Adjoint-based Adaptive Finite Element Method For The Compressible Euler Equations Using Finite Calculus

Mohammad Kouhi, Eugenio Oñate and Dimitri Mavriplis

October 9, 2014

Abstract

In this paper, an adjoint-based error estimation and mesh adaptation framework is developed for the compressible inviscid flows. The algorithm employs the Finite Calculus (FIC) scheme for the numerical solution of the flow and discrete adjoint equations in the context of the Galerkin finite element method (FEM) on triangular grids. The FIC scheme treats the instabilities normally generated in the numerical solution of the fluid equations through adding two stabilization terms, called streamline term and transverse term, to the original central-based discretized formulation. The non-linear system of equations resulting from the flow problem is solved implicitly using a damped Newton's method accompanied with the exact Jacobian matrix. A defect corrected scheme is implemented to iteratively solve the linear system of equations related to the adjoint problem benefiting from the transpose of the Jacobian matrix. At each iteration, the linear systems of equations resulting from the fluid and adjoint problems are solved using a preconditioned GMRES method. Having calculated the error of a specified output functional locally, an h-refinement methodology based on the element subdivision is performed to refine the candidate elements. The quality of the numerical results proves the capability of the presented approach for the adjoint-based error estimation and mesh adaptation problems in different flow regimes.

Keywords: Compressible Euler Equations; Stabilized Finite Element; Finite Calculus; Discrete Adjoint; h-refinement

1 Introduction

Adaptive mesh refinement (AMR) is one of the most efficient means for reducing the overall computational cost in the numerical solution of engineering fluid problems modeled by the compressible Euler and Navier-Stokes equations [1–4]. The basic idea behind AMR is the control of the mesh resolution by generating an appropriate fine mesh near the zones where the solution error is high and assigning a coarse mesh to the rest of the domain. These techniques are able to improve the accuracy of the flow solution around the high-error zones such as boundary layers, stagnation points and shock waves. This effect of AMR is more remarkable when the numerical solution of the fluid flow is to be employed in an optimization problem [5] where an appropriate evaluation of a practical output function such as the lift and drag coefficients becomes important.

The main components of any AMR technique are a reliable error estimator/indicator and a mesh refinement methodology. The error estimator/indicator introduces areas from the current mesh where refinement is needed whereas the enhancement of the current mesh in these areas through adding new elements is the task of the mesh refinement methodology. Based on the so called feature-based methods, one can consider the flow gradients [4, 6] or flow curvatures [3, 7, 8] as the error indicator for predicting the areas where the refinement is needed. In these areas, the fluid flow mostly has some complex features such as shock waves, boundary layers and stagnation points. Although this family of error indicators can predict the flow features accurately, it does not necessarily provide an efficient estimation of the practical engineering outputs (such as lift and drag) used widely during the design optimization process.

In order to resolve this drawback, another family of error estimators/indicators, called output-based methods, has been developed recently employing the sensitivities of a specified output functional with respect to the flow solution where these sensitivities are predicted through the

implementation of the adjoint variables. The general idea of these methods is to estimate the global error of the output functional as an inner product of the flow residuals and the adjoint variables a posteriori. For this end, two sets of problems, namely the flow problem and the adjoint problem, need to be solved on the current coarse mesh. The numerical solution of the flow equations provides the flow variables, whereas the adjoint variables are evaluated through the numerical solution of the adjoint equations.

Besides the application of output-based methods for estimating the global error, they can be considered as a local error indicator to find the zones where the functional error is more than a desirable tolerance. The application of the output-based error estimation and mesh adaptation methods using adjoint variables is studied by Venditti and Darmofal [9, 10], Pierce and Giles [11] and Nemec et al. [12] in the context of finite volume, whereas Becker and Rannacher [13], Rannacher [14] and Giles et al. [15] developed this technique for the finite element discretization method. Based on the recent developments of the discontinuous Galerkin finite element method in fluid problems, several implementations of the adjoint-based error estimation have been presented for inviscid [16, 17] and turbulent viscous [18] flows.

In order to solve the adjoint equations two different procedures, namely the continuous formulation and the discrete formulation, have been developed by researchers. In the continuous formulation the continuous form of the governing flow equations is first differentiated and then discretized, whereas the discrete formulation directly differentiates the discretized form of the governing equations. Although the continuous adjoint formulation has the advantage of less memory requirements, the implementation of the discrete formulation has been increased recently due to the simplicity of implementation and the unique capability of providing the exact discrete sensitivities. This simplicity comes from the fact that the transpose of the global Jacobian matrix, already calculated for the implicit solution of the flow equation, is utilized directly for the solution of the linear system of equations which arise from the discrete adjoint formulation. On the other hand, the treatment of the boundary conditions is quite straightforward in the case of the discrete formulation. A comprehensive comparison of the continuous and discrete adjoint formulations is presented in [19].

Regarding any central-based discretized formulation employed for the flow equations, it is a well established fact that the addition of stabilization terms to the original system of equations resulting from the discretization of the flow problem is essential to avoid the occurrence of numerical instabilities [20, 21]. Within the family of the stabilization techniques, the so called Finite Calculus (FIC) method has been successfully implemented for the stabilization of advective-diffusive transport and incompressible fluid flow problems [22–27]. Recently, a FIC-based stabilized formulation for the numerical solution of the compressible Euler and Navier-Stokes equations has been proposed in the context of Galerkin FEM by Kouhi and Oñate [28, 29]. Since the systems of equations obtained from both the flow and the adjoint problems contain the same eigenvalues, the stabilization techniques developed for the flow equations can be implemented for solving the adjoint problem as well.

In this article, we extend the implementation of the FIC-FEM stabilized formulation presented in [28] to develop an adjoint-based error estimation and mesh adaptation framework for compressible inviscid flows. The system of equations obtained from the discretization of the flow problem is solved implicitly using a damped Newton's method benefiting from the exact Jacobian matrix proposed in [29]. At each iteration step, the inherent linear system resulting from the flow equations is solved with a preconditioned GMRES method. The transpose of the Jacobian matrix, already constructed from the discretized flow equations containing the FIC-based stabilization terms, is employed for the solution of the adjoint equations.

AMR is performed here by using the local contributions of the functional error in conjunction with the classical h-refinement methodology, where each candidate element is divided into four by dividing each edge of the element into two. In order to demonstrate the capability of the FIC-FEM stabilized formulation in output-based error estimation problems, several examples are presented. By studying the quality of the results, it is found that the presented stabilized formulation provides enough stability for the numerical solution of the adjoint equations yielding an accurate estimation of the functional error during the AMR process.

The layout of the paper is the following: In Section 2 the compressible Euler equations along with the FIC-FEM stabilized formulation are described. Section 3 presents the derivation of the

output-based error estimation and adaptive mesh refinement method using adjoint variables. The solution strategies for the flow and adjoint equations are explained in Section 4. The numerical results corresponding to the proposed error estimation and mesh refinement strategy for different output functional in subsonic, transonic and supersonic flow regimes are shown in Section 5. Finally, conclusions and general remarks are summarized in Section 6.

2 Flow Problem Formulation

2.1 Governing Equations

The two-dimensional (2D) compressible Euler equations, including the mass balance, momentum and energy equations, are considered in this work and can be written in the following conservative form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} = \mathbf{0} \ \text{ for } \ i = 1, 2 \tag{1}$$

where $\mathbf{U}$ and $\mathbf{F}$ are the vectors of conservative variables and inviscid fluxes, respectively, which can be expressed as

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho e \end{bmatrix} \qquad \mathbf{F}_i = \begin{bmatrix} \rho v_i \\ \rho v_1 v_i + p \delta_{i1} \\ \rho v_2 v_i + p \delta_{i2} \\ v_i (p + \rho e) \end{bmatrix} \tag{2}$$

where ρ, $\mathbf{v}$, p and e are the density, the velocity vector, the static pressure and the total internal energy per unit mass, respectively. In the above equations $i, j = 1, n_d$ where n_d is the number of space dimensions ($n_d = 2$ for 2D flow problems) and δ_{ij} is the Kronecker delta.

By defining R as the ideal gas constant and neglecting chemical reactions, the equation of state for an ideal gas has the following form

$$p = R\rho T \tag{3}$$

where T is the temperature of the gas that can be obtained from

$$T = \frac{\gamma - 1}{R}(e - 0.5 v_j v_j) \tag{4}$$

where γ is the ratio of specific heats.

The standard sum convention for terms with repeated indices is adopted in the paper, unless otherwise specified.

2.2 Derivation of the Stabilized Formulation

We define r_d, r_{m_i} and r_e as the differential form of the mass balance equation, the ith momentum equation and the energy equation, respectively, as

$$r_d := \frac{\partial \rho}{\partial t} + \frac{\partial(\rho v_j)}{\partial x_j} \tag{5}$$

$$r_{m_i} := \frac{\partial(\rho v_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho v_i v_j) + \frac{\partial p}{\partial x_i} \tag{6}$$

$$r_e := \frac{\partial(\rho e)}{\partial t} + \frac{\partial}{\partial x_j}(v_j(\rho e + p)) \tag{7}$$

with $i, j = 1, n_d$. The stabilized formulation of the Euler equations based on the FIC scheme, presented in [28, 29], is obtained as

$$r_d - \frac{1}{2}(1 - \beta)\ell \ \text{sgn}(r_d)\frac{\boldsymbol{\nabla}\rho}{|\boldsymbol{\nabla}\rho|}.\boldsymbol{\nabla}r_d - \frac{1}{2}\beta\ell\frac{1}{|\mathbf{v}| + v_c}\boldsymbol{\nabla}.(\boldsymbol{\nabla}.\mathbf{F}_{m_i}) = 0 \tag{8}$$

$$r_{mi} - \frac{1}{2}(1 - \beta)\ell \ \text{sgn}(r_{mi})\frac{\boldsymbol{\nabla}v_i}{|\boldsymbol{\nabla}v_i|}.\boldsymbol{\nabla}r_{mi} - \frac{1}{2}\beta\ell\frac{\mathbf{v}}{|\mathbf{v}| + v_c}.\boldsymbol{\nabla}r_{mi} = 0 \tag{9}$$

3

$$r_e - \frac{1}{2}(1-\beta)\ell \, \mathrm{sgn}(r_e)\frac{\nabla T}{|\nabla T|}.\nabla r_e - \frac{1}{2}\beta\ell\frac{\mathbf{v}}{|\mathbf{v}|+v_c}.\nabla r_e = 0 \tag{10}$$

where β is a constant coefficient ranging between zero and one, ℓ is the characteristic element size, sgn(.) denotes the sign function, $|\mathbf{v}|$ is the module of the velocity vector and $v_c = \sqrt{\gamma\frac{p}{\rho}}$ is the speed of the sound in the flow. Also, $\nabla.\mathbf{F}_{m_i}$ is the divergence of the flux term corresponding to the ith momentum equation which may be expressed as

$$\nabla.\mathbf{F}_{m_i} = \frac{\partial(\rho v_i v_j)}{\partial x_j} + \frac{\partial p}{\partial x_i} \tag{11}$$

More details on the derivation of Equations (8), (9) and (10) can be found in [28, 29].

2.3 Space Discretization

In order to discretize the obtained stabilized formulation, the finite element method (FEM) is selected where the vector of the conservative variables $\mathbf{U}$ can be approximated by $\bar{\mathbf{U}}$ as

$$\mathbf{U} \simeq \bar{\mathbf{U}} = \sum_{J=1}^{n} \mathbf{N}_J \bar{\mathbf{U}}_J \tag{12}$$

where vector $\bar{\mathbf{U}}$ contains the approximate values corresponding to the conservative variables. $\mathbf{N}$ is the matrix of the interpolating shape functions, subscript index J represents the value for the Jth node and n is the number of nodes in each element. In our work we use 3-noded triangles with linear shape functions ($n = 3$).

Let us define the approximate residual vectors $\bar{\mathbf{r}}$ and $\bar{\mathbf{r}}_{st}$ as well as the vector of the approximated primitive variables $\bar{\bar{\mathbf{U}}}$ as

$$\bar{\mathbf{r}} = \begin{bmatrix} \bar{r}_d \\ \bar{r}_{m_1} \\ \bar{r}_{m_2} \\ \bar{r}_e \end{bmatrix} \qquad \bar{\mathbf{r}}_{st} = \begin{bmatrix} 1 \\ \bar{r}_{m_1} \\ \bar{r}_{m_2} \\ \bar{r}_e \end{bmatrix} \qquad \bar{\bar{\mathbf{U}}} = \begin{bmatrix} \bar{\rho} \\ \bar{v}_x \\ \bar{v}_y \\ \bar{T} \end{bmatrix} \tag{13}$$

where $\bar{r}_d$, $\bar{r}_{m_i}$ and $\bar{r}_e$ denote the finite element residuals for the mass balance equation, the ith momentum equation and energy equation, respectively.

By applying the Galerkin FEM to Equations (8), (9) and (10), integrating the inviscid flux terms as well as the stabilization terms by part and neglecting the boundary contributions of the stabilization terms, the general form of the FIC-FEM stabilized formulation for a problem domain Ω with a boundary $\Gamma = \partial\Omega$ is obtained as

$$\int_{\Omega} \mathbf{N}.\frac{\partial\bar{\mathbf{U}}}{\partial t}d\Omega - \int_{\Omega} \frac{\partial\mathbf{N}}{\partial x_i}.\bar{\mathbf{F}}_i d\Omega + \int_{\Gamma} \mathbf{N}.\bar{\mathbf{F}}_n d\Gamma$$

$$\tag{14}$$

$$- \sum_{e}^{n_{el}} \int_{\Omega_e} \frac{\tau}{2}\mathbf{A}_i\frac{\partial\mathbf{N}}{\partial x_i}.\bar{\mathbf{r}}_{st}d\Omega - \sum_{e}^{n_{el}} \int_{\Omega_e} \frac{1}{2}\mathbf{S}\frac{\partial\mathbf{N}}{\partial x_i}.\frac{\partial\bar{\bar{\mathbf{U}}}}{\partial x_i}d\Omega = 0$$

where n_{el} is the number of the elements and $i = 1, 2$ (for 2D problems). In the above equation, $\bar{\mathbf{F}}_i$ and $\bar{\mathbf{F}}_n$ are the ith component of the approximated inviscid flux vector $\bar{\mathbf{F}}$ and the vector of inviscid boundary flux along the normal vector to the boundary, respectively.

Also in Equation (14), $\tau = \frac{\beta\ell}{|\bar{\mathbf{v}}|+\bar{v}_c}$ is the stabilization parameter and the stabilization matrices $\mathbf{S}$ and $\mathbf{A}_i$ are

$$\mathbf{S} = (1-\beta)\ell \begin{bmatrix} \frac{|\bar{r}_d|}{|\nabla\bar{\bar{\mathbf{U}}}_1|} & 0 & 0 & 0 \\ 0 & \frac{|\bar{r}_{m_1}|}{|\nabla\bar{\bar{\mathbf{U}}}_2|} & 0 & 0 \\ 0 & 0 & \frac{|\bar{r}_{m_2}|}{|\nabla\bar{\bar{\mathbf{U}}}_3|} & 0 \\ 0 & 0 & 0 & \frac{|\bar{r}_e|}{|\nabla\bar{\bar{\mathbf{U}}}_4|} \end{bmatrix} \qquad \mathbf{A}_i = \begin{bmatrix} \nabla.\bar{\mathbf{F}}_{m_i} & 0 & 0 & 0 \\ 0 & \bar{v}_i & 0 & 0 \\ 0 & 0 & \bar{v}_i & 0 \\ 0 & 0 & 0 & \bar{v}_i \end{bmatrix} \tag{15}$$

where $\boldsymbol{\nabla}.\bar{\mathbf{F}}_{m_i}$ is the divergence of the approximate finite element flux terms corresponding to the ith momentum equation.

More details on the derivation of the proposed stabilized formulation as well as the treatment of the boundary conditions can be found in [28, 29]. As discussed in these references, the assumptions $\beta = 0.5$ and $\ell = (2\Omega^e)^{1/2}$ with Ω^e being the element area for 2D problems give yields suitable results. Hence, the same assumptions are taken in this work.

By neglecting the temporal derivative term (i.e. the first term) in Equation 14 and assembling the elemental contributions from Equation (14), the steady-state flow residual vector $\mathbf{R}_I(\bar{\mathbf{U}})$ corresponding to the Ith global node can be defined as

$$\mathbf{R}_I(\bar{\mathbf{U}}) = -\int_\Omega \frac{\partial \mathbf{N}_I}{\partial x_i}.\bar{\mathbf{F}}_i d\Omega + \int_\Gamma \mathbf{N}_I.\bar{\mathbf{F}}_n d\Gamma$$

$$-\sum_e^{n_{el}} \int_{\Omega_e} \frac{\tau}{2} \mathbf{A}_i \frac{\partial \mathbf{N}_I}{\partial x_i}.\bar{\mathbf{r}}_{st} d\Omega - \sum_e^{n_{el}} \int_{\Omega_e} \frac{1}{2}\mathbf{S} \frac{\partial \mathbf{N}_I}{\partial x_i}.\frac{\partial \bar{\mathbf{U}}}{\partial x_i} d\Omega = \mathbf{0}$$

(16)

where subscript index I represents the values corresponding to the Ith node.

Equation (16) is a non-linear system of equations whose solution is presented in Section 4.1.

3 Output-based AMR

3.1 Adjoint-based Error Estimation Formulation

Consider a spatial domain of resolution H related to a coarse mesh discretization. The corresponding output functional J_H is defined as

$$J_H = J_H(\bar{\mathbf{U}}_H)$$

(17)

where $\bar{\mathbf{U}}_H$ is the vector of the discrete flow variables obtained by satisfying the non-linear flow equation $\mathbf{R}_H(\bar{\mathbf{U}}_H) = 0$ on the coarse mesh.

Also, consider a spatial domain of resolution h related to a fine mesh discretization. By assuming $\bar{\mathbf{U}}_h$ as the solution of the flow equation $\mathbf{R}_h(\bar{\mathbf{U}}_h) = 0$ on the fine mesh, an output functional J_h can be expressed with the form

$$J_h = J_h(\bar{\mathbf{U}}_h)$$

(18)

Let us define $\bar{\mathbf{U}}_h^H$ as the flow variables obtained on the spatial domain of resolution H projected onto the spatial domain of resolution h. In general, the Taylor expansion of the functional J_h (only considering the linear terms) using the solution $\bar{\mathbf{U}}_h^H$ can be expressed as

$$J_h(\bar{\mathbf{U}}_h) \simeq J_h(\bar{\mathbf{U}}_h^H) + \left[\frac{\partial J_h}{\partial \bar{\mathbf{U}}_h}\right]_{\bar{\mathbf{U}}_h^H} (\bar{\mathbf{U}}_h - \bar{\mathbf{U}}_h^H)$$

(19)

where $J_h(\bar{\mathbf{U}}_h^H)$ is the functional value on the fine mesh calculated using $\bar{\mathbf{U}}_h^H$. Also, vector $\left[\frac{\partial J_h}{\partial \bar{\mathbf{U}}_h}\right]_{\bar{\mathbf{U}}_h^H}$ contains the sensitivities of the functional on the fine mesh respect to the flow variables evaluated using $\bar{\mathbf{U}}_h^H$.

We define now ε_t as the linear approximation of the total error in the functional evaluated on the fine mesh using projected solutions from the coarse one. By using Equation (19), ε_t can be represented as

$$\varepsilon_t = J_h(\bar{\mathbf{U}}_h) - J_h(\bar{\mathbf{U}}_h^H) = \left[\frac{\partial J_h}{\partial \bar{\mathbf{U}}_h}\right]_{\bar{\mathbf{U}}_h^H} (\bar{\mathbf{U}}_h - \bar{\mathbf{U}}_h^H)$$

(20)

It is to be mentioned that $\bar{\mathbf{U}}_h^H$ is not essentially equal to $\bar{\mathbf{U}}_h$, since $\bar{\mathbf{U}}_h^H$ does not necessarily satisfy $\mathbf{R}_h(\bar{\mathbf{U}}_h^H) = 0$. Hence, Equation (20) yields nonzero values for ε_t.

5

The same expansion as the one introduced for the $J_h(\bar{\mathbf{U}}_h)$ in Equation (19) can be written for the flow residual $\mathbf{R}_h(\bar{\mathbf{U}}_h)$ with the following form

$$\mathbf{R}_h(\bar{\mathbf{U}}_h) \simeq \mathbf{R}_h(\bar{\mathbf{U}}_h^H) + \left[\frac{\partial \mathbf{R}_h}{\partial \bar{\mathbf{U}}_h}\right]_{\bar{\mathbf{U}}_h^H} (\bar{\mathbf{U}}_h - \bar{\mathbf{U}}_h^H) \tag{21}$$

where $\left[\frac{\partial \mathbf{R}_h}{\partial \bar{\mathbf{U}}_h}\right]_{\bar{\mathbf{U}}_h^H}$ is the Jacobian matrix of the flow residual, presented in Equation (16), evaluated using $\bar{\mathbf{U}}_h^H$ on the fine mesh. By considering the fact that $\mathbf{R}_h(\bar{\mathbf{U}}_h) = 0$, the solution error $(\bar{\mathbf{U}}_h - \bar{\mathbf{U}}_h^H)$ appeared in Equation (21) can be represented as

$$(\bar{\mathbf{U}}_h - \bar{\mathbf{U}}_h^H) \simeq -\left[\frac{\partial \mathbf{R}_h}{\partial \bar{\mathbf{U}}_h}\right]_{\bar{\mathbf{U}}_h^H}^{-1} \mathbf{R}(\bar{\mathbf{U}}_h^H) \tag{22}$$

Substituting Equation (22) into (20) gives the following expression for the total functional error

$$\varepsilon_t = -\left[\frac{\partial J_h}{\partial \bar{\mathbf{U}}_h}\right]_{\bar{\mathbf{U}}_h^H} \left[\frac{\partial \mathbf{R}_h}{\partial \bar{\mathbf{U}}_h}\right]_{\bar{\mathbf{U}}_h^H}^{-1} \mathbf{R}_h(\bar{\mathbf{U}}_h^H) \tag{23}$$

The above equation states that the evaluation of the total functional error ε_t involves the inversion of the Jacobian matrix which is a process with high computational cost. In order to avoid inverting the Jacobian matrix directly, the discrete adjoint system of equations is defined as

$$\left[\frac{\partial \mathbf{R}_h}{\partial \bar{\mathbf{U}}_h}\right]_{\bar{\mathbf{U}}_h^H}^T (\mathbf{\Lambda}_h)_{\bar{\mathbf{U}}_h^H} = \left[\frac{\partial J_h}{\partial \bar{\mathbf{U}}_h}\right]_{\bar{\mathbf{U}}_h^H}^T \tag{24}$$

where $(\mathbf{\Lambda}_h)_{\bar{\mathbf{U}}_h^H}$ is the vector of the discrete adjoint variables calculated using $\bar{\mathbf{U}}_h^H$ on the fine mesh. Using Equation (24), the formulation of the total functional error presented in Equation (23) can be transformed to

$$\varepsilon_t = -(\mathbf{\Lambda}_h)_{\bar{\mathbf{U}}_h^H}^T \mathbf{R}_h(\bar{\mathbf{U}}_h^H) \tag{25}$$

From Equation (25) it can be deduced that the evaluation of the total functional error on the fine mesh would require the adjoint variables obtained by solving the discrete adjoint equations (24) on the fine mesh. As the main idea is to prevent solving any system of equations on the fine mesh, an approximation of $(\mathbf{\Lambda}_h)_{\bar{\mathbf{U}}_h^H}$ is made by solving the adjoint equations initially on the coarse mesh as

$$\left[\frac{\partial \mathbf{R}_H}{\partial \bar{\mathbf{U}}_H}\right]^T \mathbf{\Lambda}_H = \left[\frac{\partial J_H}{\partial \bar{\mathbf{U}}_H}\right]^T \tag{26}$$

where $\mathbf{\Lambda}_H$ is the vector containing the discrete adjoint variables corresponding to the coarse mesh. Once the discrete adjoint variables are computed on the coarse mesh, the projection of these values from the coarse mesh to the fine mesh, called $\mathbf{\Lambda}_H^h$, is calculated to be used as an approximation of the discrete adjoint variables corresponding to the fine mesh.

Let us define the computed functional error correction ε_c as an approximation of the total functional error ε_t. Using the projected adjoint variables, ε_c can be expressed as

$$\varepsilon_t \simeq \varepsilon_c = -(\mathbf{\Lambda}_h^H)^T \mathbf{R}_h(\bar{\mathbf{U}}_h^H) \tag{27}$$

The above equation states that ε_c is obtained as the inner product between the vector of projected adjoint variables and the vector of nonzero residuals. In contrast to Equation (25), the above equation demonstrates that the functional error can be approximately predicted without any solution on the fine mesh by just solving the flow and adjoint equations on the coarse mesh along with the proper projection procedure to the fine mesh level.

It is to be noted that the same projection operator as the one implemented for the flow solution is utilized for the discrete adjoint variables in this work. The solution method of Equation (26) is presented in Section 4.2.

3.2 Adaptation Criteria

The element-wise error indicator I_k is defined as the contribution of ε_c corresponding to the kth element belonging to the coarse mesh H with the form

$$I_k = -(\mathbf{\Lambda}_h^H)_k^T \mathbf{R}_h(\bar{\mathbf{U}}_h^H) \tag{28}$$

In fact, the local error indicator of element k is obtained as the inner product of the projected adjoint variables and local residual vectors both computed in subelements of element k in the coarse mesh. In order to find the candidate elements for the refinement, an adaptation criteria is employed where the local error indicator of each element is compared to an elemental allowable error as

$$|I_k| > \frac{E_{tol}}{N_{el}} \tag{29}$$

where E_{tol} denotes a positive desirable error tolerance and N_{el} is the number of elements in the coarse mesh. The elements satisfying Equation (29) are considered as the candidate elements for the refinement.

3.3 Refinement Methodology

The mesh refinement methodology implemented in this work is an isotropic h-refinement scheme where each candidate element is subdivided into four subelements. The element subdivision process is carried out through the insertion of extra points at the edge midpoints suited for the triangle elements.

For the candidate elements connected to a curved boundary, the new inserted point at the edge belonging to the boundary must be adjusted to the original boundary surface. For this reason, a correction on the coordinates of the new surface points is performed at the end of each adaptation level by implementing a quintic spline interpolation based on the coordinates of the original surface points. The geometrical interpretation of this process is depicted in Figure **??**.

Some consideration must be taken into account regarding the possibly generated hanging points where one or two edges of an element are split by the new points due to its candidate neighbour elements. For the case of an element with one hanging point, an anisotropic refinement into two subelements is implemented, whereas an element with two hanging points is divided regularly into four subelements, as shown in Figure 1. In order to keep the quality of the mesh, any anisotropic subelement is not allowed to be refined further. If any such element requires refinement, both anisotropic subelements are removed and the original coarse element is isotropically refined into four subelements.
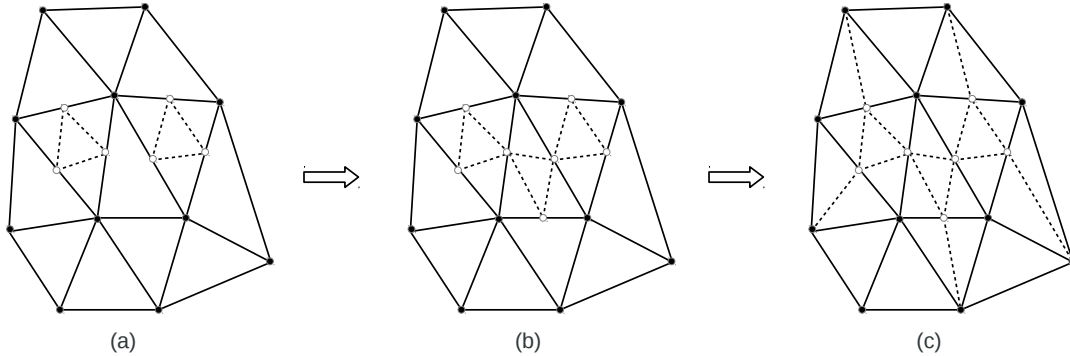

Figure 1: Mesh refinement process around the hanging nodes: (a) Isotropic refinement of two candidate elements, (b) Isotropic refinement of an element due to the presence of two handing nodes, (c) Anisotropic refinement of elements due to the presence of one hanging node.

3.4 AMR Procedure

The error estimation and mesh adaptation algorithm presented in this work consists of the following steps

1- Obtain the flow variables $\bar{\mathbf{U}}_H$ by solving Equation (16) for the coarse mesh.

2- Compute the adjoint variables $\mathbf{\Lambda}_H$ by solving Equation (26) for the coarse mesh using the converged flow solution $\bar{\mathbf{U}}_H$.

3- Evaluate the projected flow variables $\bar{\mathbf{U}}_H^h$ and the projected adjoint variables $\mathbf{\Lambda}_H^h$.

4- Find the candidate elements for refinement through Equations (28) and (29).

5- If the candidate elements exist, refine the coarse mesh and then return to step 1.

The AMR procedure finishes when there is no candidate element for refinement. It is to be mentioned that after each refinement level the solution of the flow and adjoint equations can be started using the previously projected variables.

The projection of the flow and adjoint variables from the current coarse mesh to the embedded fine one can be performed by either a linear or a higher-order interpolation method. The former requires only the values of the solution at the coarse mesh points, whereas the latter involves the calculation of the least-squares gradients of the solution. In this work, suitable results have been obtained by using a simple linear interpolation method.

4 Solution Method

In this section, the solution strategies for the flow and adjoint equations given in Equations (16) and (26), respectively, are presented. Since the exact Jacobian matrix is needed for the solution of the adjoint equations (See Equation (26)), an iterative Newton linearization scheme with an exact Jacobian matrix has been chosen as the solution approach for the nonlinear flow equations (i.e. Equation (16)). The adjoint equation given in Equation (26) corresponds to a linear system of equations. Although this system can be solved directly, an iterative scheme is presented by transferring these equations to defect correction form.

4.1 Flow Solution

The solution of Equation (16) is performed using following Newton's method

$$\left[\frac{\partial \mathbf{R}}{\partial \bar{\mathbf{U}}} \right]^n \Delta \bar{\mathbf{U}}^n = -\mathbf{R}(\bar{\mathbf{U}}^n)$$

$$\Delta \bar{\mathbf{U}}^n = \bar{\mathbf{U}}^{n+1} - \bar{\mathbf{U}}^n \tag{30}$$

where n is the iteration number. At the beginning of the solution, the Newton solver may diverge due to the start-up transients caused by a poor initial flow field. In order to overcome this problem, a damped Newton's method is implemented where the diagonal terms of the Jacobian matrix are augmented with a damping factor as

$$\left[\frac{\delta_{IJ} \mathbf{M}_{IJ}}{\Delta t_I} + \frac{\partial \mathbf{R}_I}{\partial \bar{\mathbf{U}}_J} \right]^n \Delta \bar{\mathbf{U}}_J^n = -\mathbf{R}_I(\bar{\mathbf{U}}^n)$$

$$\mathbf{M}_{IJ} = \int_\Omega \mathbf{N}_I \mathbf{N}_J d\Omega \tag{31}$$

with $I, J = 1, n_{node}$ where n_{node} is the total number of nodes in the mesh and $\mathbf{M}$ is the global mass matrix. In the above equation, the nodal (and pseudo) time step Δt_I is the minimum of the time steps corresponding to the elements connected to node I. The elemental time step for a candidate element e is determined from a uniform CFL number over the domain as

$$\Delta t_e = CFL \frac{\ell}{|\bar{\mathbf{v}}| + \bar{v}_c} \tag{32}$$

In this work the CFL number is responsible for adding a scalable damping term during the startup process and has the following form

$$CFL(n) = min(\alpha^n CFL(0), CFL(max)) \tag{33}$$

where $CFL(0) = 0.01$, $CFL(max) = 10^{12}$ and α is selected between 1.001 and 1.01 depending on the problem. Using this pattern for the CFL, by increasing the pseudo time step to a large value ($CFL = 10^{12}$) the damped unsteady terms vanish and the desired quadratic convergence rate of the Newton's method is achieved.

The linear system of equations (31) consisting of a sparse block matrix is solved using GMRES method described in [30]. In order to improve the convergence of this method, the preconditioned GMRES algorithm based on a block-diagonal preconditioning has been implemented in this work.

4.2 Adjoint Solution

Omitting the subscript H from Equation (26), the defect correction form of the adjoint equation can be written as

$$\left[\frac{\delta_{IJ}\mathbf{M}_{IJ}}{\Delta t_I} + \left[\frac{\partial \mathbf{R}}{\partial \bar{\mathbf{U}}} \right]^T \right]^k \Delta \mathbf{\Lambda}_J^k = \left[\frac{\partial J}{\partial \bar{\mathbf{U}}} \right]^T - \left[\frac{\partial \mathbf{R}}{\partial \bar{\mathbf{U}}} \right]^T \mathbf{\Lambda}^k \tag{34}$$

$$\Delta \mathbf{\Lambda}^k = \mathbf{\Lambda}^{k+1} - \mathbf{\Lambda}^k$$

where k denotes the iteration number. As seen in the above equation, the pseudo time term is added to the diagonal components of the Jacobian matrix to provide enough diagonal dominance for the system solver. This is the direct benefit of the defect correction method. The final converged solution for $\mathbf{\Lambda}$ can be obtained by continuously increasing the time step Δt_I using the pattern presented in Equations (32) and (33). Similarly as for the flow equations, a block-diagonal preconditioned GMRES method is used for the solution of the linear system of equations at each iteration step of Equation (34).

By comparing the iterative schemes developed for the solution of the flow and adjoint equations (Equations (31) and (34)), it is found that both systems have the same coefficient matrix with the exception of the transpose. Since the transpose operation does not change the eigenvalues of the coefficient matrix, both systems have the same convergence.

In the current work, a hand-coded linearization of the Galerkin terms as well as the stabilization terms in Equation (16) has been implemented giving a sparse Jacobian matrix for the unstructured triangular meshes used in this work. The Jacobian matrix based on the final steady-state flow solution is stored to be employed for the solution of the adjoint equation.

5 Numerical Results

In this section, a set of numerical examples related to the proposed adjoint-based error estimation and adaptive mesh refinement is presented. The first example involving a subsonic flow over a Gaussian bump consists of three parts. Firstly, the behavior of the proposed FIC-FEM formulation in conjunction with the presented damped Newton solver is validated. In the second part, the accuracy of the adjoint-based error estimates is examined by comparing the predicted errors obtained from the total functional error ε_t and the computed functional error correction ε_c with the true functional error between two mesh levels. In the last part, the adapted meshes related to the presented element-wise error indicator I_k as well as the error convergence graphs are delivered by considering the lift coefficient as the output functional. In the next two examples, the capability of this method in the presence of shock waves is evaluated by studying a transonic flow over NACA0012 airfoil and a supersonic flow over a ramp.

The error convergence graphs show the reduction of the functional error versus the number of nodes or the accumulative computational cost in CPU time. The computational cost corresponding to each refinement level consists of the consumed CPU time for the solution of the flow and adjoint

equations in all previous refinement levels plus an extra CPU time due to the flow solution on the current adapted mesh. The initial solution at each refinement level is obtained by interpolating the converged flow solution of the previous level from the old mesh to the new one. Regarding the uniform mesh refinement, the computational cost includes only the CPU time for a single flow solution on the uniformly refined mesh.

5.1   Example I: Subsonic Flow Over a Bump

A compressible flow with freestream Mach number of 0.35 passing a Gaussian bump is studied here to demonstrate the effectiveness of the developed error estimation and adaptation technique in a subsonic regime. The bump with a maximum thickness of 0.08 is placed at the center of the bottom side of a rectangular domain with height 2 and length 4. The initial unstructured mesh, shown in Figure 2, contains 952 nodes and 1902 3-noded triangles. The slip wall condition is imposed at the bottom boundary, whereas the far field condition is applied at the rest of the boundaries.
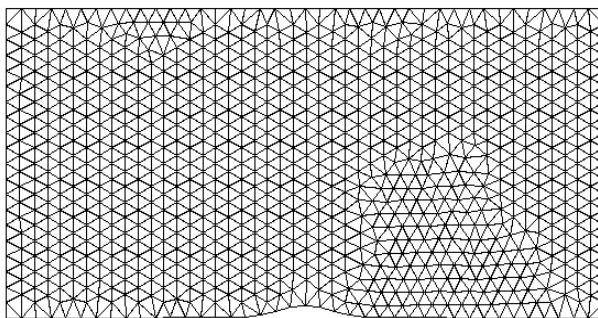
Figure 2: Subsonic flow over a bump. Initial uniform mesh.

1- Flow and Adjoint Solver
Since the main part of the error estimation and mesh adaptation procedure is the solution of the flow and adjoint equations, the characteristics of the solver in terms of the accuracy and efficiency can affect the quality of the final results during the adaptive mesh refinement process. For this reason, by assuming the lift coefficient as the functional, the numerical results corresponding to the solution of the flow and adjoint equations on the initial mesh are delivered here to demonstrate the behavior of the proposed FIC-FEM formulation in conjunction with the presented damped Newton solver.

The obtained contours corresponding to the pressure and the first component of the momentum adjoint variables are presented in Figure 3 indicating that the proposed FIC-FEM formulation gives rise to a smooth solution in all the domain even using a coarse mesh. The convergence graphs of the flow and adjoint solvers using the presented iterative strategy are shown in Figure 4 where an almost quadratic convergence is obtained for the flow solver after some iterations where the initial transients disappear and the CFL number can be slowly allowed to grow to its maximum value. As expected, an almost similar convergence rate to machine zero is obtained for the adjoint equation.
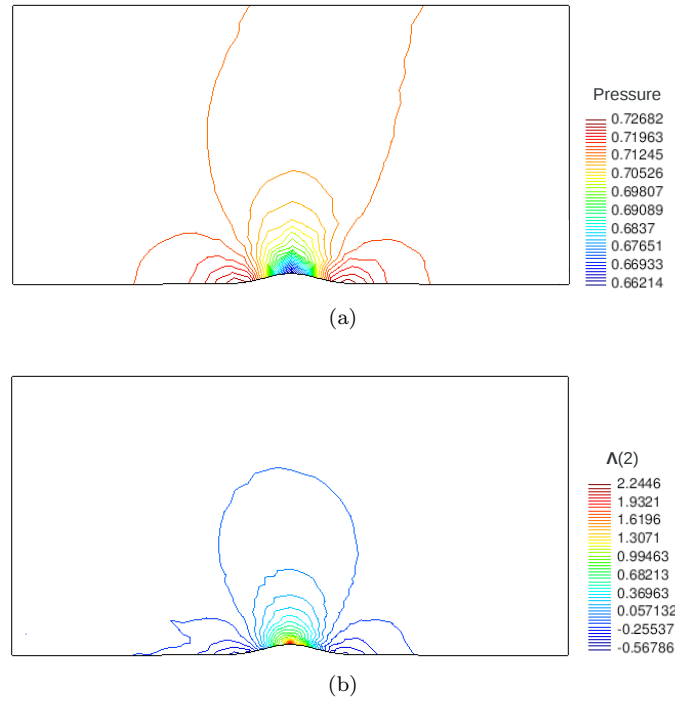
(a)

(b)

Figure 3: Subsonic flow over a bump. (a) Pressure contours and (b) the first component of the momentum adjoint variables corresponding to the initial uniform mesh.
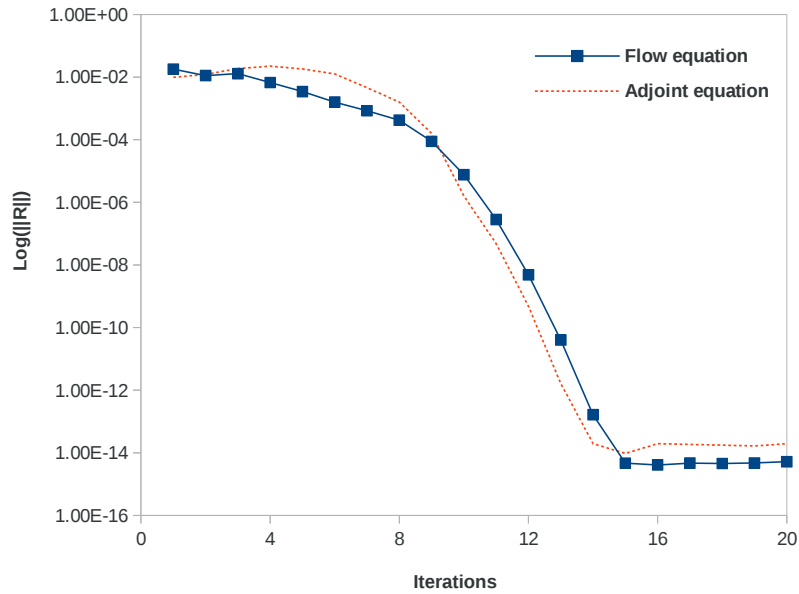

Figure 4: Subsonic flow over a bump. Convergence of the flow and adjoint equations for the initial coarse mesh.

2- Error prediction

The delivered scheme for error prediction is validated here for a set of meshes, called coarse, intermediate and fine mesh, consisting of 952, 3728 and 14852 nodes, respectively, where the lift coefficient is considered as the output functional. The idea is to compare the true functional error deduced from two mesh levels with the ones obtained from the total functional error ε_t and the computed functional error correction ε_c.

11

The obtained values for the functional error corresponding to the lift coefficient as well as the ratio values are demonstrated in Table 1. It can be seen that the ratio values obtained by both total and corrected errors asymptotically approach unity as the spatial resolution is increased. As expected, better compatibility between the true and total error values is observed for the finest discretization. The demonstrated ratio values corresponding to the total error indicate that considering only the linear term in the Taylor expansion of the functional, presented in Equation (19), is capable to predict the functional appropriately. On the other hand, the ratio values corresponding to the corrected errors illustrate that the proposed projection method for the adjoint variables is able to predict the functional error accurately.

Table 1: Subsonic flow over a bump. Comparison of the total and corrected adjoint predicted errors with the exact error for the lift coefficient.

Mesh	exact	ε_t	ε_c	$\varepsilon_t/$exact	$\varepsilon_c/$exact
Coarse	0.001562	0.001649	0.0012332	1.055	0.788
Intermediate	0.0004081	0.0004233	0.0003507	1.037	0.859
Fine	0.0001115	0.0001150	0.0001050	1.031	0.941

3- Mesh adaptation

After the validation of the flow and adjoint solvers as well as the error prediction scheme, the efficiency of the developed adjoint-based mesh adaptation strategy using the presented local error indicator I_k is studied for this example by considering the lift coefficient as the output functional. The procedure begins by using the uniform mesh, previously shown in Figure 2, while the termination criteria is the attainment of a tolerance of order 4×10^{-4} in the output functional. In this example, the true functional is computed from a solution of an adapted mesh obtained from refining the initial mesh while a relative functional difference of order 10^{-5} is considered between two consecutive adaptive meshes.

Figure 5 depicts the final adaptive mesh obtained after four levels of refinement. This figure indicates that the major part of the refinement is performed around the bump where the solution has a greater influence on the output functional. Also, the obtained pressure contours as well as the first component of the momentum adjoint variables corresponding to the final adapted mesh are presented in Figure 6 demonstrating that the final adapted mesh delivers smoother contours in comparison with the ones obtained by the initial uniform mesh plotted in Figure 3.
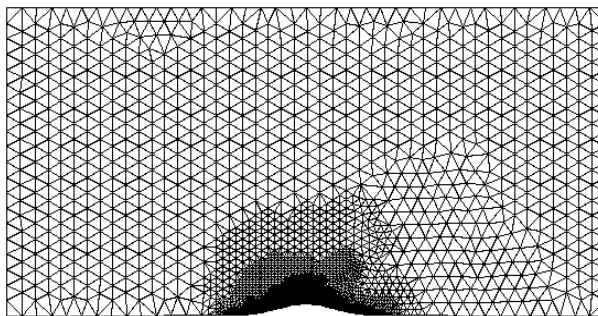

Figure 5: Subsonic flow over a bump. Final adapted mesh for lift coefficient after four refinement levels.
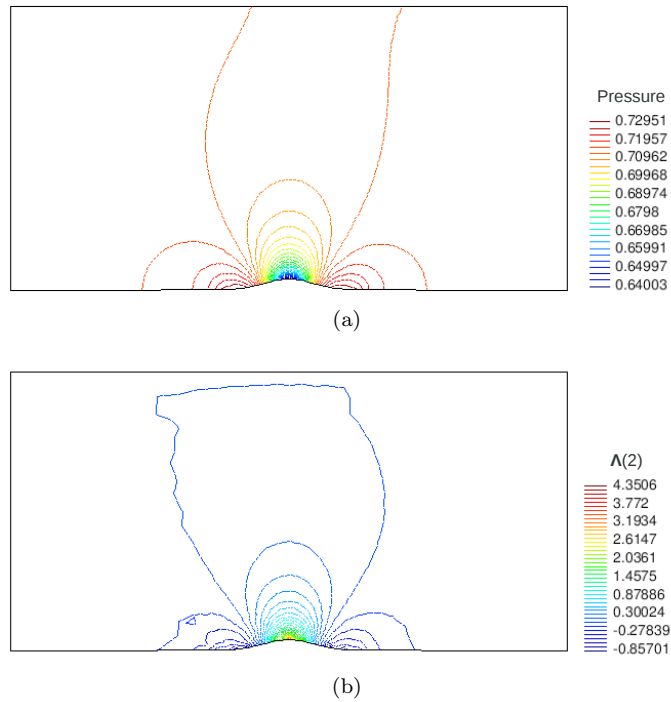
Figure 6: Subsonic flow over a bump. (a) Pressure contours and (b) the first component of the momentum adjoint variables corresponding to the final adapted mesh.

Figures 7a shows the convergence of the lift error versus the number of nodes where the error convergence of the adaptive method is compared with that of the uniform method. This figure demonstrates that the adaptive approach at each refinement level gives rise to relatively similar functional error as the uniform approach by using a lower number of nodes. As an example, we can mention that the number of nodes generated by the adaptive approach at the second refinement level is about 8% of the one generated by the uniform approach.
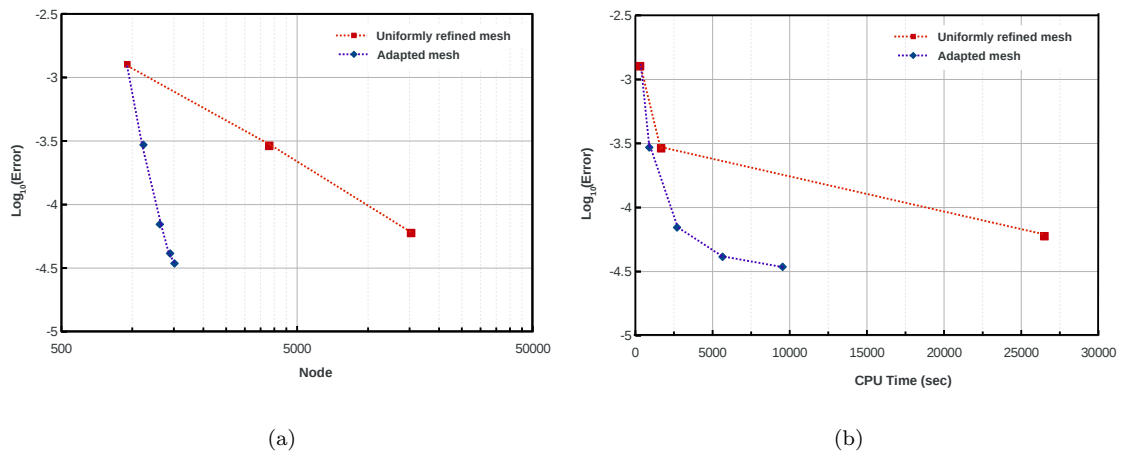
Figure 7: Subsonic flow over a bump. Comparison of the error convergence in the lift coefficient related to the uniform and adaptive mesh refinement versus (a) number of nodes and (b) CPU time (sec).

A more impressive behaviour can be seen in Figure 7b for the CPU time comparison where the adaptive approach demands much less CPU time than the uniform method due to the smaller

13

number of nodes resolved during the mesh adaptation process. Again, by comparing of the CPU time of both approaches at the second refinement level, it can be seen that the computational cost consumed by the adaptive approach is about 11% of the one used by the uniform method. It is worth noting that the CPU time of the adaptive approach involves the flow and adjoint solutions in all the previous levels, while that only the CPU time of a single flow solution on the refined mesh is counted for the case of uniform approach.

Figure 8 depicts the convergence graphs for the lift coefficient corresponding to the adaptive refinement by using the error estimation and the correction term as well as the one related to the uniform refinement. It is observed that at each refinement level the adaptive mesh can produce nearly the same functional value as the one obtained from the uniform mesh. Although, the agreement between the corrected functional values and the calculated ones from the adaptive mesh is poor for the primary refinement levels, a better agreement is obtained in the last refinement levels. This improvement is due to the fact that the corrected terms are calculated from the projected adjoint variables and by reducing the mesh size the projection process delivers better agreement between the exact and the projected values.
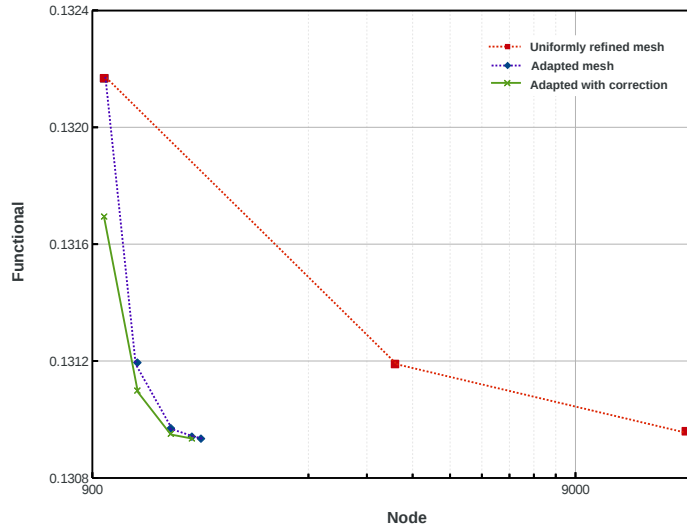

Figure 8: Subsonic flow over a bump. Convergence graphs of the lift coefficient using different refinement methods.

5.2   Example II: Transonic Flow Past NACA0012

In this example, a transonic flow past NACA0012 with the flow conditions $M_\infty = 0.8$ and $\alpha = 1.25°$ is considered involving a strong shock wave at the upper side of the airfoil and a weak shock wave at the lower side. The assumed circular domain is initially discretized into 3820 nodes and 7258 3-noded triangles. The analysis domain as well as the details of the initial mesh near the airfoil are shown in Figure 9. The slip wall condition is imposed at the airfoil surface, whereas the far field condition is applied at the outer boundary. An adjoint-based mesh adaptation for the drag coefficient is utilized in this example with a tolerance of 10^{-4}.
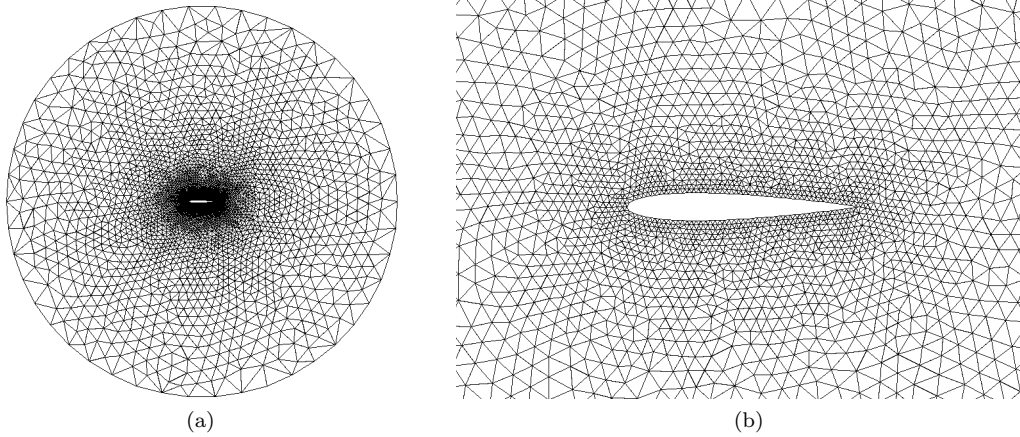
Figure 9: Transonic flow past NACA0012. (a) Initial mesh generated in the circular domain; (b) Details of the mesh near the airfoil.

The obtained pressure contours corresponding to the initial mesh are depicted in Figure 10a demonstrating that the FIC-FEM formulation is able to capture the shock waves appearing in the solution of the flow equation. The contours related to the first component of the momentum adjoint variables obtained on the initial mesh are shown in Figure 10b proving that the FIC-FEM formulation can yield a smooth solution for the adjoint equation.
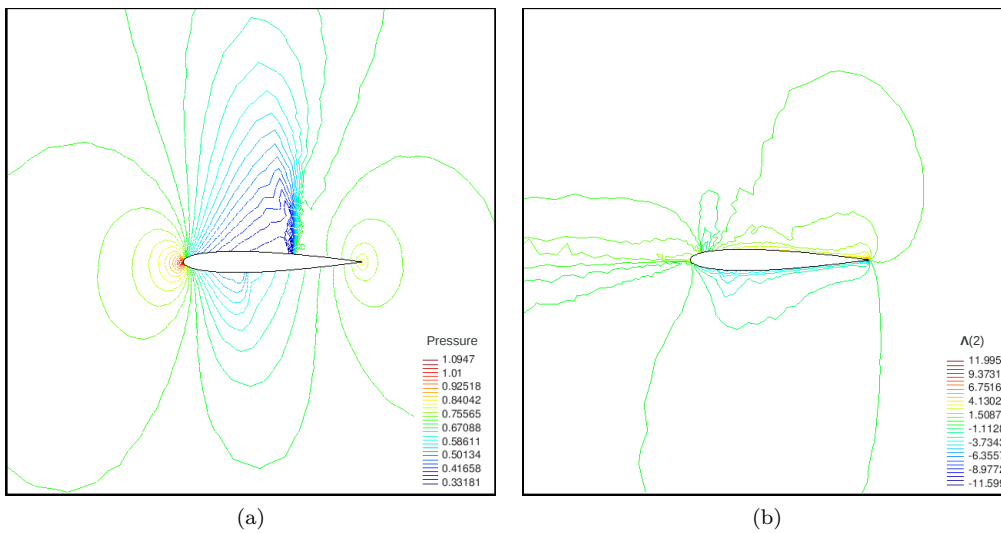
Figure 10: Transonic flow past NACA0012. (a) Pressure contours and (b) the first component of the momentum adjoint variables corresponding to the initial mesh.

The generated adaptive meshes after four refinement levels is shown in Figure 11 containing 8231 nodes and 15874 3-noded triangles. As expected, the mesh refinement is mostly carried out near the zones with higher importance like the leading and the trailing edges as well as the vicinity of the shock waves.
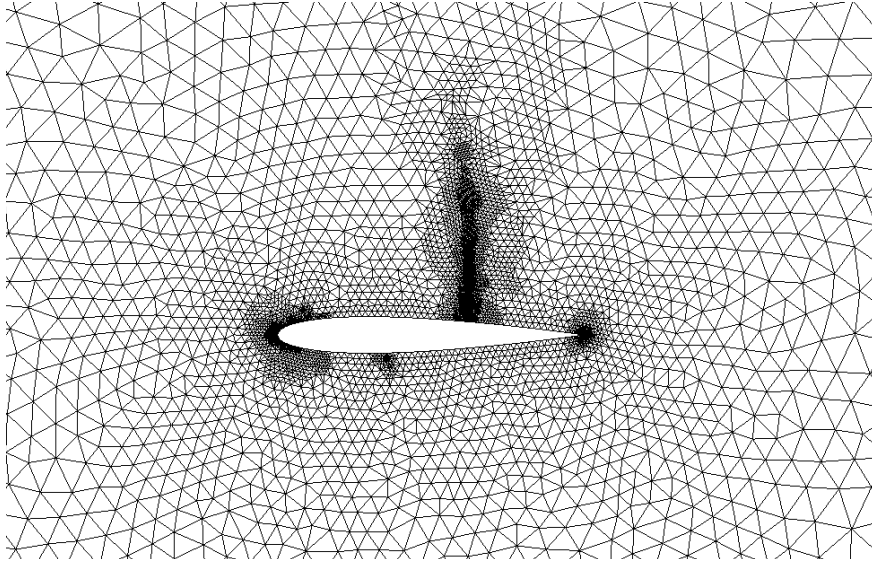
15

Figure 11: Transonic flow past NACA0012. Final adapted mesh for drag coefficient after four refinement levels.

The capability of the proposed error estimation and correction method is demonstrated in Figure 12 where the convergence graphs of the lift values obtained from different refinement methods versus the number of nodes are depicted. The comparison of the convergence graphs related to the uniform and the adaptive mesh refinement confirms that the functional values at each refinement level obtained from the adaptive method are in close agreement with the ones obtained from the uniform refinement method. Also, the predicted values at each refinement level provided by the added correction term are compared to the ones calculated on the adapted mesh. Although at the primary levels large differences are observed between the corrected functional values and the calculated ones, the corrected values become closer to the calculated ones as the refinement advances such that at the last refinement level these two values are coincident.
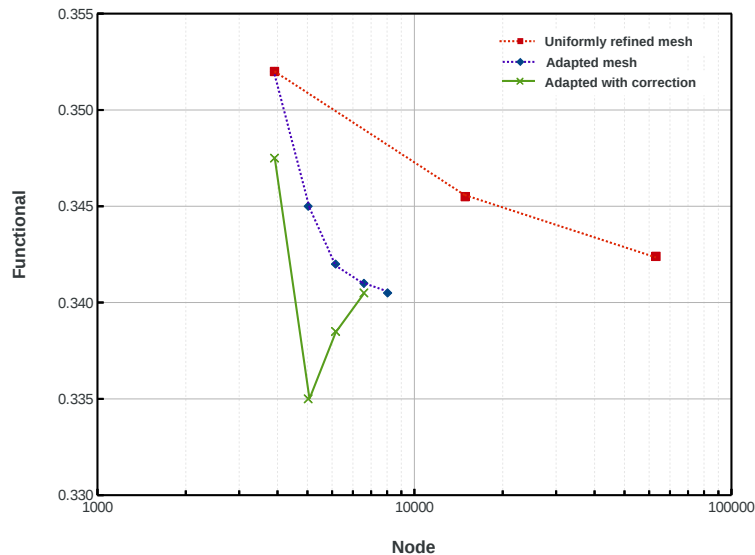

Figure 12: Transonic flow past NACA0012. Convergence graphs of the predicted drag coefficient using different refinement methods.

5.3 Example III: Supersonic Flow Over Ramp

In this example, a flow with Mach number 3.0 passing over a ramp of angle $19°$ is considered to investigate the effectiveness of the proposed adjoint-based error estimation and mesh adaptation method for a supersonic regime where both the flow and the adjoint variables are discontinuous. The analysis domain and the initial mesh consisting of 345 nodes and 612 3-noded triangles are shown in Figure 13a. The slip boundary condition is applied to the bottom side of the domain whereas the far field condition is imposed on the rest of the boundaries. The drag coefficient is chosen as the output functional in this example where a tolerance of 10^{-4} is used.

The contours of the pressure and the first component of the momentum adjoint variables are shown in Figures 13b and 13c, respectively, demonstrating that the FIC-FEM formulation can successfully capture the shock wave generated in the flow solution as well as the discontinuities appearing in the adjoint solution.

(a)   (b)   (c)
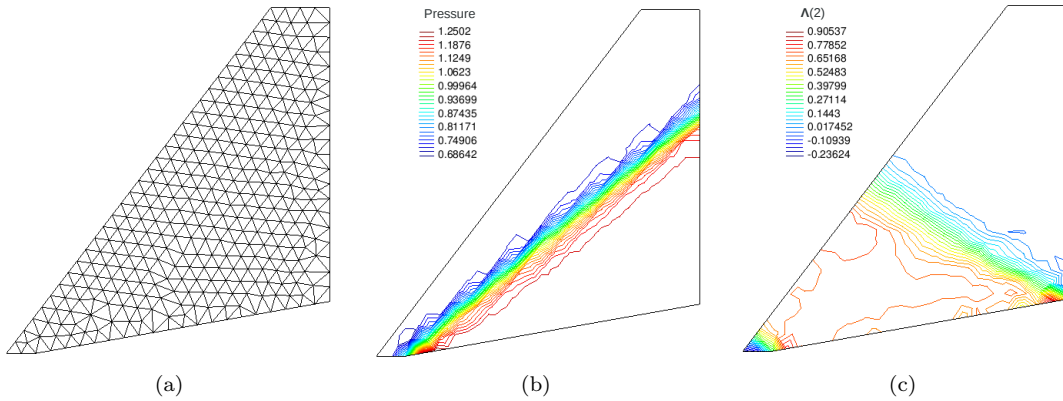
Figure 13: Supersonic flow past from ramp. (a) Initial mesh; (b) Pressure contours and (c) contours of the first component of the momentum adjoint variables corresponding to the initial mesh.

Based on the assumed tolerance, five refinement levels are carried out in this example. The generated adaptive mesh after one, three and five refinement levels are shown in Figure 14. It is to be noted that although the shock wave is expanded across the domain length entirely, the refinement is carried out just in the zones with higher effect on the drag coefficient. That is the benefit of implementing the adjoint-based mesh adaptation that does not allow refinement to be performed in the parts of the domain that have minimal impact on the output functional.

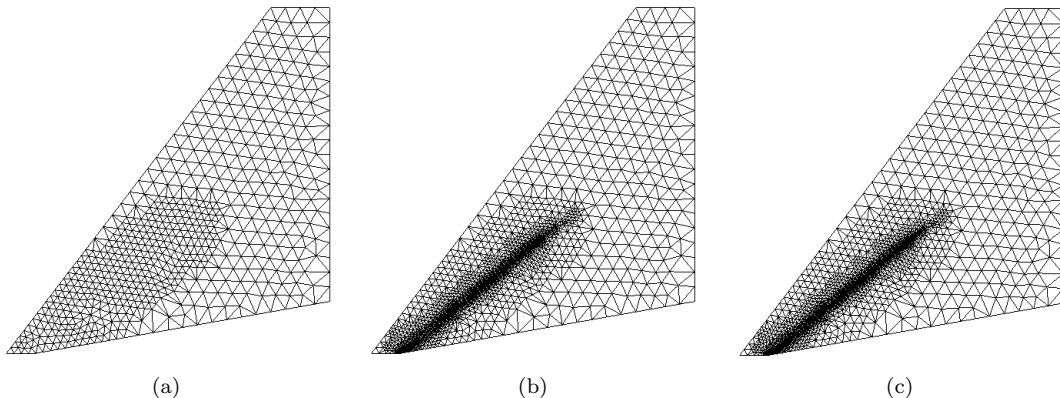(a)   (b)   (c)

Figure 14: Supersonic flow past from ramp. The obtained refined meshes after (a) one, (b) three and (c) five refinement levels.

The pressure contours are depicted in Figure 15 where the direct effect of the adjoint-based mesh adaptation on improving the resolution of the shock wave can be observed. Also, the obtained

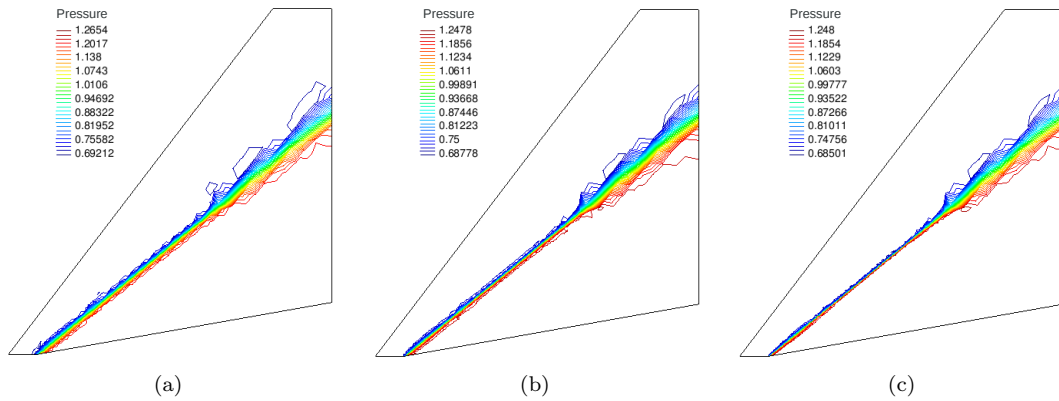contours related to the first component of the momentum adjoint variables are shown in Figure 16.

Figure 15: Supersonic flow past from ramp. The obtained pressure contours after (a) one, (b) three and (c) five refinement levels.
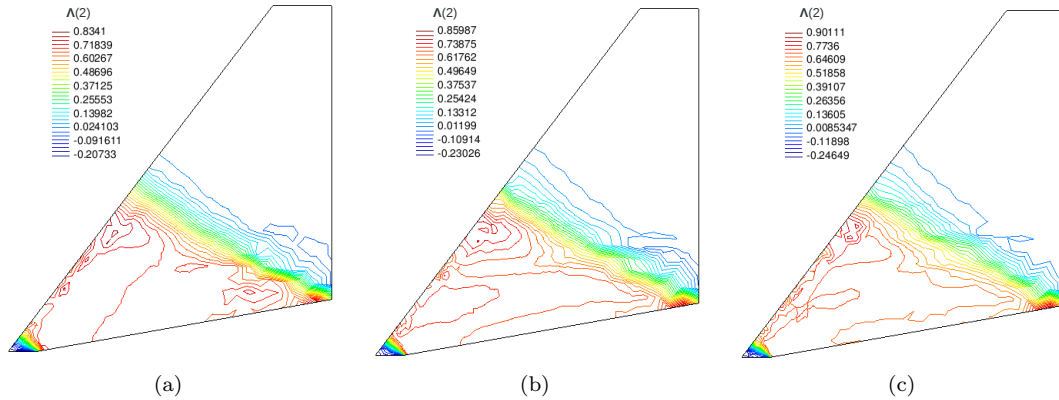

Figure 16: Supersonic flow past from ramp. The obtained contours of the first component of the momentum adjoint variables after (a) one, (b) three and (c) five refinement levels.

Figure 17 shows the convergence history of the calculated drag values in conjunction with the predicted values delivered by the addition of the correction term. This graph confirms the capability of the adjoint-based mesh refinement approach for delivering almost the same functional value as the uniform refinement method at each refinement level using fewer number of nodes. It can be also deduced from this graph that although the adjoint solution involves discontinuities, the accordance of the corrected functional values with the calculated ones progresses by advancing in the refinement process.
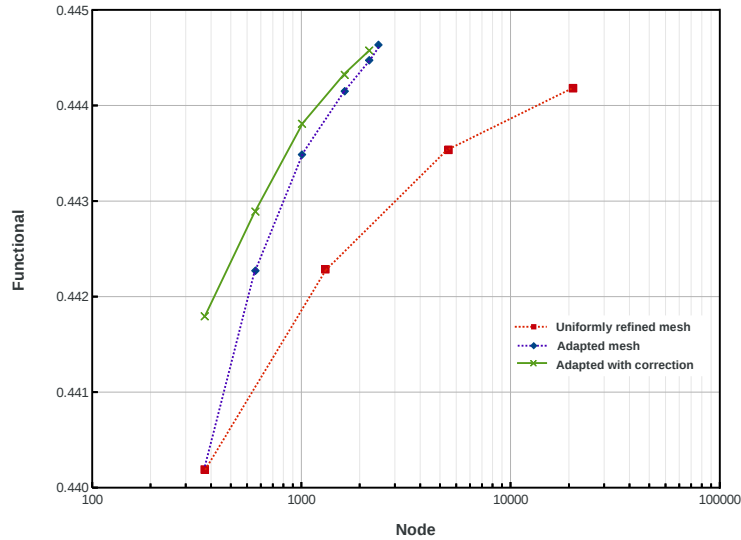
Figure 17: Supersonic flow past from ramp. Convergence graphs for the drag coefficient using different refinement methods.

6 Concluding Remarks

An adjoint-based error estimation and a mesh adaptation methodology have been developed and applied to compressible inviscid flow problems in the subsonic, transonic and supersonic regimes. By solving the flow and adjoint equations on the coarse mesh, the error related to a desirable functional was calculated as the inner product of the local residual and the adjoint variables. A linear interpolation has been employed to project the flow and adjoint variables from the coarse mesh into the fine one. For the mesh adaptation procedure, an error indicator based on the local contributions of the functional error in conjunction with the classical h-refinement methodology was used. The flow equations were discretized using the FIC-FEM stabilized formulation. A damped Newton method was implemented to solve the discretized flow equations whereas the adjoint equations were introduced in a defect corrected form. At each iteration step, the linear system of equation deduced from the flow and adjoint equations were solved by using a preconditioned GMRES solver. The efficiency of the proposed method was demonstrated by comparing the presented method with a uniform mesh refinement in terms of the number of nodes as well as the CPU time. The development of the current method for 3D turbulent viscous flows will be considered as the next task in this research. Another extension of this work will be the implementation of high-order finite element methods with a p-refinement methodology.

7 Acknowledgements

References

[1] T.J. Baker. Mesh adaptation strategies for problems in fluid dynamics. *Finite Elements Anal. Design*, 25:243–273, 1997.

[2] R. Löhner. Mesh adaptation in fluid mechanics. *Engineering Fracture Mechanics*, 50(5/6):819–847, 1995.

[3] J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, 72:449–466, 1987.

[4] O. C. Zienkiewicz and J. Wu. Automatic directional refinement in adaptive analysis of compressible flows. *Int. J. Numer. Meth. Eng.*, 72:2189–2210, 1994.

[5] M. Kouhi, D. S. Lee, G. Bugeda, and E. Oñate. Multi-objective aerodynamic shape optimization using MOGA coupled to advanced adaptive mesh refinement. *Computers & Fluids*, 88:298–312, 2013.

[6] S.Z. Pirzadeh. An adaptive unstructured grid method by grid subdivision, local remeshing, and grid movement. *AIAA Paper*, 99-3255, 1999.

[7] P. Nithiarasu and OC. Zienkiewicz. Adaptive mesh generation for fluid mechanics problems. *Int J Numer. Met.h Eng.*, 47:629–662, 2000.

[8] D. Li G. Xia and C.L. Merkle. Anisotropic grid adaptation on unstructured meshes. *AIAA Paper*, 2001-0443, 2001.

[9] D.A. Venditti and D.L. Darmofal. Grid adaptation for functional outputs: application to two-dimensional inviscid flows. *J. Comput. Phys.*, 176:40–69, 2002.

[10] D.A. Venditti and D.L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *J. Comput. Phys.*, 187:22–46, 2003.

[11] N. A. Pierce and M. Giles. Adjoint and defect error bounding and correction for functional estimates. *J. Comput. Phys.*, 200:769794, 2004.

[12] M. J. Aftosmis M. Nemec and M. Wintzer. Adjoint-based adaptive mesh refinement for complex geometries. *AIAA Paper*, 2008-725, 2008.

[13] R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica 2000*, 2001.

[14] R. Rannacher. Adaptive Galerkin finite element methods for partial differential equations. *J. Comput. Appl. Math.*, 128:205233, 2001.

[15] M. Levenstam M.B. Giles, M.G. Larson and E. Suli. Adaptive error control for finite element approximations of the lift and drag in a viscous flow. *Technical Report NA 97/06, Oxford Computing Laboratory, Oxford*, 1997.

[16] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys.*, 183:508532, 2002.

[17] L. Wang and D. J. Mavriplis. Adjoint-based h-p adaptive discontinuous Galerkin methods for the 2D Euler equations. *J. Comput. Phys.*, 228(20):7643–7661, 2009.

[18] N. K. Burgess and D. J. Mavriplis. An hp-adaptive discontinuous galerkin method for the navier-stokes equations. *In Proceedings of the SIAM Conference on Computational Science and Engineering*, Reno, NV, Mar 2011. Society of Industrial and Applied Mathematics, 2011.

[19] S. Nadarajah and A. Jameson. A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization. *AIAA 38th Aerospace Sciences Meeting and Exhibit, AIAA-2000-0667, Reno, NV*, January 2000.

[20] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. Wiley, 2003.

[21] O.C. Zienkiewicz, R.L. Taylor, and P. Nithiarasu. *The Finite Element Method. Vol. 3 Fluid Dynamics*. Elsevier, 6th Edition, 2005.

[22] E. Oñate. Derivation of stabilized equations for advective-diffusive transport and fluid flow problems. *Comput. Meth. Appl. Mech. Engng.*, 151:233–267, 1998.

[23] E. Oñate, J. Garcia, and S. Idelsohn. An alpha-adaptive approach for stabilized finite element solution of advective-diffusive problems with sharp gradients. *New Adv. in Adaptive Comp. Met. in Mech., P. Ladeveze and J.T. Oden (Eds.), Elsevier*, 1998.

[24] E. Oñate and M. Manzan. A general procedure for deriving stabilized space-time finite element methods for advective-diffusive problems. *Int. J. Num. Meth. Fluids.*, 31:203–221, 1999.

[25] E. Oñate. A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation. *Comp. Meth. Appl. Mech. Eng*, 182:355–370, 2000.

[26] E. Oñate and J. Garcia. A finite element method for fluid-structure interaction with surface waves using a finite calculus formulation. *Comput. Meth. Appl. Mech. Eng.*, 191:635–660, 2001.

[27] E. Oñate. Possibilities of finite calculus in computational mechanics. *Int. J. Num. Meth. Engng*, 60:255–281, 2004.

[28] M. Kouhi and E. Oñate. A stabilized finite element formulation for high-speed inviscid compressible flows using Finite Calculus. *Accepted to Int. J. Num. Meth. Fluids*, July 2013.

[29] M. Kouhi and E. Oñate. An implicit finite element formulation for compressibl Navier-Stokes equations using Finite Calculus. *Submitted to Int. J. Num. Meth. Fluids*, January 2014.

[30] Y. Saad and M. H. Schultz. GMRES: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal of Scientific and Statistical Computing*, 7:856–869, 1986.