behind the Euler solver CART3D that is widely used at NASA. Both TRANAIR and CART3D have been augmented using a 2D streamline integral boundary layer method based on the work of Drela *et al.*[6] so that quasi-2D viscous flows can be reasonably approximated in the same fast and reliable way.

The problem comes when engineers try to use these tools to analyze non-conventional geometries for future transport aircraft such as the Blended Wing Body or the D8 concepts in which many elements (wing, body, propulsion, etc.) are highly integrated. In this case, 3D viscous effects might be substantial and the validity of the results questionable. Several attempts were made at trying to devise 3D integral boundary layer formulations[7,8] with limited success. It seems like a more general approach would be required to circumvent these issues.

In this work, building on our previous research,[9] we will use a differential method based on the solution of the Navier-Stokes Equations in the viscous layer only. In principle, the restriction of the viscous effects to a thin layer readily introduces savings in computational cost. Still, the cost can be significant in terms of pre-process time (mostly mesh generation). In the proposed solver, mesh generation in the viscous domain is automated to the point where only a surface mesh and extrusion directions are required. Furthermore, in order to make the most out of the available resolution in the normal direction (that is a function of the order of the approximation and the number of elements), we use the approach by Allmaras[10] and evolve said mesh on the fly so that it follows the boundary layer thickness in an optimal way.

To deal with the external inviscid flow, we will use a high order solver to discretize the full potential or Euler equations on a fixed mesh. The choice of inviscid model is flexible and depends on the problem at hand (subsonic, transonic, etc.). Other options such as a panel method would also be compatible with this approach but where not implemented in this work.

Combining both solvers yields a scheme where some of the physics is simplified. The level of simplification depends on the coupling conditions between them. In any case, we should expect this scheme to be more physically correct than integral boundary layer methods since these are a further simplification of the Navier-Stokes Equations. The main gain of the segregated approach is that each solver takes care of what it does best. More precisely, we let the external solver deal with inviscid features (such as shocks, wakes, isentropic flows, etc.) on a mesh that can be very coarse compared to the boundary layer, while we solve for the viscous effects on a self adapted grid that grows from the surface and is conforming with the boundaries by construction. We believe that the combination of this philosophy with the advantages of a high order discretization yields a balanced scheme in terms of accuracy versus computational cost. Throughout this work we will refer to our scheme as Virtual Grid Navier-Stokes or VGNS.

Our goal is to devise a scheme that is faster than a full Navier-Stokes solver of similar characteristics and can provide a solution to a transonic turbulent problem with a certain level of reliability. We do expect the scheme to be able to deal with separation, as will be shown later, broadening the range of applications beyond clean configurations such as cruise. It would be interesting to see how this kind of solver fits in a multi-fidelity CFD tool as an intermediate step between an integral boundary layer code and a full Navier-Stokes solver.

In what follows, we will first describe the formulation of the problem. Next, we will say a few words about the discretization schemes used. We then present some results to validate our approach. Conclusions and future research directions will be discussed last.

II. Formulation

We are interested in solving the problem of the laminar compressible viscous flow around a geometry of interest, which can be fully described by the Navier-Stokes equations. This system of conservation laws can be split into inviscid and viscous terms, of which the latter are negligible away from shear layers. Dropping the viscous terms yields the Euler equations, which describe the external inviscid flow *outside* the boundary layer.

In principle, one could devise a strategy to solve this problem based on a non-overlapping partition of the domain into viscous and inviscid regions, where the corresponding system of conservation laws is solved. For this, we require a way to identify these regions (e.g. identifying the boundary layer edge), which is not a trivial task.

In this work we will use the opposite approach and assume that the viscous and inviscid domains overlap. In particular, we will solve the inviscid equations in the whole domain, all the way to the wall, where they will overlap with the viscous equations, that are solved only in a region adjacent to the wall (or wake). We will refer to these fields as the Equivalent Inviscid Flow (EIF) and the Real Viscous Flow (RVF) respectively. A sketch of the velocity profiles for the EIF and RVF in a typical case is included in Figure 1.

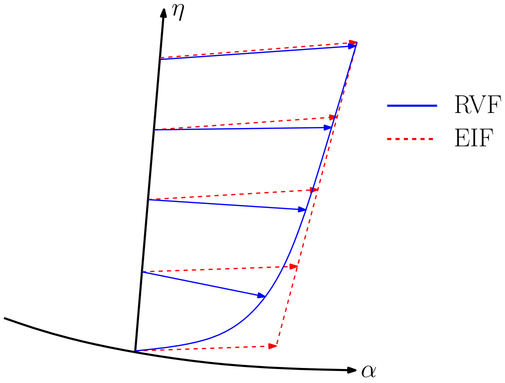


Figure 1: EIF and RVF velocity profiles in a typical boundary layer.

In what follows, we will describe the fluid models used for each of them as well as the coupling boundary conditions that ensure the EIF and the RVF match at the edge of the boundary layer.

A. RVF model

The Real Viscous Flow will be modeled using the Navier-Stokes equations in 2D:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{F} = \nabla \cdot \mathbf{G}, \tag{1}$$

where:

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho E \end{bmatrix}, \qquad \mathbf{F} = \begin{bmatrix} \rho v_i \\ \rho v_i v_1 + p\delta_{i1} \\ \rho v_i v_2 + p\delta_{i2} \\ \rho v_i H \end{bmatrix}, \qquad \mathbf{G} = \begin{bmatrix} 0 \\ \tau_{1i} \\ \tau_{2i} \\ \sum_{j=1}^{d} \tau_{ij} u_j + q_i \end{bmatrix}, \tag{2}$$

$$\tau_{ij} = \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial v_k}{\partial x_k} \right), \qquad q_i = \kappa \frac{\partial T}{\partial x_i}. \tag{3}$$

Here, ρ represents the density, v_i is the i-th component of the velocity, E is the total specific energy and $H = E + p/\rho$ is the total specific enthalpy. The pressure (p), the density and the temperature (T) obey the ideal gas law ($p = \rho RT$). The coefficients μ and κ are the dynamic viscosity and heat conductivity respectively. Here, μ is taken to follow Sutherland's law ($\mu = \mu(T)$) and a constant Prandtl number assumption is used to compute κ: $Pr = \frac{c_p \mu}{\kappa} = 0.72$.

The most commonly used boundary conditions are the following:

- Far-field: $\mathbf{u} = \mathbf{u}_\infty, \quad |\mathbf{x}| \to \infty$,
- Subsonic outlet: $p = p_{\text{out}}$, extrapolate ρ and $\mathbf{v}$,
- Isothermal wall: $(v_1, v_2) = 0, \quad T = T_w$,
- Adiabatic wall: $(v_1, v_2) = 0, \quad \mathbf{q} \cdot \mathbf{n} = q_w$.

The implementation of the first two conditions requires a Riemann decomposition at the boundary, described by Peraire *et al.*[11] The coupling boundary condition with the EIF is not included here and will be described in Section C.

B. Inviscid models

The VGNS solver can accommodate a variety of models for the inviscid flow. In this work we will use both the full potential equation as well as the Euler equations. Both models are commonly used in external aerodynamics.

1. *Euler equations*

The Euler Eq. are derived from the Navier-Stokes Eq. by dropping the viscous fluxes $\mathbf{G}$ and read:

$$\frac{\partial \mathbf{u}_\mathrm{i}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{u}_\mathrm{i}) = 0. \tag{4}$$

Here, $\mathbf{u}_\mathrm{i}$ is a vector field for the same conserved variables as the original Navier-Stokes, this is, density, momentum, and total energy (see Eq. 2). Throughout this paper we will use the subscript $(\cdot)_\mathrm{i}$ to denote *inviscid* or EIF quantities. As in Navier-Stokes, the ideal gas law is used to relate density, temperature and pressure. In terms of the conserved variables, one can obtain pressure using the expression: $p_\mathrm{i} = (\gamma - 1)(\rho E_\mathrm{i} - \frac{1}{2}\rho_\mathrm{i}(\mathbf{v}_\mathrm{i} \cdot \mathbf{v}_\mathrm{i}))$.

The Euler equations lack a physical mechanism to set zero velocity at the wall. Instead, a non-penetration boundary condition needs to be used:

- Solid wall: $\mathbf{v}_\mathrm{i} \cdot \mathbf{n} = 0$, extrapolate ρ_i, $\mathbf{v}_\mathrm{i} \cdot \mathbf{t}$ and ρE_i.

We postpone the discussion of the coupling boundary condition between the RVF and the EIF to the next section.

2. *Full Potential equation*

The full potential equation is a simplification of the Euler equations where the flow is considered isentropic and irrotational. Under these conditions, a velocity potential can be introduced so that the velocity field can be related to the gradient of said potential, which in turn is related to the thermodynamic state of the fluid through the isentropic flow relationships. Here we will use this model for subsonic steady flows. The nonlinear PDE to solve reads:

$$\nabla \cdot (\rho_\mathrm{i}(\mathbf{v}_\mathrm{i})\mathbf{v}_\mathrm{i}) = 0, \tag{5}$$

$$\mathbf{v}_\mathrm{i} - \nabla \phi = 0, \tag{6}$$

where the density is given by:

$$\rho_\mathrm{i}(\mathbf{v}_\mathrm{i}) = \rho_\infty \left(1 + \frac{(\gamma - 1)M_\infty^2}{2}(1 - \mathbf{v}_\mathrm{i} \cdot \mathbf{v}_\mathrm{i})\right)^{1/(\gamma-1)}. \tag{7}$$

The most relevant boundary conditions for the full potential equation can be summarized as:

- Far-field: $\phi = \mathbf{v}_\infty \cdot \mathbf{x}, \quad |\mathbf{x}| \to \infty$,
- Solid Wall: $\nabla\phi \cdot \mathbf{n} = 0$,
- Inflow: $\nabla\phi \cdot \mathbf{n} = \mathbf{v}_\infty \cdot \mathbf{n}$,
- Parallel velocity outflow: $\phi = 0$.

The coupling with the viscous solver happens through a transpiration term that will be discussed in the next section. As we will see later, a recovery of the EIF state (density, specific momentum and specific total energy) into the RVF domain is required for coupling purposes. The first two follow directly from Eq. 6 and Eq. 7. The last one can be easily constructed using the isentropic relation for the pressure:

$$p_\mathrm{i} = p_\infty \left(\frac{\rho_\mathrm{i}}{\rho_\infty}\right)^\gamma, \tag{8}$$

and the definition of total energy:

$$\rho E_\mathrm{i} = \frac{p_\mathrm{i}}{\gamma - 1} + \frac{1}{2}\rho_\mathrm{i}(\mathbf{v}_\mathrm{i} \cdot \mathbf{v}_\mathrm{i}). \tag{9}$$

C. Coupling the RVF and the EIF

In principle, solving the EIF and RVF separately on overlapping domains introduces flexibility in terms of mesh generation for the boundary layer. This comes at the cost of a more complicated coupling to guarantee that both solutions coincide outside the shear layer.

1. Approximation of the EIF inside the RVF domain

A key factor in this coupling is the availability of a reasonable approximation of the EIF profile inside the viscous domain. For a fixed viscous mesh, one could come up with an interpolation scheme to do this only once. Unfortunately, we are interested in evolving the RVF mesh on the fly, which would require a re-evaluation at each iteration. To avoid this, we introduce a surrogate of the EIF denoted by $\tilde{\mathbf{u}}_\mathrm{i}$ that we hope is close enough to the actual EIF.

The simplest approach is to use a constant value for it across the boundary layer taken from the wall, $\tilde{\mathbf{u}}_\mathrm{i} = \mathbf{u}_\mathrm{i}|_\mathrm{wall}$. All the results presented in this work were computed this way. We expect this to be a very reasonable approximation when the boundary layer is thin or curvature is not too strong. Higher order surrogates would include reconstructions based on the gradients at the wall (available for the kind of Euler solvers we will use) or an irrotationality condition (when full potential is used). As we will see later in the results, some of the cases we present might benefit from this.

2. Boundary conditions for the EIF

To take into account the effect of the boundary layer on the external flow, certain boundary conditions need to be imposed at the wall. These conditions cannot be deduced directly from physical principles, since the EIF is not a physical solution inside the shear layer. In this work, we will use the well known mass transpiration analogy proposed by Lighthill,[12] that gives the required mass flux at the wall (or wake center) so that the EIF matches the RVF:

$$\rho \mathbf{v}_\mathrm{i} \cdot \mathbf{n}|_{\eta=0} = \frac{\partial}{\partial \alpha} \int_0^{\eta_\mathrm{edge}} (\rho \mathbf{v} - \widetilde{\rho \mathbf{v}}_\mathrm{i})_\alpha d\eta. \tag{10}$$

Here, (α, η) is a system of curvilinear coordinates for which $\eta = 0$ at the wall (see Figure 1).

In the case of the full potential equation, mass transpiration is enough to produce the desired effect. However, when the Euler equations are used, this is not the case anymore. Other researchers investigated the coupling[10, 13] and concluded that as long as the dividing streamline is right, little more is relevant since the EIF is fictitious and non-unique inside the shear layer. One can then take advantage of this and impose boundary conditions that do not penalize the numerics, this is, that produce smooth EIF profiles in the overlap region. In this work, we will use the following boundary conditions:[13]

- $\frac{\partial \mathbf{v}_\mathrm{i} \cdot \mathbf{t}}{\partial \eta} = 0$, $\frac{\partial H_\mathrm{i}}{\partial \eta} = 0$, extrapolate p.

3. Boundary Conditions for the RVF

The EIF computed using the previous boundary conditions is used as a driver for the viscous flow at the edge of the RVF domain. In particular, the following boundary conditions are imposed there:

$$\rho = \widetilde{\rho}_\mathrm{i}, \tag{11}$$

$$\rho \mathbf{v} \cdot \mathbf{t} = \widetilde{\rho \mathbf{v}}_\mathrm{i} \cdot \mathbf{t}, \tag{12}$$

$$H = \widetilde{H}_\mathrm{i}, \tag{13}$$

$$\rho \mathbf{v} \cdot \mathbf{n} = \text{free}, \tag{14}$$

where extrapolation of mass flux at the edge is required since the approximation of the EIF ($\tilde{\mathbf{u}}_\mathrm{i}$) inside the viscous domain does not exactly satisfy the governing equations.

D. Real Viscous Flow Geometry

The coupling conditions just presented ensure that the EIF and the RVF match outside the boundary layer, which in turn allows the RVF domain to be truncated a certain distance away from the wall. This introduces potential savings in computational cost provided the mesh is properly adapted to the solution, in which case the high order discretization quickly leverages its asymptotic convergence properties. Here, we will perform this adaptation on the fly by means of an evolution of the thickness of the boundary layer mesh, what we call the normal scaling δ.

Our proposed approach for this is to extrude the domain from the wall outwards following a predefined direction $\hat{\mathbf{n}}$ that does not necessarily need to coincide with the normal to the surface. The length of the extrusion is dictated by the field $\delta(\alpha)$, that lives over the surface of interest. The transformation from reference coordinates (α, η) to physical ones $\mathbf{x}$ is then given by an explicit mapping $G : (\alpha, \eta) \mapsto (x, y)$ defined as:

$$\mathbf{x} = \mathbf{g}_{\text{surf}}(\alpha) + \delta(\alpha) \times \eta \times \hat{\mathbf{n}}(\alpha), \tag{15}$$

where $\mathbf{g}_{\text{surf}}$ is the parametric representation of the wall.

Of the two parameters that define the extrusion, $\hat{\mathbf{n}}$ can be generated a priori based on the geometric features of the surface. This only leaves the normal scaling to be computed simultaneously with the flow. For this, δ will be evolved on the surface according to the following surface reaction-diffusion equation:[10]

$$\frac{\partial \delta}{\partial t} = \frac{\delta_{\text{spec}}(\mathbf{u}, \delta) - \delta}{\tau} + \Delta_\Gamma(\epsilon\delta), \tag{16}$$

where Δ_Γ is the Laplace-Beltrami operator[14] and the parameters τ and ϵ are constants that control the smoothness of the solution. For the cases computed here we set $\tau = \Delta t/5$ and $\mu = h^2/\tau$, where h is the surface element size. Notice the first quotient on the right hand side forces δ to follow the target value $\delta_{\text{spec}}(\mathbf{u}, \delta)$, which represents a measure of the thickness of the boundary layer. In this work we will use a well tested function proposed by Drela for 2D flows:[6]

$$\delta_{\text{spec}}(\mathbf{u}, \delta) = c_0 + c \times \left(\delta_k^* + \theta_k \left(3.15 + \frac{1.72}{H_k - 1}\right)\right). \tag{17}$$

Here, δ_k^* is the kinematic displacement thickness, θ_k is the kinematic momentum thickness and $H_k = \frac{\delta_k^*}{\theta_k}$ is the shape factor. The constants c_0 and c are set so that the solution is a few times bigger than the indicator alone and bounded away from zero. All the results presented here used $c_0 = 10^{-4}$ and $c = 2$. Notice this definition of the normal scaling target requires an integration across the boundary layer that can be easily performed thanks to the extruded character of the RVF domain. The boundary conditions for this equation are only required at the edges of the surface. For all the results presented here we used homogeneous Neumann conditions.

E. Arbitrary Lagrangean-Eulerian formulation

As it stands, our proposed formulation consists of three systems of equations (RVF, EIF and Normal Scaling) that have to be solved simultaneously. Of these, the RVF has to be solved on a mesh that will change implicitly with the solution at each iteration. To converge this very nonlinear problem, some kind of time relaxation will be required. This means the RVF is solved on a deformable domain in time which requires the use of an Arbitrary Lagrangean-Eulerian description (ALE) to enforce conservation and prevent unphysical states. Here we will loosely follow the ALE formulation introduced by Persson *et al.*[15]

To write the ALE form of our system, all we need is a mapping from our reference domain $(\mathbf{r})$ to the physical domain $(\mathbf{x})$: $\mathbf{x} = \mathcal{G}(\mathbf{r}, t)$, for which the gradients:

$$\mathbf{G} = \frac{\partial \mathcal{G}}{\partial \mathbf{r}}, \tag{18}$$

$$\mathbf{v}_G = \frac{\partial \mathcal{G}}{\partial t}, \tag{19}$$

are available. Given these, we can transform a general conservation law in the physical domain $\Omega_{\mathbf{x}}$:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{A}(\mathbf{u}, \mathbf{Q}) = \mathbf{s}, \tag{20}$$

$$\mathbf{Q} - \nabla \mathbf{u} = 0, \tag{21}$$

into an equivalent conservation law in the reference domain $\Omega_{\mathbf{r}}$:

$$\frac{\partial g\mathbf{u}}{\partial t} + \nabla_{\mathbf{r}} \cdot \left\{ g\mathbf{G}^{-1} \left(\mathbf{A}(\mathbf{u}, \mathbf{G}^{-1}\mathbf{Q}_{\mathbf{r}}) - \mathbf{u} \otimes \mathbf{v}_G \right) \right\} = g\mathbf{s}, \tag{22}$$

$$\mathbf{Q}_{\mathbf{r}} - \nabla_{\mathbf{r}}\mathbf{u} = 0, \tag{23}$$

where $g = \det(\mathbf{G})$ and the divergence and gradient operators work on the reference coordinates $\mathbf{r}$. In this work, the reference coordinates will be the curvilinear system (α, η) introduced before (see Figure 1). In the reference coordinates, the mesh is fixed and uniform as shown in Figure 2.

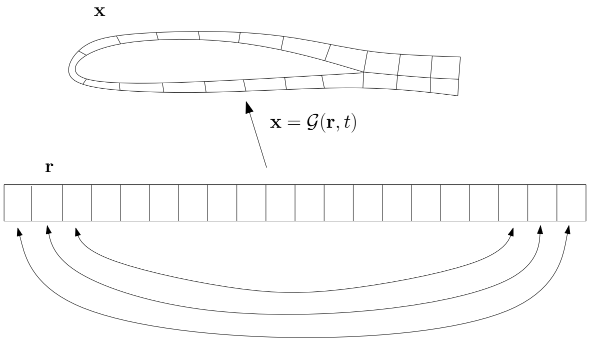


Figure 2: comparison between the physical and reference domains. Arrows denote extra connectivities at the assembly level to treat the wake.

In cases where the transient evolution is the subject of study (e.g. flapping flight), an extra condition is required to guarantee a fully conservative solution. This is known in the literature as the geometric conservation law.[15] Since our interest is steady state solutions, we did not implement it in our solver. Convergence of the cases shown later seems unaffected by this absence.

III. Discretization and solution

All the equations presented in the previous sections are discretized using a variety of Finite Element techniques. In this section, we will say a few words about them.

A. EIF Discretization

To discretize the EIF (Eq. 4 or Eqs. 5-6) we use the high order Hybridizable Discontinuous Galerkin scheme introduced by Peraire *et al.*[11] The HDG method requires two sets of unknowns: the approximation to the solution inside the elements (denoted by $\mathbf{u}_{h\mathrm{i}}$) and the numerical trace at the faces of the mesh (denoted by $\tilde{\mathbf{u}}_{h\mathrm{i}}$). Both $\mathbf{u}_{h\mathrm{i}}$ and $\tilde{\mathbf{u}}_{h\mathrm{i}}$ consist of piecewise polynomials of order k, discontinuous across their support (elements for $\mathbf{u}_{h\mathrm{i}}$ and faces for $\tilde{\mathbf{u}}_{h\mathrm{i}}$). In HDG, stability in the convective regime relies on a special form of the trace fluxes, that requires the so called stabilization matrix $\mathbf{A}_n$. For all the cases presented here, it was taken to be $\mathbf{A}_n = \frac{1}{M_\infty}\mathbf{I}$.

The EIF meshes used here were generated with Distmesh.[16] The EIF mesh is fixed in time, hence there is no need to reformulate the problem in ALE form.

Details on the implementation of HDG for the Euler and Navier-Stokes equations, including a discussion on boundary conditions and the choice of the stabilization matrix can be found in the literature.[11] For the case of the full potential solver, a similar discussion can be found for elliptic problems.[17]

B. RVF Discretization

To discretize the RVF system (Eq. 1-3) we use the same kind of HDG scheme. Given the deformable nature of the RVF mesh, and ALE reformulation is used to express the system in the reference mesh. Similar approximation spaces are introduced for the solution $\mathbf{u}_h$ and the trace $\tilde{\mathbf{u}}_h$. In the case of the RVF, the space of the solution is a tensor product of polynomials on the surface and 1D polynomials in the normal direction. We will take advantage of this property and use sum-factorization[18] to reduce the cost of the assembly. A local Lax-Friedrichs stabilization matrix is used, properly modified to account for the effects of the mapping.[11,15]

The surface mesh for the RVF, which is a discrete version of the surface parametrization $\mathbf{g}_{\text{surf}}(\alpha)$ (see Eq. 15) is extracted from the boundaries of the EIF mesh where viscous effects are required. The extrusion direction $\tilde{\mathbf{n}}$ at the vertices of the surface mesh is computed using an averaging of the normal vector between neighboring elements and propagated linearly into the element. This ensures the RVF mesh is conforming, even though it is never explicitly formed. Instead, the gradients of the mapping are evaluated and the ALE formulation is solved on a reference mesh that contains the right logic to deal with neighbors on the surface and across the wake (see Figure 2)

C. Normal Scaling discretization

The normal scaling equation is discretized on the surface using a Continuous Galerkin Surface Finite Element Method (SFEM).[14] The approximation space for the discrete normal scaling δ_h consists of continuous piecewise linear polynomials on the surface that defines the wall or wake. The continuous piecewise linear approximation fulfills several requirements: it produces conforming meshes since δ_h is continuous, allows for a simple check of unphysical meshes ($\delta_h < 0$) and introduces very little new unknowns.

In this work we will deviate from the original method by Dziuk[14] and use a high order representation of the surface geometry where the PDE is solved. This does not introduce extra complexity or cost since a high order surface mesh was already extracted to generate the RVF geometry.

D. Solution procedure

The discretization of the three models (EIF, RVF and Normal Scaling) is carried out on a system of grids that is conformal only at the surface. This is depicted in Figure 3. The result of this discretization is a coupled system of nonlinear differential-algebraic equations in time. To time-march this system towards steady state, a suitable discretization of the time derivatives is required. Given our interest in steady state solutions, we are willing to sacrifice time accuracy for numerical stability. For this reason, a backwards Euler scheme (BDF1) is used here.

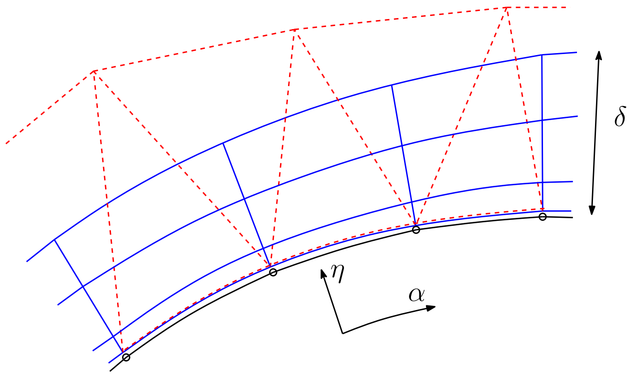


Figure 3: EIF (- - - -), RVF (——) and Normal scaling (-o-) meshes used to discretize the problem. All the meshes conform at the wall.

At each time-step of BDF1, a system of nonlinear algebraic equations needs to be solved. For this, we use Newton's method combined with a backtracking line-search to select the update step.[19] In this work, the Jacobian required by the Newton iteration is computed explicitly and enters a linear system that is solved using sparse direct techniques.

To initialize the flow, first the EIF is solved alone without interaction. Then, the RVF is initialized following the EIF on the surface and assuming no variation in η. Last, the normal scaling is initialized using an analogy to the problem of an impulsive plate so that $\delta_h = \mathcal{O}(\sqrt{\Delta t / Re})$. At this point the time-stepping is started. The first time step can be hard to converge due to the strong interaction between the normal scaling and the RVF solution. We have found that freezing the normal scaling evolution for the first time-step prevents this from happening. The time step Δt is multiplied or divided by a fixed fraction (typically 2) depending on the number of Newton iterations required to converge the current step. A final steady state solution is attempted once Δt is big enough or the total accumulated time is greater than a set value (usually greater than the convective time for the length of the domain).

IV. Results

In this section we present the cases used to verify the implementation of VGNS, which will be performed in two stages. In the first one, we present results for the RVF solver alone. Both steady and unsteady cases will be used to asses the proper implementation of the ALE formulation through grid convergence studies. In the second stage, we will apply the whole viscous-inviscid solver to a variety of problems and compare it against a full Navier-Stokes discretization as well as an integral boundary layer method.

A. Verification of the RVF solver: Convection-Diffusion

The first case we would like to discuss is that of the linear convection-diffusion transport of a scalar quantity released steadily at a point $\mathbf{x}_0$. We will use this case to assess the implementation of the ALE formulation and the RVF solver. The PDE that governs this problem reads:

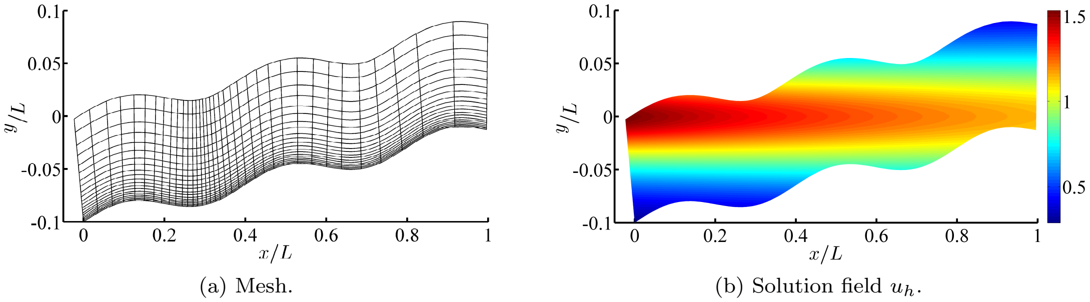
$$\frac{\partial u}{\partial x} - D_x \frac{\partial^2 u}{\partial x^2} - D_y \frac{\partial^2 u}{\partial y^2} = \delta(\mathbf{x} - \mathbf{x}_0), \tag{24}$$

where D_x and D_y are the diffusion coefficients in the x and y direction respectively. In the case of an infinite domain, an analytical solution exists:[20]

$$u = \exp\left(\frac{x - x_0}{2D_x}\right) K_0 \left(\frac{1}{2}\sqrt{\left(\frac{x - x_0}{D_x}\right)^2 + \frac{y^2}{D_x D_y}}\right), \tag{25}$$

where K_0 is the modified Bessel function of the second kind and zero-th order. We can use this result to impose Dirichlet boundary conditions on a truncated deformable domain where we solve the ALE version of Eq.24. By comparison with the analytical solution, we can easily compute the error in our solver.

First, we present a grid convergence study for the steady problem with parameters $D_x = 1$ and $D_y = 10^{-3}$. In these cases the normal scaling was held fixed $\frac{\partial \delta}{\partial t} = 0$. The sequence of meshes was obtained by uniform refinement of the initial coarse mesh. An example of mesh and solution field is shown in Figure 4. The computed errors for different polynomial approximation orders ($k = 1$ to $k = 4$) are summarized in Figure 5. We experience optimal $k + 1$ convergence rates in the solution field u_h and the gradient $\mathbf{q}_h$.



(a) Mesh.

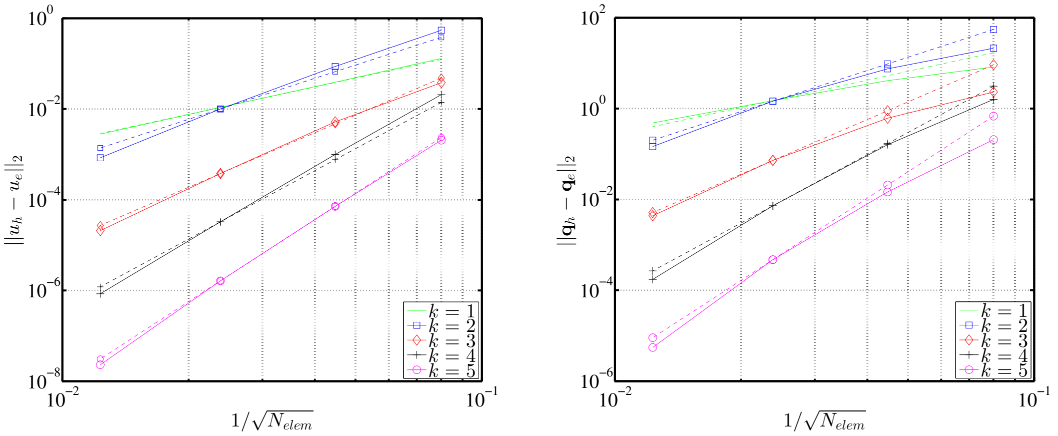
(b) Solution field u_h.

Figure 4: sample mesh (left) and solution (right) for the steady convection-diffusion convergence study.

Next, we would like to asses the error in the unsteady ALE formulation. To do this, we prescribed δ_{spec}:

$$\delta_{\text{spec}} = 0.1 + 0.05 \sin\left(\frac{\pi}{2}(x - t)\right), \tag{26}$$

and run the problem on the same surface mesh as in the steady case using polynomials of order $k = 3$. The Dirichlet boundary conditions were set with the help of Eq. 25, effectively restricting the free-space solution to the computational domain even though this deformed in time. The evolution of the error with time is

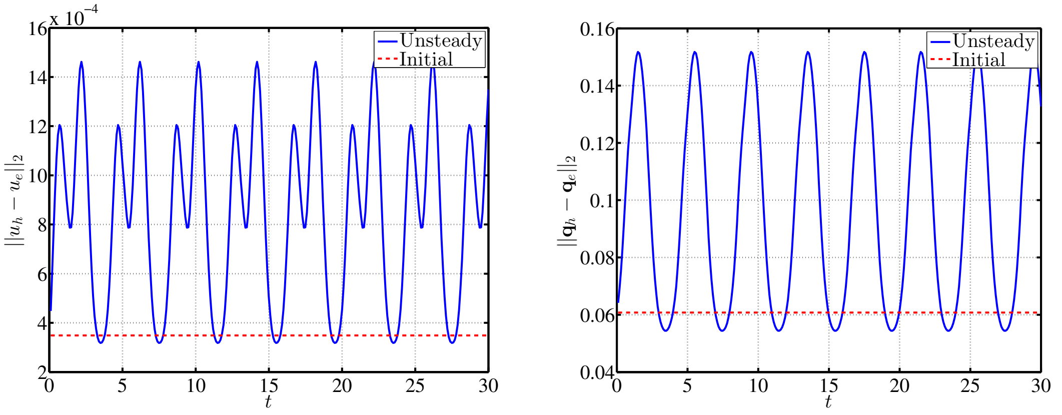


(a) Error in the solution u_h compared to asymptotic estimates ($\mathcal{O}(h^{k+1})$).

(b) Error in the gradient $\mathbf{q}_h$ compared to asymptotic estimates ($\mathcal{O}(h^{k+1})$).

Figure 5: grid convergence studies for the convection-diffusion problem. Solid lines (——) denote computed error. Dashed lines (-----) denote asymptotic error estimate. The convergence is optimal for the solution as well as for the gradient.

plotted in Figure 6. As we can see, the error grows with respect to the initial value but is bounded in time. We would expect the geometric conservation law to slightly change this result, however, given our interest in steady state solutions, we will consider the current implementation good enough to accommodate transients in the solution.



(a) Error in the solution u_h.

(b) Error in the gradient of the solution $\mathbf{q}_h$.

Figure 6: evolution of the error in the solution as a function of time, compared to the initial error. Time step in this case was set to $\Delta t = 0.1$.

B. Verification of the RVF solver: compressible Navier-Stokes

Next we would like to present a similar convergence study for the compressible Navier-Stokes equations. For this we require an analytical solution that we obtain using the method of manufactured solutions.[21] We prescribe the solution field to be:[11]

$$p = p_\infty = \frac{1}{\gamma M_\infty^2}, \tag{27}$$

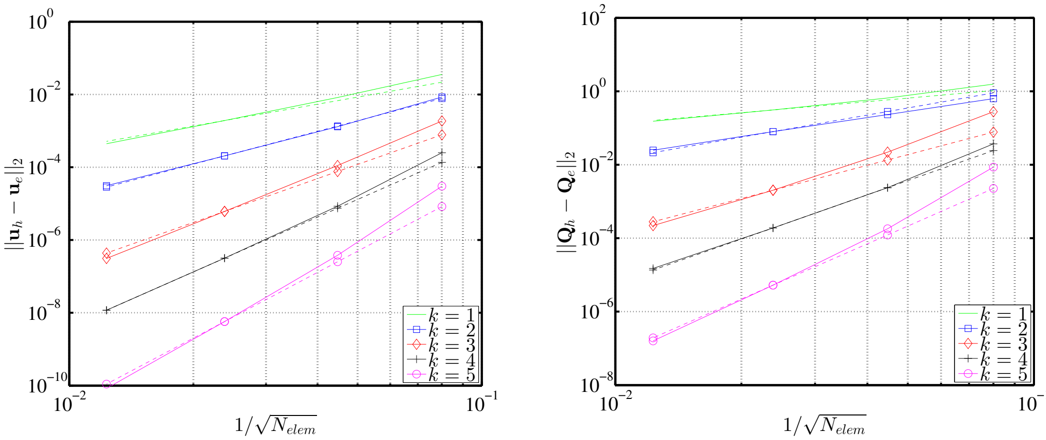
$$T = T_0 + \frac{y}{H}(T_1 - T_0) + \frac{(\gamma-1)M_\infty Pr}{2}\frac{y}{H}\left(1 - \frac{y}{H}\right), \tag{28}$$

$$u = \frac{y}{H}\log(\left(1 + \frac{y}{H}\right), \tag{29}$$

$$v = 0, \tag{30}$$

with parameters: $T_0 = 0.8$, $T_1 = 0.85$, $H = 0.2$, $Pr = 0.72$ and $M_\infty = 0.15$. Using this solution we can reconstruct the state and fluxes in Eqs. 2-3. Filtering these through the Navier-Stokes operator (Eq. 1) yields an equivalent source term. This can easily be done using a symbolic manipulator.

Using the equivalent source term and Eqs. 27 -30 as Dirichlet boundary conditions we can easily asses the error of the scheme. Here we will focus on the steady case ($\frac{\partial\delta}{\partial t} = 0$) on a domain similar to the one used in the previous example (see Figure 4a). We followed the same strategy as the convection diffusion case, this is, uniform refinement from an initial mesh and a sweep in polynomial order. The results are summarized in Figure 7 and show optimal convergence rates in the solution but not in the gradient. This is consistent with the results found in the literature for this same case.[11]



(a) Error in the solution $\mathbf{u}_h$ compared to asymptotic estimates ($\mathcal{O}(h^{k+1})$).

(b) Error in the gradient $\mathbf{Q}_h$ compared to asymptotic estimates ($\mathcal{O}(h^k)$).

Figure 7: grid convergence studies for the Navier-Stokes problem. Solid lines (−) denote computed error. Dashed lines (−−) denote an asymptotic error estimate. The convergence is optimal for the solution but not for the gradient.

C. Laminar flow over smooth bump

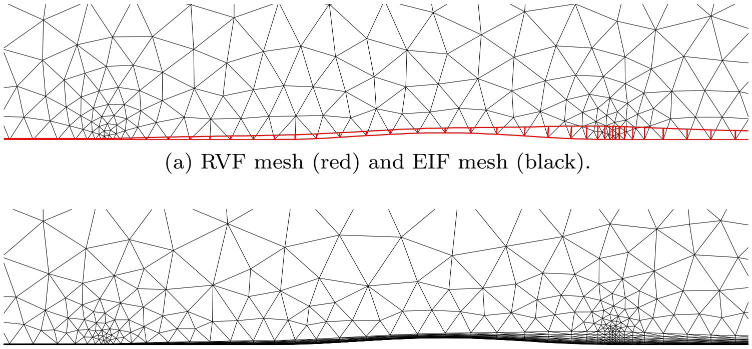
VGNS belongs to the class of strongly coupled viscous-inviscid Interaction schemes, which are known to be able to solve flows past separation. We would like to asses this using a case that develops a laminar separation bubble. More precisely, we are concerned with the laminar flow at Reynolds number $Re = 3 \cdot 10^5$ and Mach number $M_\infty = 0.3$ over a smooth bump protruding from a semi-infinite flat plate. The height of the bump h is given by:

$$h = 0.02\sin^2(\pi(x - 0.5)), \qquad 0.5 < x < 1.5. \tag{31}$$

The leading edge of the flat plate is placed at $x = 0$, half a reference length ahead of the bump. In this case we will compare VGNS, using both full-potential and Euler for the EIF, against two other techniques: a full Navier-stokes solver and Xfoil[22] (panel method coupled to integral boundary layer).

To compute the full Navier-Stokes solutions, we first had to generate a suitable mesh. For this, we took the RVF mesh that VGNS delivers, broke it in into triangles, and filled in the rest of the external domain with an unstructured triangulation using Distmesh (see Figure 8). With this mesh at hand, we solved the Navier-Stokes equations using the EIF solver of Section A with the right viscous fluxes.

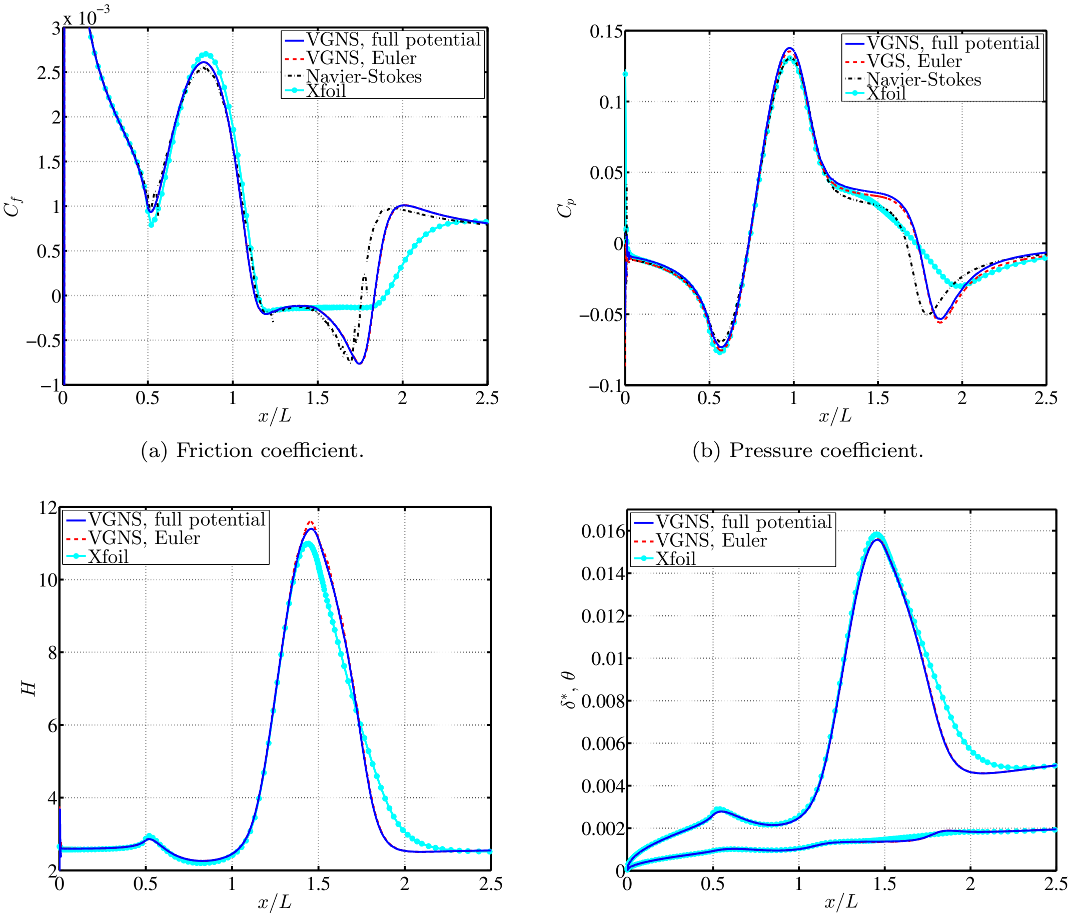
For the Xfoil solution, we only needed to define the panel distribution. In this case, we took this from our high order surface mesh. This way we ensured the number of nodes (and hence the resolution) on the surface was the same. To prevent transition in Xfoil, the critical amplification factor was set to $N_{\text{crit}} = 100$.



(a) RVF mesh (red) and EIF mesh (black).

(b) full Navier-Stokes mesh.

Figure 8: meshes used for the computation of the laminar bump case. Top, EIF mesh (black) and extrusion of the RVF mesh (red) for the converged solution. Bottom, full Navier-Stokes mesh generated from VGNS.



(a) Friction coefficient.

(b) Pressure coefficient.

(c) Boundary layer shape parameter.

(d) Displacement and momentum thicknesses.

Figure 9: quantities of interest at the wall for the laminar bump case.

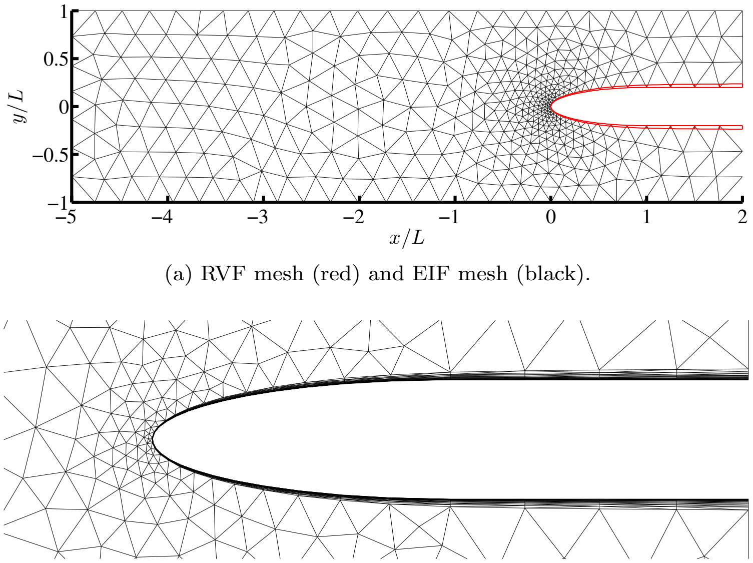
The results for this case using polynomials of order $k = 3$ (4th order accurate scheme) and 10 elements across the boundary layer are plotted in Figure 9. Notice how the different EIF models (Euler or full potential) barely change the solution. The comparison with full Navier-Stokes in C_p and C_f is very good and much more accurate than the integral boundary layer results. This seems to indicate our segregated approach retains high fidelity despite the simplifications it involves.

D. Laminar stagnation flow

One of the peculiarities of VGNS is the use of the Navier-Stokes equations for the RVF instead of thin shear layer approximations. Amongst other things, this means the RVF model is valid at stagnation points, which suggests no boundary layer initialization is required. The goal of the next case we present is to confirm this property.

We are interested in the flow over a body defined by a semi-ellipse of aspect ratio $b/a = 0.2$ that continues into a constant section of radius equal to the minor axis. The flow around this body is constrained by two flat inviscid walls that effectively force a favorable pressure gradient, preventing separation of the boundary layer downstream of the semi-ellipse. The flow parameters for this case are $M_{\text{inlet}} = 0.3$ and $Re = 10^5$.

We solved the problem using both the VGNS approach as well as a full Navier-Stokes solver. As in the previous case, the mesh for the full Navier-Stokes problem was constructed based on the VGNS one (see Figure 10). The latter was generated using Distmesh, which places the nodes randomly on the geometry unless instructed otherwise. This ensures general stagnation flows can be computed since no alignment or special distribution is enforced. For this case, the RVF mesh consisted of 10 elements across the boundary layer.



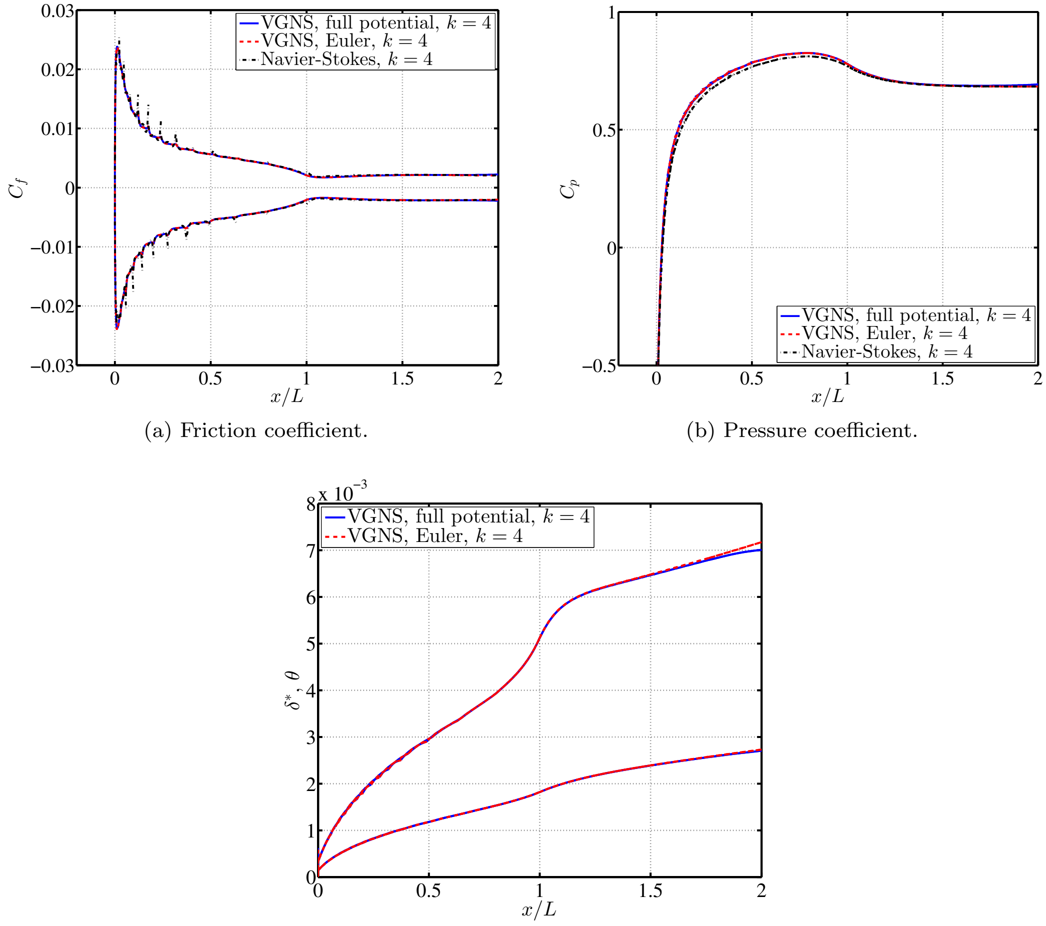
(a) RVF mesh (red) and EIF mesh (black).

(b) Detail of the full Navier-Stokes mesh.

Figure 10: meshes used for the computation of the stagnation flow case. Top, EIF mesh (black) and extrusion of the RVF mesh (red) for the converged solution. Bottom, full Navier-Stokes mesh generated from VGNS.

The results are summarized in Figure 11. As we can see there, the match between VGNS and the full Navier-Stokes solution is very good, especially for the friction coefficient. Notice how the choice of inviscid model for the EIF does not affect the solution, thus, in a practical application to subsonic flows, we would favor the full potential model for cost reasons.

Our only concern with these results are the obvious oscillations in the friction coefficient and the pressure (to a lesser extent) that affect both the full Navier-Stokes solution as well as the VGNS one. Other researchers have found similar problems with similar schemes.[23] We believe these might be related with the high order meshes, in particular, the fact that the tangents to the surface are not enforced to be continuous across elements. We are currently investigating this issue.



(a) Friction coefficient.

(b) Pressure coefficient.

(c) Displacement and momentum thicknesses.

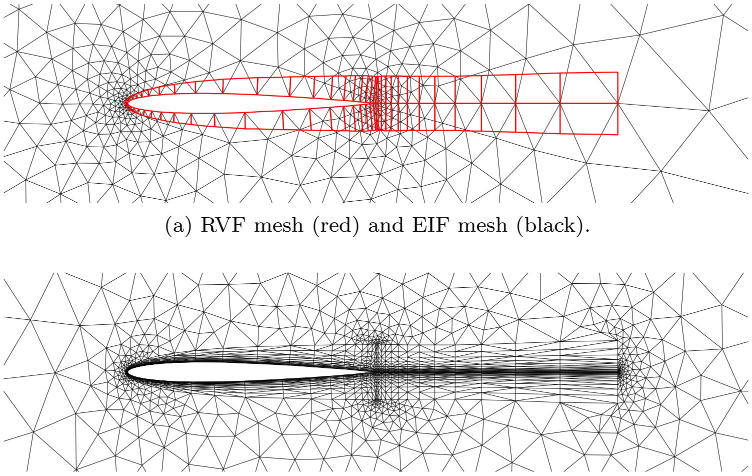
Figure 11: quantities of interest at the wall for the stagnation flow case.

E. Laminar flow over NACA 0008 airfoil

The last case we would like to discuss concerns the flow around a non-lifting airfoil. This case represents our first attempt at solving a truly aerodynamic flow in this paper. Unlike the previous cases, that isolated effects, this example presents at the same time several different flow structures such as stagnation flow, trailing edge separation and a wake, hence, it is a stepping stone for more complicated lifting cases.

The parameters for this problem are the following: $Re = 10^4$, $\alpha = 0$ and $M_\infty = 0.1$. We solved this flow using VGNS with full potential for the EIF and full Navier-Stokes, both with polynomials of order $k = 3$. For comparison purposes, the same case was solved using Xfoil. The initial mesh for the VGNS solver was generated with Distmesh and the rest of the meshes followed from it as in the previous examples (see Figure 12). The RVF mesh consisted of 10 elements in the normal direction.

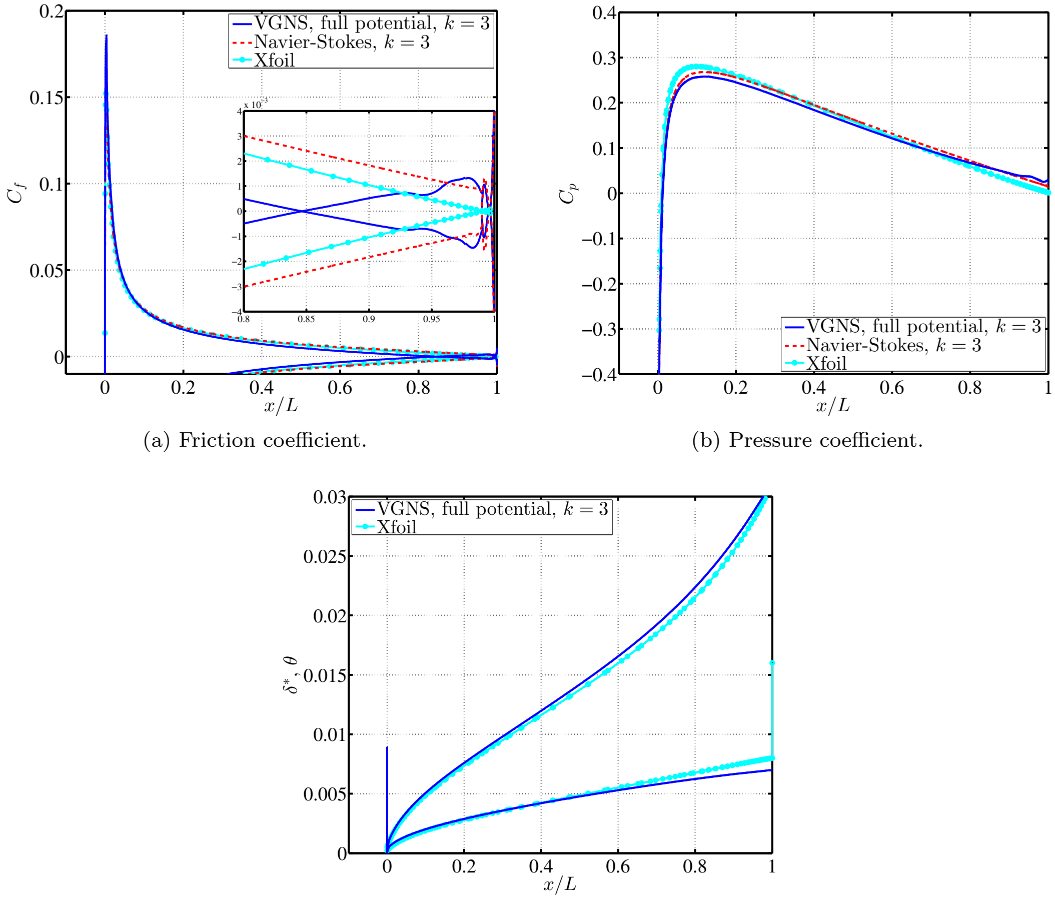
The results are summarized in Figure 13 and show how VGNS agrees well with Navier-Stokes and Xfoil. The biggest discrepancy in friction coefficient is found at the trailing edge, where VGNS predicts a small amount of separation that none of the others show. The differences in pressure coefficient are generally small too, but seem to indicate VGNS is closer to the Navier-Stokes solution.



(a) RVF mesh (red) and EIF mesh (black).

(b) full Navier-Stokes mesh.

Figure 12: meshes used for the computation of the laminar flow around a NACA 0008 airfoil. Top, EIF mesh (black) and extrusion of the RVF mesh (red) for the converged solution. Bottom, full Navier-Stokes mesh generated from VGNS.



(a) Friction coefficient.

(b) Pressure coefficient.

(c) Displacement and momentum thicknesses.

Figure 13: quantities of interest at the wall for the laminar flow around a NACA 0008 airfoil.

V. Conclusions and future work

We have presented a new strategy for the solution of boundary layer flows based on a segregated approach in which viscous effects are solved on a truncated domain that adapts automatically to the thickness of the boundary layer. We have applied this strategy to a variety of cases and verified it against solutions computed using the full Navier-Stokes equations as well as Xfoil. The results indicate that our approach retains the high-fidelity of full Navier-Stokes solvers and greatly simplifies the process of generating adapted meshes for the boundary layer.

In this work we only focused on laminar flows, mostly because a high fidelity solution using the Navier-Stokes equations can be computed for comparison. In some of the cases presented here, transition would have naturally occurred which would have had required some sort of turbulence model. We decided not to include any of it, even though fully turbulent solutions have already been computed within the VGNS framework.[9] We are currently interested in taking advantage of the extruded viscous mesh to extract boundary layer parameters and build a transition prediction capability.

VI. Aknowledgments

N. C. Nguyen and J. Peraire would like to acknowledge the Singapore-MIT Alliance and the Air Force Office of Scientific Research under the grant FA9550-12-0357 for partially supporting this work. M. Drela would like to aknowledge the support of The Boeing Company (technical supervisors David P. Young and Mori Mani). D. Moro would like to acknowledge the support of LaCaixa Foundation and The Boeing Company. Special thanks to Xevi Roca for many fruitful conversations, suggestions and comments.

References

[1] Johnson, F. T., Tinoco, E. N., and Yu, N. J., "Thirty years of development and application of CFD at Boeing Commercial Airplanes, Seattle," *Computers & Fluids*, Vol. 34, No. 10, 2005, pp. 1115–1151.

[2] Fidkowski, K. J. and Darmofal, D. L., "A triangular cut–cell adaptive method for high–order discretizations of the compressible Navier–Stokes equations," *Journal of Computational Physics*, Vol. 225, 2007, pp. 1653–1672.

[3] Persson, P. O. and Peraire, J., "Curved mesh generation and mesh refinement using Lagrangian solid mechanics," *47th AIAA Aerospace Sciences Meeting and Exhibit*, 2008.

[4] Moro, D., Nguyen, N. C., and Peraire, J., "Navier-Stokes solution using Hybridizable Discontinuous Galerkin methods," *20th AIAA Computational Fluid Dynamics Conference*, 2011.

[5] Allmaras, S. R., Bussoletti, J. E., Hilmes, C. L., Johnson, F. T., Melvin, R. G., Tinoco, E. N., Venkatakrishnan, V., Wigton, L. B., and Young, D. P., "Algorithm Issues and Challenges Associated with the Development of Robust CFD Codes," *Variational Analysis and Aerospace Engineering*, Springer New York, 2009.

[6] Drela, M., *Two-dimensional transonic aerodynamic design and analysis using the Euler equations*, Ph.D. thesis, Massachusetts Institute of Technology, 1986.

[7] Mughal, B. and Drela, M., "A calculation method for the three-dimensional boundary-layer equations in integral form (AIAA Paper 93-0786)," *31st Aerospace Sciences Meeting*, 1993.

[8] Nishida, B. A. and Drela, M., *Fully simultaneous coupling of the full potential equation and the integral boundary layer equations in three dimensions*, Ph.D. thesis, Massachusetts Institute of Technology, 1996.

[9] Moro, D., Nguyen, N. C., Drela, M., and Peraire, J., "A High Order Self-Adaptive Monolithic Solver for Viscous-Inviscid Interacting flows," *51st AIAA Aerospace Sciences Meeting*, 2013.

[10] Allmaras, S. R., *A coupled Euler/Navier-Stokes algorithm for 2-D unsteady transonic shock/boundary-layer interaction*, Ph.D. thesis, Massachusetts Institute of Technology, 1989.

[11] Peraire, J., Nguyen, N. C., and Cockburn, B., "A hybridizable discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations," *48th AIAA Aerospace Sciences Meeting and Exhibit*, Orlando, FL, 2010.

[12] Lighthill, M. J., "On displacement thickness," *Journal of Fluid Mechanics*, Vol. 4, No. 4, 1958, pp. 383–392.

[13] Murman, E. M. and Bussing, T. R. A., "On the Coupling of Boundary-Layer and Euler Equation Solutions," *Numerical and Physical Aspect of Aerodynamic Flows II*, 1984.

[14] Dziuk, G. and Elliott, C. M., "Surface finite elements for parabolic equations," *Journal of Computational Mathematics*, Vol. 25, No. 4, 2007, pp. 385–407.

[15] Persson, P. O., Bonet, J., and Peraire, J., "Discontinuous Galerkin solution of the Navier-Stokes equations on deformable domains," *Computer Methods in Applied Mechanics and Engineering*, Vol. 198, No. 17-20, 2009, pp. 1585–1595.

[16] Persson, P. O. and Strang, G., "A simple mesh generator in MATLAB," *SIAM review*, Vol. 46, No. 2, 2004, pp. 329–345.

[17] Nguyen, N. C., Peraire, J., and Cockburn, B., "An implicit high-order hybridizable Discontinuous Galerkin method for nonlinear convection-diffusion equations," *Journal of Computational Physics*, Vol. 228, No. 23, 2009, pp. 8841–8855.

[18] Vos, P. E., Sherwin, S. J., and Kirby, R. M., "From h to p efficiently: Implementing finite and spectral/hp element methods to achieve optimal performance for low- and high-order discretisations," *Journal of Computational Physics*, Vol. 229, No. 13, July 2010, pp. 5161–5181.

[19] Nocedal, J. and Wright, S. J., *Numerical optimization*, Springer series in operations research, Springer, 1999.

[20] Franz, S. and Kopteva, N., "Greens function estimates for a singularly perturbed convection-diffusion problem in three dimensions," *Iinternational Journal of Numerical Analysis and Modeling*, Vol. 1, No. 1, 2011, pp. 1–18.

[21] Steinberg, S. and Roache, P. J., "Symbolic manipulation and computational fluid dynamics," *Journal of Computational Physics*, Vol. 57, No. 2, Jan. 1985, pp. 251–284.

[22] Drela, M., "XFOIL: An analysis and design system for low Reynolds number airfoils," *Low Reynolds Number Aerodynamics*, edited by T. J. Mueller, No. 54 in Lecture Notes in Engineering, Springer-Verlag, 1989.

[23] Yano, M., *An Optimization Framework for Adaptive Higher-Order Discretizations of Partial Differential Equations on Anisotropic Simplex Meshes*, Ph.D. thesis, Massachusetts Institute of Technology, 2012.