V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

A ZONAL EULER/VISCOUS SOLVER FOR COMPRESSIBLE FLOWS

C. Biotto and J. Peiró

Imperial College London,
Department of Aeronautics,
South Kensington Campus, SW7 2AZ, U.K.
e-mail: cbiotto@imperial.ac.uk
j.peiro@imperial.ac.uk

Key words: Domain decomposition method, zonal solver, penalty method, viscous/inviscid, interface

Abstract. *We present a CFD zonal solver which efficiently simulates compressible viscous flows by reducing the computational cost through the use of a lower fidelity model, the Euler equations of inviscid flow, in regions where the flow is not dominated by viscous effects. This strategy also guarantees an acceptable level of accuracy. The design process can be accelerated if adequate solvers are used within each subdomain and a suitable method for passing information is used at the interface of these subdomains. In particular, the interface conditions play an extremely important role in ensuring continuity of the variables across the interface and they strongly affect the stability of the solver.*

Two implementations of the interface conditions are analysed: a strong approach where the variables are directly imposed on the interface and a weak approach based on a penalty method.

The two methods are tested for the solution of the turbulent flow over a flat plate and a subsonic airfoil. Both approaches are able to perform a turbulent simulation saving 43% *of the computational time respect to a complete turbulent simulation with an error of* 10% *in the wall friction coefficient.*

1 INTRODUCTION

Despite the large number of studies and the ongoing improvements in computer efficiency over the years, CFD simulations of turbulent flows about complex geometries are still challenging in terms of memory allocation and computational time.

Domain decomposition methods (DDMs) have been proposed in the literature to reduce the cost of simulations. The idea underlying DDMs is to split the domain of the problem into several independent partitions (subdomains) so that the convergence of the numerical method is speeded up. Several strategies can be used: such as, defining different meshes, modifying the mesh according to the flow features, using parallel computing[1] and applying the concept of the zonal solver, which involves the use of different equations in different subdomains. This last choice is the subject of this paper.

Since a high level of accuracy is frequently required only for some parts of the computational domain, such as near bodies and on wakes, the zonal solver is able to reduce the computational cost by using a high fidelity solver in these regions and a low fidelity solver elsewhere. The success of such approach relies on an appropriate treatment of the interfaces.

In particular, the interface conditions play an extremely important role because they are responsible for the continuity of the variables across the interface and they strongly affect the stability of the solver.

The literature contains a wealth of references to DDMs that partition the domain into subdomains in which an appropriate set of equations is solved and ensure a suitable matching strategy at the interface[1,2].

Several approaches for coupling subdomains using distinct solvers are available in the literature (for instance, potential flow/RANS[3], potential flow/Euler[4], vortex panel/Navier–Stokes[5], potential flow/Navier–Stokes[6] and LES/RANS[7]), however we focus exclusively on methods that divide the domain into viscous RANS and inviscid Euler subdomains, which are coupled together at the interface.

Within these methods, two types of zonal strategy have been proposed, depending on the turbulent model used by the RANS solver, which can be either algebraic or partial differential equation based (PDE).

Zonal algorithms for algebraic turbulence models, such as Baldwin–Lomax or Cecebi–Smith[8] have been proposed in the literature[9–11]. These models are based on a definition of turbulent viscosity as a function of Prandtl's mixing length, which is taken to be a function of the distance from the object. As the interface is easily determined from the negligible values of the turbulent viscosity, zonal strategies that are based on these algorithms are relatively straightforward to implement. Other authors have described methods based on one and two-equation turbulent models[12–14].

The present work deals with two approaches to the zonal solution by coupling Euler and RANS solvers for non-overlapping meshes. The first approach is inspired by a multi-domain method[15–17] and the second is an extension of a characteristic based method[13].

We will refer to these approaches as *weak* and *strong* approaches, respectively.

The weak approach is a modification of the method proposed by Hesthaven and co-workers. It couples equations on distinct subdomains using a penalty term to satisfy the interface matching conditions. In the strong method proposed by Quarteroni and Stolcis[13], the matching at the interface is achieved by imposing continuity of fluxes and incoming characteristic variables. A reduced turbulence model is used in the Euler region and therefore the method do not require interface conditions for the turbulent equations.

The choice of the interface conditions for the turbulence model is essential in both approaches, especially when the use of a turbulent model is completely avoided in the inviscid region. An efficient zonal solver has to treat the interface problem carefully, in order to obtain the best compromise between speed, stability and accuracy of the calculation.

In this paper we report different approaches that have been used to perform a zonal turbulent simulation, with the aim of reducing computational time. The inviscid region is assumed to be turbulence-free and thus the turbulent equations need to be solved only in the viscous region. As a result, the same boundary conditions which are usually employed at the far-field boundary have been applied at the interface between the two zones.

2 NUMERICAL FORMULATION

The solver used here was originally developed to solve steady/unsteady Euler, laminar Navier-Stokes and RANS equations, including a k-ε two-equation turbulence model using multi-stage Runge–Kutta time integration. The equations are discretized via the finite volume method on polyhedral unstructured grids.

2.1 Governing equations

The integral form of the Navier-Stokes equations are

$$\frac{\partial}{\partial t}\int\limits_{\Omega} \mathbf{Q}\, d\Omega + \int\limits_{\partial\Omega} (\mathbb{F}\cdot\mathbf{n})\; d(\partial\Omega) + \int\limits_{\Omega} \mathbf{S}\, d\Omega = \mathbf{0} \tag{1}$$

where $\mathbf{Q} = [\rho, \rho u, \rho v, \rho e, \rho k, \rho\varepsilon]^\top$ denotes the vector of conservative variables, e the internal energy, $\mathbb{F} = [\mathbf{F}_x, \mathbf{F}_y]^\top$ denotes the matrix of fluxes with reference to a Cartesian frame (x, y), the vector $\mathbf{n} = [n_x, n_y]^\top$ represents the outer normal to the boundary $\partial\Omega$, the velocity vector is $\mathbf{u} = [u, v]^\top$ and $\mathbf{S} = [0, 0, 0, 0, S_k, S_\varepsilon]^\top$ is the source term of the turbulence model[18]. This is a two-equation k-ε turbulence model where k denotes the turbulent kinetic energy and ε is the dissipation. The flux vectors are given by

$$\mathbb{F} = \mathbb{F}^i + \mathbb{F}^v = \begin{bmatrix} \rho u & \rho v \\ \rho u^2 + p & \rho v u \\ \rho u v & \rho v^2 + p \\ (\rho e + p)u & (\rho e + p)v \\ \rho u k & \rho v k \\ \rho u \varepsilon & \rho v \varepsilon \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ \tau_{xx} & \tau_{xy} \\ \tau_{yx} & \tau_{yy} \\ u\tau_{xx} + v\tau_{yx} + q_x & u\tau_{xy} + v\tau_{yy} + q_y \\ (\mu_l + \frac{\mu_t}{\sigma_k})\dfrac{\partial k}{\partial x} & (\mu_l + \frac{\mu_t}{\sigma_k})\dfrac{\partial k}{\partial y} \\ (\mu_l + \frac{\mu_t}{\sigma_\varepsilon})\dfrac{\partial \varepsilon}{\partial x} & (\mu_l + \frac{\mu_t}{\sigma_\varepsilon})\dfrac{\partial \varepsilon}{\partial y} \end{bmatrix} \tag{2}$$

where $\mathbb{F}^i$ are the inviscid fluxes, $\mathbb{F}^v$ are the viscous fluxes and p is the pressure. For a perfect gas we also have

$$p = (\gamma - 1)\rho\left(e - \frac{u^2 + v^2}{2}\right); \tag{3}$$

with $\gamma = 1.4$. The stress tensor and thermal fluxes are expressed as

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}\left(\mu\frac{\partial u_k}{\partial x_k} + \rho k\right) \qquad q_i = \frac{\gamma}{\gamma - 1}\frac{\mu}{p_r}\frac{\partial T}{\partial x_i} \tag{4}$$

where $\mu = \mu_l + \mu_t$ is the effective viscosity with μ_l and μ_t denoting the laminar and turbulent viscosity respectively, p_r is the reference constant pressure and T is the temperature.

The turbulence model is a low–Reynolds number k-ε turbulence model[19,20]. The turbulent viscosity and the source terms are given by

$$\mu_{\mathrm{t}} = C_\mu f_\mu \rho \frac{k^2}{\varepsilon} \qquad S_k = P_k - \rho\varepsilon \qquad S_\varepsilon = (C_{\varepsilon_1} f_1 P_k - C_{\varepsilon_2} f_2 \rho\varepsilon)\frac{\varepsilon}{k} \tag{5}$$

where $P_k = \boldsymbol{\tau} \otimes \nabla \mathbf{u}$ is the rate of term production of turbulent energy. The constant are $C_\mu = 0.09$, $C_{\varepsilon 1} = 1.42$, $C_{\varepsilon 2} = 1.83$, $\sigma_k = 1.367$ and $\sigma_\epsilon = 1.367$, and f_μ, f_1 and f_2 are damping functions[21].

2.2 Finite volume scheme

The code is an explicit cell-centred solver. All dependent flow properties are stored within a cell and interface properties are found via interpolation, which will be described later. The standard divergence theorem is applied to integrate the governing equations and using a cell as the control volume of the discretization, the finite volume formulation reads

$$\mathcal{V}_I \frac{d\mathbf{Q}_I}{dt} + \sum_J \mathcal{S}_{IJ}\,\hat{\mathbf{F}}_{IJ} + \mathcal{V}_I\,\mathbf{S}_I = \mathbf{0} \tag{6}$$

where $\hat{\mathbf{F}} = \mathbb{F} \cdot \mathbf{n}$ represents the (outer) normal flux through the face shared by the cells I and J. Here $\mathcal{V}_I$ denotes the volume of the control volume and $\mathcal{S}_{IJ}$ represents the area of the face shared by cells I and J. The index J in the summation runs over all the cells that share a face with the cell I.

The inviscid fluxes are calculated using a second-order upwind scheme and the viscous stress terms are calculated using standard central difference representations based on the gradient calculations. The details of the upwind formulation are given by the Roe scheme[22]. The flux at the interface is written as

$$\hat{\mathbf{F}}_{IJ} = \frac{1}{2}\left(\hat{\mathbf{F}}_I + \hat{\mathbf{F}}_J\right) - \frac{1}{2}\left|A(\mathbf{Q}_{IJ}, \mathbf{n})\right|(\mathbf{Q}_J - \mathbf{Q}_I) \tag{7}$$

where A is the decomposed average state Jacobian which allows the method to identify the direction of the wave propagation. This term weights the interface in terms of the direction of the wave propagation and the resulting method is in essence an upwind scheme[23,24]. Note the viscous fluxes are evaluated independently of the inviscid fluxes and calculated explicitly from the previous time level.

Gradient terms which are required for second-order calculations and flow problems that require stress terms are calculated using the divergence theorem around each cell. Face gradients are then taken as the average of the cell centre values with a correction being made in the face normal direction

$$\mathbf{G}_I = \frac{1}{\mathcal{V}_I}\sum_J \frac{1}{2}\mathcal{S}_{IJ}(\mathbf{Q}_I + \mathbf{Q}_J)\mathbf{n} + \tilde{\mathbf{G}}_I \tag{8}$$

When a second-order procedure is applied the methodology is strictly speaking exactly the same as in (7), but the values $\mathbf{Q}$ either side of the interface are now determined using a second-order extrapolation. In addition a limiter is employed to prevent non-physical spatial oscillations in the solution. The second order state value is expressed by

$$\tilde{\mathbf{Q}}_I = \mathbf{Q}_I + c_{\lim}\mathbf{G}_I \cdot \mathbf{s} \tag{9}$$

where $c_{\lim}$ is found by a modified van Albada limiter[25] and $\mathbf{s}$ is the cell centre to cell centre vector.

2.3 Boundary Conditions

Boundary conditions are stored in fictitious cells so that the interface value is set correctly through the interface fluxes. Solid walls are treated using the no-slip boundary condition. The pressure, density and temperature values at the solid surfaces are extrapolated from the interior using appropriate relationships usually devise to conserve enthalpy. The turbulence energy is simply set to enforce a zero value at the wall and the turbulent dissipation at the wall is

$$\varepsilon_{wall} = 2\frac{\mu k}{\rho y_p^2} \tag{10}$$

The normal distance from the surface to the centroid of the first adjacent cell attached to a wall is denoted by y_p[26]. Inflow and outflow boundary conditions are based on the Navier–Stokes characteristic boundary conditions for viscous flows[27].

3 TREATMENT OF THE INTERFACE

We refer to the computational domain as Ω, which is divided into two regions, Ω_I and Ω_V, where the subindices I and V refer to the "inviscid" and "viscous" regions, respectively. The boundaries of Ω_I and Ω_V are denoted by $\partial\Omega_I$ and $\partial\Omega_V$ respectively. The interface between the regions is denoted by $\Gamma = \Omega_I \cap \Omega_V = \partial\Omega_I \cap \partial\Omega_V$. We adopt the convention that its normal $\mathbf{n} = (n_x, n_y)$ points to the viscous region and $\mathbf{t} = (t_x, t_y)$ is the tangential vector ($\mathbf{t} \cdot \mathbf{n} = 0$).

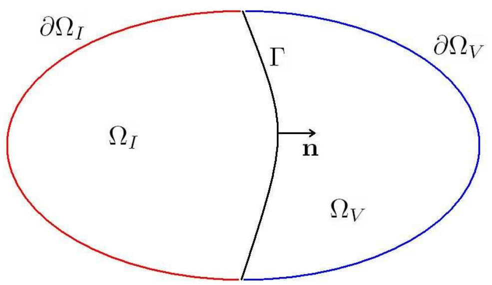

Figure 1: Zonal solver notation: inviscid region, Ω_I, with boundary $\partial\Omega_V$; viscous region, Ω_V, with boundary $\partial\Omega_V$; zonal interface, $\Gamma = \Omega_I \cap \Omega_V = \partial\Omega_I \cap \partial\Omega_V$.

For convenience of notation, we split the flow equations from the two turbulent equations, as

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho\mathbf{u} \\ \rho e \\ \hline \rho k \\ \rho\varepsilon \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{Q}} \\ \hline \bar{\mathbf{Q}} \end{bmatrix} \qquad \mathbb{F} = \begin{bmatrix} \tilde{\mathbb{F}} \\ \hline \bar{\mathbb{F}} \end{bmatrix} \tag{11}$$

This permits us to write the coupled inviscid–viscous problem as

$$\begin{cases} \dfrac{\partial \tilde{\mathbf{Q}}_I}{\partial t} + \nabla \cdot \tilde{\mathbb{F}}^i(\tilde{\mathbf{Q}}_I) & = \mathbf{0} \quad \text{in } \Omega_I \\ \dfrac{\partial \mathbf{Q}_V}{\partial t} + \nabla \cdot \mathbb{F}^i(\mathbf{Q}_V) & = \nabla \cdot \mathbb{F}^v(\mathbf{Q}_V) + \mathbf{S} \quad \text{in } \Omega_V \end{cases} \tag{12}$$

with appropriate boundary and interface conditions.

3.1 Strong approach

In the strong approach the interface is treated as a boundary where continuity of normal fluxes and of the Riemann invariants associated with the characteristic curves is imposed[13]. The continuity of the normal fluxes is expressed by

$$\tilde{\mathbb{F}}^i(\tilde{\mathbf{Q}}_I) \cdot \mathbf{n} = \tilde{\mathbb{F}}^i(\tilde{\mathbf{Q}}_V) \cdot \mathbf{n} - \tilde{\mathbb{F}}^v(\tilde{\mathbf{Q}}_V) \cdot \mathbf{n} \quad \text{on } \Gamma \tag{13}$$

From a numerical point of view, the continuity of the fluxes is automatically guaranteed by the single evaluation of the fluxes on the edges of the control volumes since the interface is always placed on a mesh edge.

The complexity of matching the characteristic variables at the interface is reduced by projecting of relevant variables in the direction normal to the face and restricting the matching to the normal direction only.

To accomplish this, we define $\mathbf{L}$ as the left eigenvectors matrix of the projection of the matrix $\mathbf{A}$ along the normal vector $\mathbf{n}$, where

$$\left\{\mathbf{A}^l\right\}_{r,s} = \frac{\partial \tilde{\mathbb{F}}^i(\tilde{\mathbf{Q}}_I)_{r,l}}{\partial \tilde{\mathbf{Q}}_s} \tag{14}$$

so that continuity of the invariants on the interface is described by

$$\mathbf{L}^{-1}\tilde{\mathbf{Q}}_I = \mathbf{L}^{-1}\tilde{\mathbf{Q}}_V \tag{15}$$

for all the negative eigenvalues of $\mathbf{A}$.

This approach intends to satisfy simultaneously the continuity of fluxes and of characteristic variables on the interface. The choice of the conditions to be applied to the variables has a strong impact on the solution of the zonal problem. Here we use non-reflecting boundary conditions[24] in order not to generate spurious numerical reflections at the interface.

The vanishing of the local perturbation carried along the characteristics can be written in discretized form as

$$\begin{bmatrix} \Delta\rho - \dfrac{\Delta p}{c^2} \\ \Delta(\mathbf{u}\cdot\mathbf{t}) \\ \Delta(\mathbf{u}\cdot\mathbf{n}) + \dfrac{\Delta p}{\rho c} \\ \Delta(\mathbf{u}\cdot\mathbf{n}) - \dfrac{\Delta p}{\rho c} \end{bmatrix} = \mathbf{0} \tag{16}$$

where c is the speed of sound and symbol $\Delta\beta$ is interpreted as the jump of the generic quantity β between the values at both side of the interface. The condition applied on the interface depends on the direction of the velocity vector normal to the interface that identify the direction of propagation. The variables for a subsonic "viscous to inviscid" interface are given by

$$\begin{aligned} (\mathbf{u}\cdot\mathbf{n})_\Gamma &= \frac{1}{1+\dfrac{\rho_V c_V}{\rho_I c_I}}\left(\mathbf{u}_I\cdot\mathbf{n} + \frac{\rho_V c_V}{\rho_I c_I}\mathbf{u}_V\cdot\mathbf{n} + \frac{\Delta p_{VI}}{\rho_I c_I}\right) \\ (\mathbf{u}\cdot\mathbf{t})_\Gamma &= (\mathbf{u}\cdot\mathbf{t}) \\ \rho_\Gamma &= \frac{(\mathbf{u}\cdot\mathbf{n})_\Gamma \rho_V}{c_V} + \rho_V \\ p_\Gamma &= p_V - (\mathbf{u}\cdot\mathbf{n})_\Gamma \rho_V c_V \end{aligned} \tag{17}$$

where the subindex Γ refers to the value of the interface variables. Once the variables are fixed, it is possible to compute the fluxes through the interface. The computation of the inviscid fluxes is straightforward

$$\tilde{\mathbb{F}}^i(\tilde{\mathbf{Q}}_\Gamma) = \begin{cases} \rho_\Gamma(\mathbf{u}\cdot\mathbf{n})_\Gamma \\ \rho_\Gamma(\mathbf{u}\cdot\mathbf{n})_\Gamma u_\Gamma + p_\Gamma n_x \\ \rho_\Gamma(\mathbf{u}\cdot\mathbf{n})_\Gamma v_\Gamma + p_\Gamma n_y \\ (\mathbf{u}\cdot\mathbf{n})_\Gamma(\rho e + p)_\Gamma \end{cases} \tag{18}$$

On the other hand, the calculation of the viscous fluxes is more complex because it involves the computation of the derivatives of the velocity. However, it is important to notice that the viscous fluxes are needed only in the case of flow coming from the viscous region. In the case of fluid flowing from inviscid to viscous region, the viscous fluxes are zero. Different methods are tested to compute derivatives of the viscous flux, either using only the variables in the viscous region or using variables in both regions. However, the best result is obtained supposing that there is no gradient on the interface and therefore the viscous interface fluxes are always zero.

3.2 Weak approach

The starting point of this method is a penalty procedure for the treatment of an interface in a domain decomposition method for viscous flows where the same solver is used in all the subdomains[15]. The method can be adapted to deal with viscous-inviscid coupling by a modification of the interface treatment. The method is proven to be well-posed and numerically stable[16,17].

The interface conditions are implemented using a penalty term, which is added to equations (12), of the form

$$\frac{\partial\tilde{\mathbf{Q}}}{\partial t} + \nabla\cdot\tilde{\mathbb{F}}^i = \nabla\cdot\tilde{\mathbb{F}}^v - \beta\boldsymbol{\mathcal{S}}\left[\boldsymbol{\mathcal{R}}_\pm(\mathbf{R}-\mathbf{R}_{BC}) + \boldsymbol{\mathcal{G}}_\pm(\mathbf{G}-\mathbf{G}_{BC})\right] \tag{19}$$

where β is a penalty parameter that is different from zero only at the interface, $\boldsymbol{\mathcal{S}}$ is the matrix of eigenvectors in the direction normal to the interface, $\boldsymbol{\mathcal{R}}$ represents the corresponding (diagonal) matrix of eigenvalues and $\boldsymbol{\mathcal{G}}$ is a diagonal matrix that arises from an energy integral introduced to achieve maximal dissipation for the interface conditions.

The subindices $-$ and $+$ indicate that their evaluation involves only the values that are compatible with upwind and downwind locations, respectively, with respect to the normal velocity at the interface. The symbols $\mathbf{R}$ and $\mathbf{G}$ denote the vector of characteristic variables in the inviscid and viscous regions, respectively. The subindex "BC" refers to the matching conditions at the interface. Detailed expressions of the matrices are given in (20). Here q is the velocity magnitude, H the total enthalpy, λ the eigenvalues associated with $\boldsymbol{\mathcal{S}}$, $\theta = \frac{\mu}{PrC_v}$, Pr is the Prandtl number and $\mathbf{R}^n = \mathbf{R}\cdot\mathbf{n}$. The derivative $\frac{\partial}{\partial n}$ in the term $\mathbf{G}$ is calculated in the direction normal to the interface.

$$\boldsymbol{\mathcal{S}} = \begin{bmatrix} \dfrac{1}{2c} & n_x & n_y & \dfrac{1}{2c} \\ \dfrac{1}{2c}(u + cn_x) & un_x & un_y & \dfrac{1}{2c}(u - cn_x) \\ \dfrac{1}{2c}(v + cn_y) & vn_x & vn_y & \dfrac{1}{2c}(v - cn_y) \\ \dfrac{1}{2c}(H + c\mathbf{u}\cdot\boldsymbol{n}) & \dfrac{q^2}{2}n_x & \dfrac{q^2}{2}n_y & \dfrac{1}{2c}(H - c\mathbf{u}\cdot\boldsymbol{n}) \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} \rho(\mathbf{u}-\mathbf{u}_\Gamma)\cdot\boldsymbol{n} + \dfrac{\gamma-1}{c_\Gamma}\left(\rho e + \dfrac{1}{2}\rho q_\Gamma^2 - \rho\mathbf{u}_\Gamma\cdot\mathbf{u}\right) \\ n_x\left[\rho - \dfrac{\gamma-1}{c_\Gamma^2}\left(\rho e + \dfrac{1}{2}\rho q_\Gamma^2 - \rho\mathbf{u}_\Gamma\cdot\mathbf{u}\right)\right] + \rho\left[(\mathbf{u}-\mathbf{u}_\Gamma)\times\boldsymbol{n}\right]\cdot n_x \\ n_y\left[\rho - \dfrac{\gamma-1}{c_\Gamma^2}\left(\rho e + \dfrac{1}{2}\rho q_\Gamma^2 - \rho\mathbf{u}_\Gamma\cdot\mathbf{u}\right)\right] + \rho\left[(\mathbf{u}-\mathbf{u}_\Gamma)\times\boldsymbol{n}\right]\cdot n_y \\ -\rho(\mathbf{u}-\mathbf{u}_\Gamma)\cdot\boldsymbol{n} + \dfrac{\gamma-1}{c_\Gamma}\left(\rho e + \dfrac{1}{2}\rho q_\Gamma^2 - \rho\mathbf{u}_\Gamma\cdot\mathbf{u}\right) \end{bmatrix}$$

$$\boldsymbol{G} = \begin{bmatrix} \dfrac{1}{2\rho}\left\{\left[\dfrac{4}{3}\mu + \theta\right]\dfrac{\partial\mathbf{R}_1^n}{\partial n} - \dfrac{2c\theta}{\gamma-1}\dfrac{\partial\mathbf{R}_2^n}{\partial n} + \left[-\dfrac{4}{3}\mu + \theta\right]\dfrac{\partial\mathbf{R}_4^n}{\partial n}\right\} \\ -\dfrac{\theta}{2\rho c}\left(\dfrac{\partial\mathbf{R}_1^n}{\partial n} + \dfrac{\partial\mathbf{R}_4^n}{\partial n} - \dfrac{2c}{\gamma-1}\dfrac{\partial\mathbf{R}_2^n}{\partial n}\right) \\ \dfrac{\mu}{\rho}\dfrac{\partial\mathbf{R}_3^n}{\partial n} \\ \dfrac{1}{2\rho}\left\{\left[-\dfrac{4}{3}\mu + \theta\right]\dfrac{\partial\mathbf{R}_1^n}{\partial n} - \dfrac{2c\theta}{\gamma-1}\dfrac{\partial\mathbf{R}_2^n}{\partial n} + \left[\dfrac{4}{3}\mu + \theta\right]\dfrac{\partial\mathbf{R}_4^n}{\partial n}\right\} \end{bmatrix} \tag{20}$$

$$\boldsymbol{\mathcal{R}}_- = \begin{bmatrix} |\lambda_1| & 0 & 0 & 0 \\ 0 & |\lambda_2| & 0 & 0 \\ 0 & 0 & |\lambda_3| & 0 \\ 0 & 0 & 0 & (1-\alpha)|\lambda_4| \end{bmatrix}, \qquad \boldsymbol{\mathcal{G}}_- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{\mathcal{R}}_+ = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha|\lambda_4| \end{bmatrix}, \qquad \boldsymbol{\mathcal{G}}_+ = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

We have adapted the formulation for the treatment of a viscous-inviscid interface as

$$\begin{cases} \dfrac{\partial \tilde{\mathbf{Q}}_I}{\partial t} + \nabla \cdot \tilde{\mathbb{F}}^i(\tilde{\mathbf{Q}}_I) = -\beta \boldsymbol{\mathcal{S}} \left[\boldsymbol{\mathcal{R}}_\pm(\mathbf{R}_I - \mathbf{R}_V) - \boldsymbol{\mathcal{G}}_\pm \mathbf{G}_V \right] \quad \text{in } \Omega_I \\ \dfrac{\partial \tilde{\mathbf{Q}}_V}{\partial t} + \nabla \cdot \mathbb{F}^i(\tilde{\mathbf{Q}}_V) = \nabla \cdot \mathbb{F}^v(\tilde{\mathbf{Q}}_V) + \beta \boldsymbol{\mathcal{S}} \left[\boldsymbol{\mathcal{R}}_\pm(\mathbf{R}_I - \mathbf{R}_V) - \boldsymbol{\mathcal{G}}_\pm \mathbf{G}_V \right] \quad \text{in } \Omega_V \end{cases} \tag{21}$$

In this formulation there are no viscous fluxes in Ω_I, but we need to impose interface conditions deriving from the viscous terms in Ω_V. Otherwise, in the inviscid region there will not be boundary conditions associated with the viscous terms in the viscous region.

3.3 Turbulence interface conditions

In the turbulent equations, two different approaches are proposed when treating the interface. In the first one the complete set of turbulent equations are used in the viscous domain, while in the Euler region a reduced turbulent model is used where only the advection is considered while the viscous and source terms are neglected. In this case there is no need for interface conditions between the viscous and inviscid zone. This can be written as

$$\begin{cases} \dfrac{\partial \mathbf{Q}_I}{\partial t} + \nabla \cdot \mathbb{F}^i(\mathbf{Q}_I) &= \mathbf{0}, \quad \text{in } \Omega_I, \\ \dfrac{\partial \mathbf{Q}_V}{\partial t} + \nabla \cdot \mathbb{F}^i(\mathbf{Q}_V) &= \nabla \cdot \mathbb{F}^v(\mathbf{Q}_V) + \mathbf{S}, \quad \text{in } \Omega_V. \end{cases} \tag{22}$$

In the second case the Euler region is turbulence free and on the interface far-field turbulent boundary condition are applied as described in 2.3. The new system of equation is written

$$\begin{cases} \dfrac{\partial \tilde{\mathbf{Q}}_I}{\partial t} + \nabla \cdot \tilde{\mathbb{F}}^i(\tilde{\mathbf{Q}}_I) &= \mathbf{0}, \quad \text{in } \Omega_I, \\ \dfrac{\partial \mathbf{Q}_V}{\partial t} + \nabla \cdot \mathbb{F}^i(\mathbf{Q}_V) &= \nabla \cdot \mathbb{F}^v(\mathbf{Q}_V) + \mathbf{S}, \quad \text{in } \Omega_V. \end{cases} \tag{23}$$

4 RESULTS

We analyse the performance of strong and weak formulation of the interface condition and the effect of the position of the interface in the accuracy of the solution for two turbulent flows test cases, a flatplate and an airfoil.

4.1 Turbulent flat plate

We consider first the compressible turbulent flow past a flat plate. The geometry of the problem is very simple and this permits the generation of meshes where the definition of an interface is straightforward. The flow conditions are those of a flow parallel to the plate corresponding to free-stream conditions given by $M_\infty = 0.1$ and $Re = 10^7$. The

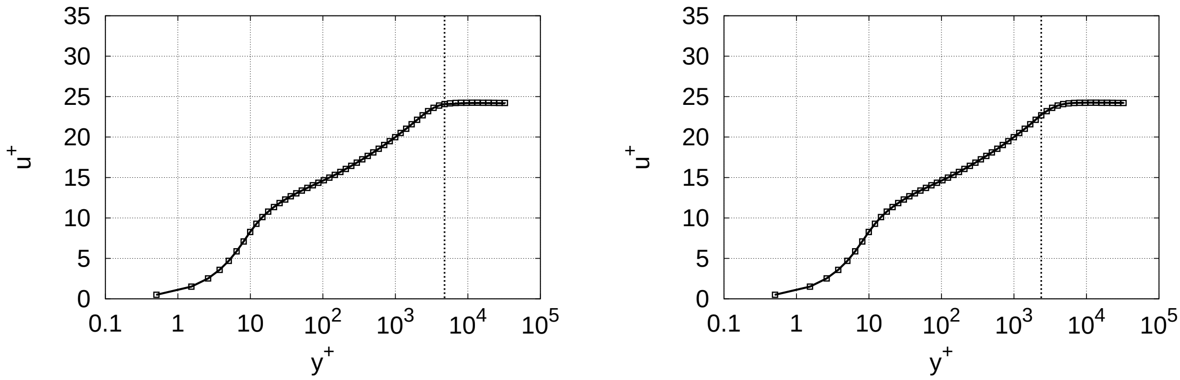

Figure 2: Velocity profile obtained by the zonal solver imposing the strong matching at the interface with $y/\delta = 1$ (left) and $y/\delta = 0.5$ (right). Reference solution (squares), zonal solution (bold line) and interface (dotted line).

boundary corresponding to the flat plate is treated as a viscous wall and inflow and outflow boundary conditions are imposed on the left and right boundaries. The top boundary is treated as a symmetry boundary.

We undertake a preliminary study to assess the accuracy and performance of the zonal approaches and their sensitivity to the location of the interface. The accuracy of the computed zonal solution will be determined by comparison with that obtained with the standard viscous solver, which is our target.

The zonal interface is defined through a line that is placed at a certain distance from the wall. The reference line has been found from the complete turbulent solution so that its distance from the wall is always at the position where $u = 0.999\, U_\infty$. The position of the zonal interface is then defined the ratio between the normal distance from the wall y and the height of the boundary layer δ (i.e. for $y/\delta = 1$ the interface is placed at the edge of the boundary layer).

We want to access the performance of the zonal approaches in terms of computational time and accuracy of the solution and how it is affected by the position of the zonal interface. The computational time of the simulation depends upon the ratio between the number of viscous and inviscid faces, as well as from the convergence rate of the solution. The accuracy of the zonal solution is accessed calculating the error on the friction coefficient C_f on the flat plate as

$$\epsilon_{C_f} = \sum \frac{|C_{f_V} - C_{f_Z}|}{\max(|C_{f_V}|)} \tag{24}$$

The velocity profiles calculated using different approaches for various locations of the interface are shown in figures 2 and 3. The results obtained when the interface is located on the edge or out of the boundary layer are fairly accurate, but the accuracy deteriorates if the interface is taken inside the boundary layer. When the interface is placed at a position $y/\delta < 0.5$, the strong approach does not converge. However, a stable result

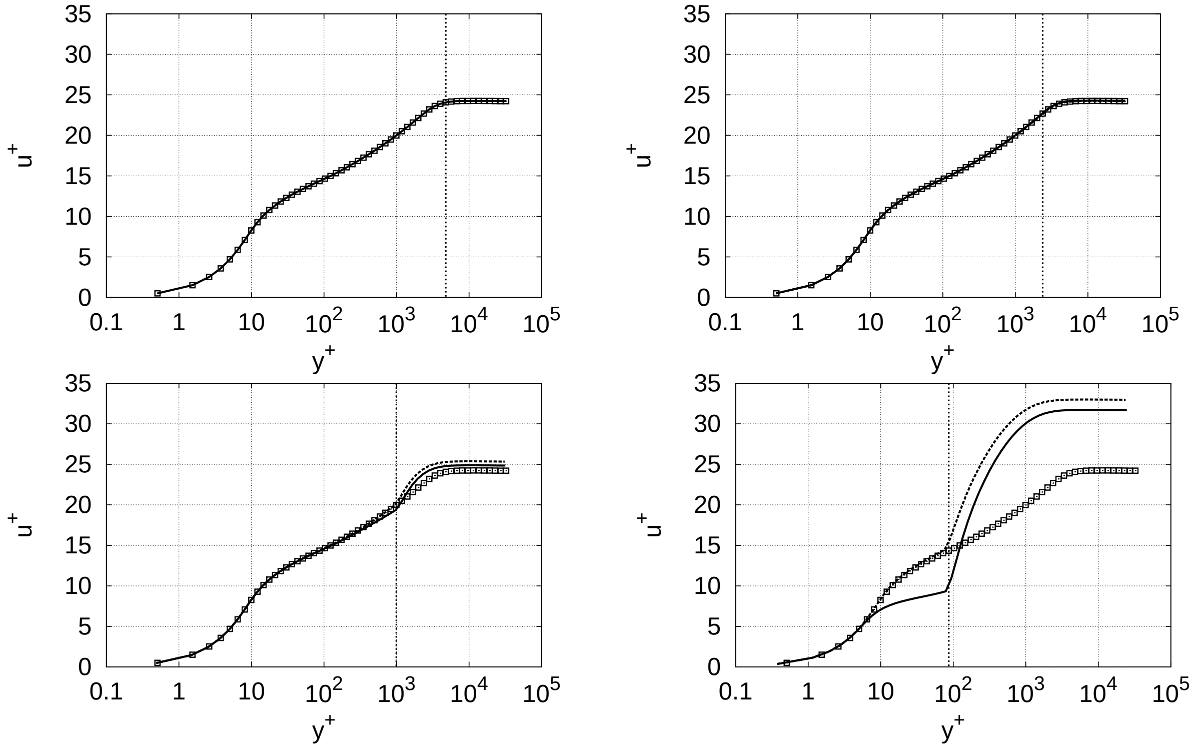

Figure 3: Velocity profile obtained by the zonal solver imposing the weak matching at the interface with $y/\delta = 1$ (top left), $y/\delta = 0.5$ (top right), $y/\delta = 0.2$ (bottom left) and $y/\delta = 0.01$ (bottom right). Reference solution (squares), zonal solution without source terms in the Euler region (bold line), zonal solution without turbulence model in the Euler region (dashed line) and interface (dotted line).

is obtained with the weak approach, even though the convergence rate is significantly reduced. If the interface is located within the logarithmic region the difference between the two treatment of the turbulent term can be noticed. The velocity profile remains close to the reference simulation when the far-field boundary are imposed directly at the interface and the error is present only in the Euler region. When only the turbulent source terms are avoided in the outer region, the solution presents a larger error that propagates also inside the viscous region.

The computational cost of the simulations has been obtained timing each part of the code and extracting the time cost for the solution of each different edge (i.e. Euler, turbulent or interface). Since the modifications that we are introducing affect only the interface edges, the time cost for internal and boundary faces does not change with the zonal approach if the same method to treat the turbulent interface is used. A considerable difference is instead found in the solution of the interface edges. The solution of the interface edge using the strong approach takes 19% more than the time to solve a turbulent face, and the weak approach takes a substantial 206%. However, the number of interface edges is usually very small compared to the total number of edges and therefore the

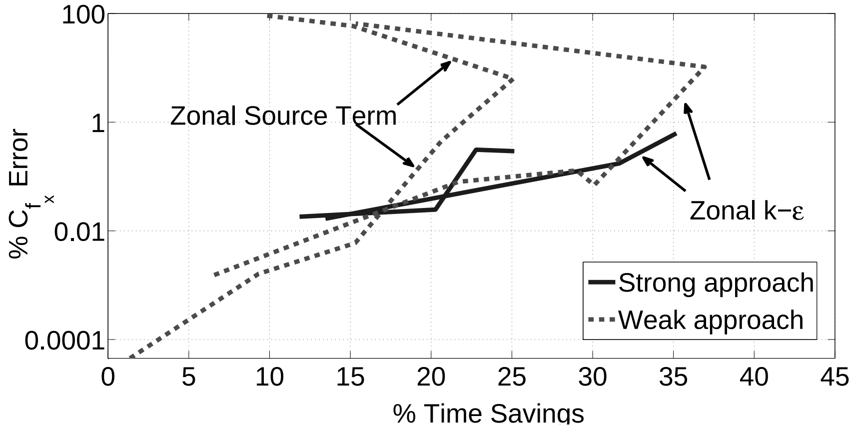

Figure 4: Saving time versus accuracy for different zonal approaches.

influence of this penalty is negligible for large meshes.

The performance of the various method is shown in figure 4. As expected, the two different treatments of the turbulent interface return the same accuracy in terms of C_f error but the time saving is clearly increased in the case of an Euler region free from turbulent equations (*zonal* k-ε). Although the strong approach does not converge for value of $y/\delta < 0.5$, in the case of low y/δ the time saving of the weak approach is affected by the reduced rate of convergence of the solution. Therefore, if we accept an error of about 1% on C_f, the curves present an optimal performance value of the method when $y/\delta \approx 0.5$ with a maximum time saving of 35% respect to a complete turbulent solution.

4.2 Turbulent Flow past a RAE2882 aerofoil

A test case with turbulent compressible flow over a RAE2822 aerofoil was employed to test the zonal solvers. The original data is $M_\infty = 0.8$, $\alpha = 1°$ and $Re = 6.5 \times 10^6$.

The aerofoil test introduces a more difficult geometry with respect to the flatplate and it creates a complex viscous wake. The interface is defined at a distance from the aerofoil that increases in the downstream direction. The wake is not completely included in the viscous region.

In figure 5 the mesh is displayed, together with the pressure distribution for complete viscous and zonal solutions. The pressure distribution do not visibly differ from the complete viscous solution and it presents a smooth transition through the interface even on the wake.

To better appreciate the dramatic discontinuity introduced in the flowfield, the kinetic energy k is shown in figure 6. In case the turbulent equations are removed from the inviscid region, the turbulence is neglected at a certain distance from the airfoil. As a result, large values of turbulent variables are present where the flow passes from the viscous to the Euler region. Although these zone may affect strongly the stability of the numerical method, both approaches are able to provide a stable and smooth solution.

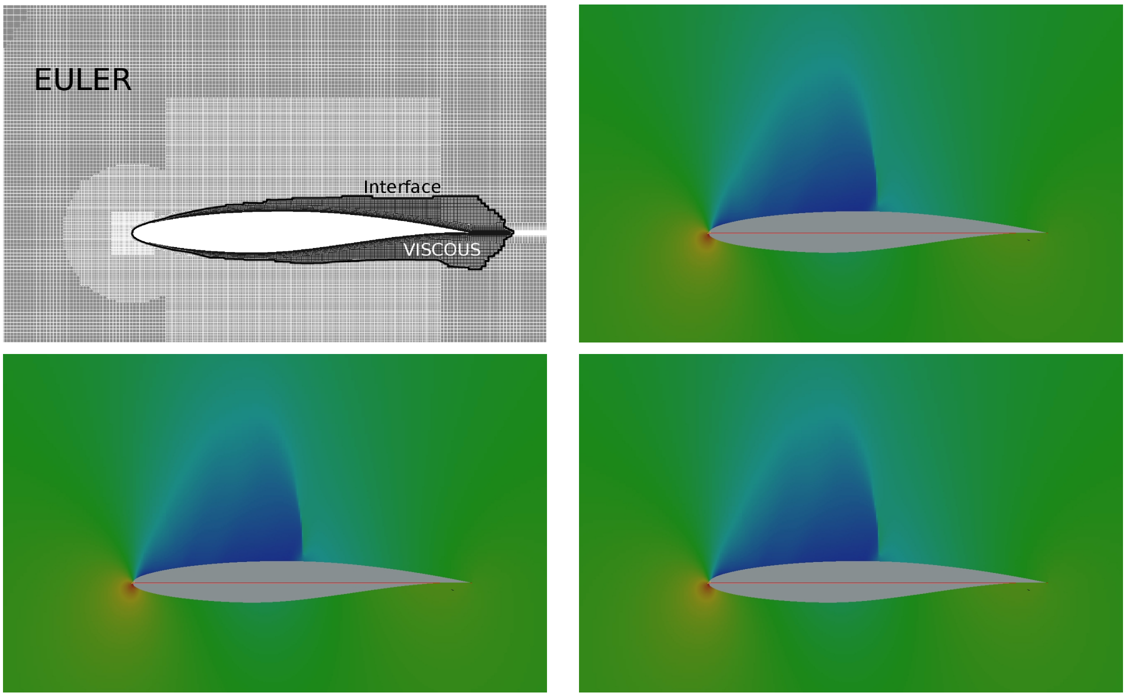

Figure 5: Enlargement of the mesh around the aerofoil and zones definition (top left). Pressure distribution: complete turbulent (top right), strong approach (bottom left), weak approach (bottom right).

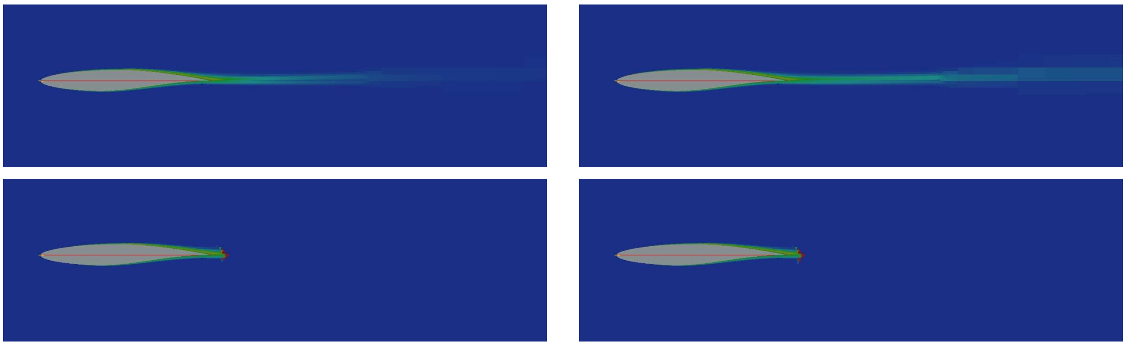

Figure 6: Turbulent kinetic energy: complete turbulent solution (top left), strong approach with zonal source term (top right), strong approach with zonal k-ε (bottom left), weak approach with zonal k-ε (bottom right)

The differences among various approaches are more evident when we analyse two relevant quantities on the surface of the aerofoil. Errors of the pressure and friction coefficients over the aerofoil and time savings are compared in table 1. Here we fix the accuracy requirement of C_{f_x} at about 10% respect to the complete viscous solution. The weak and the strong approaches present similar performance and the difference between time savings is reduced compared with the timings on the flatplate. Since the mesh is larger, the number of interface faces is always below 0.005% the total number of faces and therefore the time penalty introduced by the weak approach is strongly reduced.

To guarantee the required accuracy, the zonal method that uses a reduced turbulent model in the inviscid region needs less cells than the other approach but the time to complete the simulation is larger. Although the method that considers the inviscid region turbulence free requires a viscous region that is larger, the time of the simulation is diminished.

Tests show that in the zonal k-ε approach, the accuracy is more influenced by a reduction of the viscous zone at the beginning of the aerofoil rather than in the wake. This effect may be explained by the observation that the turbulence model requires some physical space to develop and therefore a certain length of viscous region has to be guaranteed.

	$N_{\text{faces}_I}/N_{\text{faces}_V}$	ϵ_{C_p}	$\epsilon_{C_{f_x}}$	Time Savings
Strong - Zonal Source Term	5.8	0.58%	10.97%	24.58%
Weak - Zonal Source Term	5.8	0.44%	9.58%	24.07%
Strong - Zonal k-ε	3.4	0.33%	10.54%	44.77%
Weak - Zonal k-ε	3.4	0.40%	10.56%	43.85%

Table 1: Performance and accuracy of various zonal methods.

5 CONCLUSIONS

A zonal method that uses two different approaches for the treatment of the interface between inviscid/viscous regions has been presented. The treatment of the interface for both approaches has been described and two different method for the implementation of the turbulent model have been tested.

Here we have been focused on the implementation of a strong approach where the variables are directly imposed on the interface and a weak approach where the interface is treated within the Navier–Stokes equations.

The two methods have been successfully used for the solution of the turbulent flow over a flat plate and a subsonic airfoil. Accuracy and performance analysis are carried out in the tests. A zonal solution is 43% more cheaper than a complete viscous solution with a loss of accuracy on the friction coefficient of 10%.

Even though it is not possible to identify the best zonal method in terms of accuracy and time savings since both present similar results, the flatplate test shows that the weak approach is more stable in case the interface is placed within the boundary layer. The results are encouraging and the zonal solver appears to be a suitable tool to speed up the computations.

6 ACKNOWLEDGEMENTS

The authors would like to thank Jan S. Hesthaven and Alfio Quarteroni for their availability to discuss their methods.

REFERENCES

[1] Quarteroni A, Valli A. *Domain Decomposition Methods for Partial Differential Equations.* Oxford University Press, 1999.

[2] Domain Decomposition Method page.
www.ddm.org [9 March 2010]

[3] Yang Z, Sankar LN, Smith M, Bauchau O. Recent improvements to a hybrid method for rotors in forward flight. *Journal of Aircraft*, 2002; **39**:804–812.

[4] Cai X-C, Paraschivoiu M, Sarkis M. An explicit multi-model compressible flow formulation based on the full potential equations and the Euler equations on 3D unstructured meshes. In *11th International Conference on DDMs*, 1999.

[5] Schmitz S, Chattot J-J. A parallelized coupled Navier–Stokes/vortex-panel solver. *Journal of Solar Energy Engineering*, 2005; November,**127**:475–487.

[6] Su J. A viscous-inviscid zonal method for compressible and incompressible viscous flows. *AIAA Paper 2005-5340*, 2005.

[7] Tessicini F, Temmerman L, Leschziner MA. Approximate near-wall treatments based on zonal and hybrid RANS/LES methods for LES at high Reynolds numbers. *International Journal of Heat and Fluid Flow*, 2006; **27**:789–799.

[8] Wilcox DC. *Turbulence Modeling for CFD.* DCW Industries, 1993.

[9] Schmatz S, Hirschel EH. Zonal solution for airfoils using Euler, boundary layer and Navier–Stokes equations. *Applications of Computational Fluid Dynamics in Aeronautics - AGARD conference, 412*, 1986; April(20):1–13.

[10] Mensink C, Deconinck H. A 2D multiblock method for viscous and inviscid flow computations. *Numerical Methods for Fluid Dynamics*, 1993; vol IV:509–515.

[11] Ekaterinaris JA, Cricelli AS, Platzer MF. A zonal method for unsteady, viscous, compressible airfoil flows. *Journal of Fluids and Structures*, 1994; **8**(1):107–123.

[12] Arina R, Canuto C. A χ-formulation of the viscous-inviscid domain decomposition for the Euler/Navier–Stokes equations. *Contemporary Mathematics*, 1994; **180**:453–458.

[13] Quarteroni A, Stolcis L. Homogeneous and heterogeneous domain decomposition for compressible fluid flow at high Reynolds number. *Research Report CRS4-Appmath*, 1995; vol 2.

[14] Hafez H, Shatalov A, Wahba E. Numerical simulations of incompressible aerodynamic flows using viscous/inviscid interaction procedures. *Computer Methods in Applied Mechanics and Engineering*, 2006; **195**:3110–3127.

[15] Hesthaven JS, Gottlieb D. A stable penalty method for the compressible Navier–Stokes equations. I. Open boundary conditions. *SIAM Journal on Scientific Computing*, 1996; **17**(3):579–612.

[16] Hesthaven JS. A stable penalty method for the compressible Navier–Stokes equations. II. One dimensional domain decomposition schemes. *SIAM Journal on Scientific Computing*, 1997; **18**(3):658–685.

[17] Hesthaven JS. A stable penalty method for the compressible Navier–Stokes equations. III. Multi-dimensional domain decomposition schemes. *SIAM Journal on Scientific Computing*, 1998; **20**(1):62–93.

[18] Roe PL. *Numerical Methods in Aeronautical Fluid Dynamics.* Elsevier Science & Technology Books, 1982.

[19] Lam CKG, Bremhorst K. A modified form of the k-ε model for predicting wall turbulence. *ASME Journal of Fluids Engineering*, 1981; **103**:456–460.

[20] Henkes R, Hoogendoorn CJ. Comparison of turbulence models for the natural convection boundary layer along a heated vertical plate. *International Journal of Heat and Mass Transfer*, 1989; **32**:157–169.

[21] Knaus H, Richter S, Uterberger S, Schnell U, Maier H, Hein KRG. On the application of different turbulence models for the computation of fluid flow and combustion processes in small scale wood heaters. *Experimental Thermal and Fluid Science*, 2000; **21**(1–3):99–108.

[22] Roe PL. Characteristic-based schemes for the Euler equations. *Annual Review of Fluid Mechanics*, 1986; **18**(1):337–365.

[23] Laney CB. *Computational Gas Dynamics.* Cambridge University Press, 1998.

[24] Hirsch C. *Numerical Computation of Internal and External Flows.* vol II; John Wiley & Sons, 1990.

[25] Van Albada GD, Van Leer B, Roberts WW. A comparative study of computational methods in cosmic gas dynamics. *Astronomy & Astrophysics*, 2002. **108**:76–84.

[26] Goldberg U, Palaniswamy S. The k-ε-μ turbulence closure model. *Computer Methods in Applied Mechanics and Engineering*, 1999. **179**(1–2):139–149.

[27] Thompson KW. Time dependent boundary conditions for hyperbolic systems, part II. *Journal of Computational Physics*, 1990; **89**:439–461.