Math. Model. Nat. Phenom.
Vol. 6, No. 3, 2011, pp.189-212
DOI: 10.1051/mmnp/20116308

Aerodynamic Computations Using a Finite Volume Method with an HLLC Numerical Flux Function

L. Remaki*, O. Hassan and K. Morgan

School of Engineering, Swansea University, Swansea SA2 8PP, Wales, UK

Abstract. A finite volume method for the simulation of compressible aerodynamic flows is described. Stabilisation and shock capturing is achieved by the use of an HLLC consistent numerical flux function, with acoustic wave improvement. The method is implemented on an unstructured hybrid mesh in three dimensions. A solution of higher order accuracy is obtained by reconstruction, using an iteratively corrected least squares process, and by a new limiting procedure. The numerical performance of the complete approach is demonstrated by considering its application to the simulation of steady turbulent transonic flow over an ONERA M6 wing and to a steady inviscid supersonic flow over a modern military aircraft configuration.

Key words: aerodynamic flows, finite volume method, HLLC flux function, solution reconstruction, slope limiting.
AMS subject classification: 65M08, 65M50, 76J20

1. Introduction

Unstructured mesh methods are now widely employed for the simulation of general industrial aerodynamic flow problems. This is mainly due to the ease and rapidity with which complex geometrical domains may be meshed, using fully automatic mesh generation procedures. A wide variety of unstructured mesh flow solution algorithms have been proposed, employing finite volume or finite element methods and different approaches for stabilisation and shock capturing [10]–[6]. A major

*Corresponding author. E-mail: l.remaki@swansea.ac.uk

Article published by EDP Sciences and available at http://www.mmnp-journal.org or http://dx.doi.org/10.1051/mmnp/20116308

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

difficulty, that is often encountered by an analyst employing these methods, is the maintenance of an acceptable level of accuracy while, simultaneously, ensuring robustness over a range of flow speeds.

In this paper, the first step in the discretisation process is the triangulation of the computational boundaries and this is accomplished using an advancing front approach [21]. For inviscid flows, the computational domain is discretised using isotropic tetrahedral elements created by a Delaunay procedure with automatic point creation [31]. For viscous flows, the advancing layers method [23, 7] is used first to generate layers of stretched tetrahedra in the vicinity of solid surfaces and then the remainder of the computational domain is discretised using isotropic tetrahedral elements. Using the structure in the mesh generated by the advancing layers method, the stretched tetrahedra are merged where possible to create a consistent hybrid mesh of prisms and tetrahedra [24]. On these meshes, the compressible Navier–Stokes equations are discretised using a cell–vertex finite volume approach. Stabilisation of the resulting centered discreisation and shock capturing is achieved by replacing the physical inviscid flux function by a consistent numerical upwind flux function [19]. The numerical inviscid flux function employed here is computed using the parameter free HLLC Riemann solver [29], which is a modification of that introduced in the original HLL scheme [5, 28]. A modification to the traditional HLLC acoustic wave speeds is introduced, to improve stability at transition from subsonic to supersonic flow conditions. A higher order extension is achieved in the traditional manner of solution reconstruction and limiting [8]. Galerkin [12] and least squares methods [15] have been widely used in the reconstruction process and, here, an iteratively corrected least squares approach is employed to achieve an accurate estimation of the solution gradient. Stability and conservation concepts are then used to develop a differentiable slope limiter that is a modification of that introduced initially by Barth and Jespersen [2]. The numerical performance of the complete procedure is illustrated by considering the simulation of transonic turbulent flow over a wing and supersonic inviscid flow over a generic modern military aircraft configuration.

2. Governing Equations

The Favre averaged equations governing three dimensional turbulent compressible flow are expressed, relative to a Cartesian (x_1, x_2, x_3) coordinate system, over a fixed volume $\mathscr{V}$ with a closed surface $\mathscr{S}$, in the integral form

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathscr{V}} \boldsymbol{Q}\,\mathrm{d}\mathscr{V} = \int_{\mathscr{S}} \boldsymbol{F}^{\alpha}(\boldsymbol{Q})\,n^{\alpha}\,\mathrm{d}\mathscr{S} - \int_{\mathscr{S}} \boldsymbol{G}^{\alpha}(\boldsymbol{Q})\,n^{\alpha}\,\mathrm{d}\mathscr{S} \qquad \alpha = 1,2,3 \tag{2.1}$$

where the summation convention is employed and $\boldsymbol{n} = (n^1, n^2, n^3)$ denotes the unit outward normal vector to $\mathscr{S}$. In this equation, the unknown $\boldsymbol{Q}$, the inviscid flux vectors $\boldsymbol{F}^{\alpha}$ and the viscous flux vectors $\boldsymbol{G}^{\alpha}$ are defined by

$$\boldsymbol{Q} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_2 \\ E \end{pmatrix} \qquad \boldsymbol{F}^{\alpha} = \begin{pmatrix} \rho u_{\alpha} \\ \rho u_1 u_{\alpha} + p\delta_{\alpha_1} \\ \rho u_2 u_{\alpha} + p\delta_{\alpha_2} \\ \rho u_2 u_{\alpha} + p\delta_{\alpha_3} \\ (E+p)u_{\alpha} \end{pmatrix} \qquad \boldsymbol{G}^{\alpha} = \begin{pmatrix} 0 \\ \tau_{1\alpha} \\ \tau_{2\alpha} \\ \tau_{3\alpha} \\ u_{\beta}\tau_{\beta\alpha} - q_{\alpha} \end{pmatrix} \tag{2.2}$$

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

where ρ, p and E denote the averaged density, pressure and total energy of the fluid respectively, u_α is the averaged velocity of the fluid in direction x_α, t is the time and $\delta_{\alpha,\beta}$ is the Kronecker delta. The averaged deviatoric stress tensor is defined by

$$\tau_{\beta\alpha} = -\frac{2}{3}\mu\frac{\partial u_k}{\partial x_k}\delta_{\beta\alpha} + \mu\left(\frac{\partial u_\beta}{\partial x_\alpha} + \frac{\partial u_\alpha}{\partial x_\beta}\right) \tag{2.3}$$

and the averaged heat flux is given by

$$q_\alpha = -k\frac{\partial T}{\partial x_\beta} \tag{2.4}$$

In these equations, μ denotes the sum of the laminar and the turbulent viscosity, k is the sum of the laminar and the turbulent thermal conductivity and T is the averaged absolute temperature. The equation set is completed by the addition of the perfect gas equations of state. The laminar viscosity is assumed to vary with temperature according to the Sutherland Law [32] and the distribution of the turbulent viscosity is determined using a one equation Spalart–Allmaras model [26]. The Prandtl number is taken to be constant. Steady state solutions of the resulting equation set are sought in a fixed spatial computational domain Ω.

3. Finite Volume Method

On the generated consistent hybrid primal mesh, nodes are located at the vertices of the elements and the spatial discretisation of equation (2.1) is accomplished using a cell vertex finite volume method. This requires the construction of a dual mesh, in which each cell of the dual is associated with a single node of the primal mesh. For those regions in which the primal mesh consists only of isotropic tetrahedral cells, a median dual mesh is constructed by connecting cell edge midpoints, cell centroids and cell face centroids, such that only one node is present within each dual mesh cell [24]. With this strategy, each node I of the domain mesh is associated with a volume Ω_I of the dual mesh. The boundary surface of the volume Ω_I is denoted by Γ_I. Each edge of the domain mesh is associated with a segment of the dual mesh interface between the nodes connected to the edge. This segment is a surface constructed from triangular facets, where each facet is connected to the midpoint of the edge, a neighboring element centroid and the centroid of an element face connected to the edge, as illustrated in Figure 1. The midpoint of the edge between node I and J is denoted by $\boldsymbol{x}_m^{IJ}$, the centroid of the face with vertices I, J and K is denoted by $\boldsymbol{x}_s^{IJK}$ and the element centroid is designated by $\boldsymbol{x}_c$. The bold lines on the dual mesh in this figure illustrate the boundaries between the edges with which the dual mesh segment is associated. With this dual mesh definition, the volume Ω_I can be viewed as being constructed in terms of a set of tetrahedra, as illustrated for a typical interior node I in Figure 2. The surface of the dual mesh cell surrounding node I is defined in terms of the closed set of planar triangular facets Γ_I^K, where each facet only touches a single edge of the domain mesh. The set of facets touching the edge between nodes I and J is denoted by Γ_{IJ}.

In general, the median dual approach cannot be used for the hybrid elements produced by merging the stretched tetrahedra generated by the advancing layers method. This is because cells

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

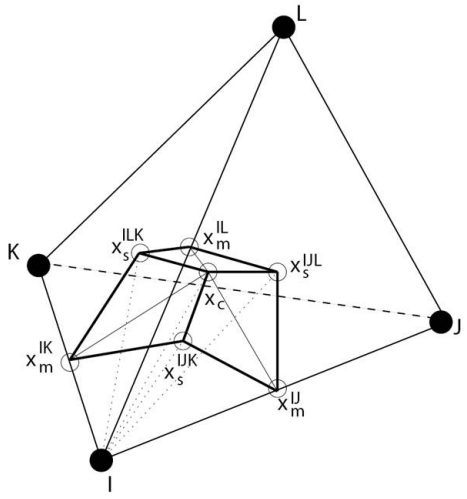


Figure 1: Illustration of that part of the dual mesh cell surrounding node I that is contained within a tetrahedral cell.

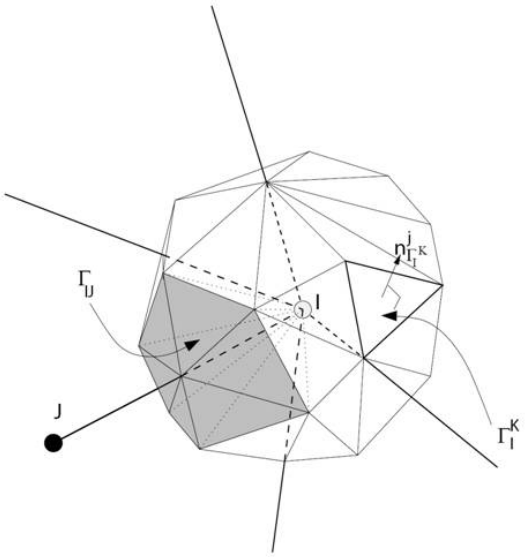


Figure 2: Illustration of the dual mesh cell surrounding an internal node I.

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

created in this way may be warped so severely that a vertex can lie outside the corresponding median dual cell. This may occur in regions of high geometry curvature or at the interface between the hybrid and isotropic meshes. To overcome this problem, the information contained in the primal tetrahedral mesh is used to ensure that the topology of the control volume cells is valid [25, 24].

Equation (2.1) is applied to each cell Ω_I of the dual mesh in turn. To perform the numerical integration of the inviscid fluxes over the surface Γ_I of this cell, a set of coefficients is calculated for each edge using the dual mesh segment associated with the edge. The values of these coefficients for an internal edge are evaluated as [19, 14]

$$n_{IJ}^{\alpha} = \sum_{K \in \Gamma_{IJ}} A_{\Gamma_I^K} n_{\Gamma_I^K}^{\alpha} \tag{3.1}$$

where $A_{\Gamma_I^K}$ is the area of facet Γ_I^K and $n_{\Gamma_I^K}^{\alpha}$ is the component, in direction x_α, of the outward unit normal vector of the facet from the viewpoint of node I. The integral of the inviscid flux over the surface Γ_I is then approximated, using the summation of edge contributions, as [19, 22]

$$\int_{\Gamma_I} \boldsymbol{F}^{\alpha} n^{\alpha} dS \approx \sum_{J \in \Lambda_I} \tilde{\boldsymbol{F}}_{IJ} \tag{3.2}$$

where Λ_I denotes the set of nodes connected to node I by an edge in the domain mesh. Here,

$$\tilde{\boldsymbol{F}}_{IJ} = \begin{pmatrix} \rho q_{IJ} \\ \rho u_1 q_{IJ} + p n_{IJ}^1 \\ \rho u_2 q_{IJ} + p n_{IJ}^2 \\ \rho u_3 q_{IJ} + p n_{IJ}^3 \\ (E+p) q_{IJ} \end{pmatrix} \tag{3.3}$$

is a consistent numerical inviscid flux function and

$$q_{IJ} = n_{IJ}^{\alpha} \, (u_\alpha)_I \tag{3.4}$$

is the velocity in the direction of the edge connecting nodes I and J. Similarly, the integral of the viscous flux over the surface Γ_I is approximated as

$$\int_{\Gamma_I} \boldsymbol{G}^{\alpha} n^{\alpha} dS \approx \sum_{J \in \Lambda_I} \boldsymbol{G}_{IJ} = \sum_{J \in \Lambda_I} \frac{n_{IJ}^{\alpha}}{2} \left(\boldsymbol{G}_I^{\alpha} + \boldsymbol{G}_J^{\alpha} \right) \tag{3.5}$$

where the physical viscous flux function is employed [24].

4. HLLC Numerical Inviscid Flux Function

Stabilisation of the resulting discretisation is achieved by replacing the physical inviscid flux function in equation (3.2) by a consistent numerical upwind flux function [19] and the numerical flux

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

function employed here is computed using the HLLC Riemann solver [29, 28]. With this solver, for the edge connecting nodes I and J, the central idea is to assume that the solution of the Riemann problem consists of three waves separating the four constant states $\boldsymbol{Q}_I$, $\boldsymbol{Q}^{\mathscr{L}}_{IJ}$, $\boldsymbol{Q}^{\mathscr{R}}_{IJ}$ and $\boldsymbol{Q}_J$, as illustrated in Figure 3. In this case, the solution of the Riemann problem for the edge consists of a contact wave and two acoustic waves, which may be either shocks or expansion fans. Godunov's method [8] is employed to construct the approximate solution by averaging intermediate states in the exact solution, respecting certain principles, such as exactly resolving isolated shocks and contact discontinuities. The respective wave speeds $S^{\mathscr{L}}$, $S^{\mathscr{M}}$ and $S^{\mathscr{R}}$ are determined by averaging, in an appropriate manner [28]. In this way, the entries in the vector $\boldsymbol{Q}^{\mathscr{L}}_{IJ}$ are evaluated as

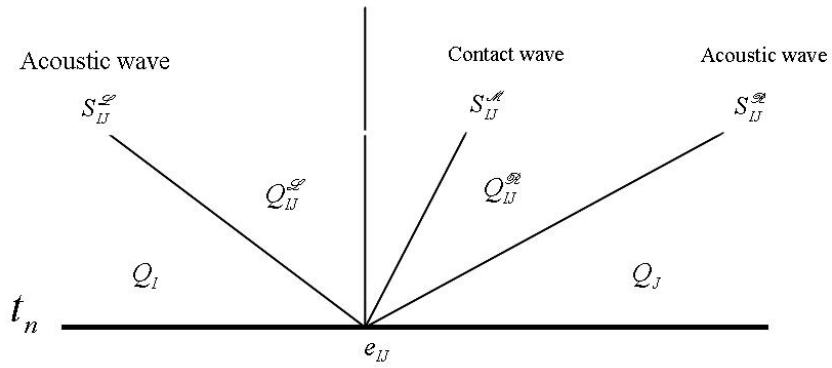


Figure 3: Schematic illustration of the HLLC solution of the Riemann problem for the edge e_{IJ} connecting nodes I and J.

$$\begin{aligned}
\rho^{\mathscr{L}}_{IJ} &= \rho_I \frac{(S^{\mathscr{L}}_{IJ} - q^{\mathscr{L}}_{IJ})}{S^{\mathscr{L}}_{IJ} - S^{\mathscr{M}}_{IJ}} \\
(\rho u_\alpha)^{\mathscr{L}}_{IJ} &= \frac{(S^{\mathscr{L}}_{IJ} - q^{\mathscr{L}}_{IJ})\rho_I u_{\alpha I} + (p^*_{IJ} - p_I) n^\alpha_{IJ}}{S^{\mathscr{L}}_{IJ} - S^{\mathscr{M}}_{IJ}} \\
E^{\mathscr{L}}_{IJ} &= \frac{(S^{\mathscr{L}}_{IJ} - q^{\mathscr{L}}_{IJ})E_I - p_I q^{\mathscr{L}}_{IJ} + p^*_{IJ} S^{\mathscr{M}}_{IJ}}{S^{\mathscr{L}}_{IJ} - S^{\mathscr{M}}_{IJ}}
\end{aligned} \tag{4.1}$$

and the entries in $\boldsymbol{Q}^{\mathscr{R}}_{IJ}$ are defined similarly. In these equations,

$$p^*_{IJ} = \rho_I (q^{\mathscr{L}}_{IJ} - S^{\mathscr{L}}_{IJ})(q^{\mathscr{L}}_{IJ} - S^{\mathscr{M}}_{IJ}) + p_I \tag{4.2}$$

and

$$q^{\mathscr{L}}_{IJ} = (u_\alpha)_I \, n^\alpha_{IJ} \qquad\qquad q^{\mathscr{R}}_{IJ} = (u_\alpha)_J \, n^\alpha_{IJ} \tag{4.3}$$

while the speed of the contact wave is computed as

$$S^{\mathscr{M}}_{IJ} = \frac{\rho_J q^{\mathscr{R}}_{IJ}(S^{\mathscr{R}}_{IJ} - q^{\mathscr{R}}_{IJ}) - \rho_I q^{\mathscr{L}}_{IJ}(S^{\mathscr{L}}_{IJ} - q^{\mathscr{L}}_{IJ}) + p_I - p_J}{\rho_J (S^{\mathscr{R}}_{IJ} - q^{\mathscr{R}}_{IJ}) - \rho_I (S^{\mathscr{L}}_{IJ} - q^{\mathscr{L}}_{IJ})} \tag{4.4}$$

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

When the acoustic wave speeds $S_{IJ}^{\mathscr{L}}$ and $S_{IJ}^{\mathscr{R}}$ are defined, the HLLC numerical inviscid flux function, for use in equation (3.2), is evaluated as

$$\tilde{\boldsymbol{F}}_{IJ}^{HLLC} = \begin{cases} \boldsymbol{F}(\boldsymbol{Q}_I) & \text{if } S_{IJ}^{\mathscr{L}} > 0 \\ \boldsymbol{F}_{IJ}^{\mathscr{L}} & \text{if } S_{IJ}^{\mathscr{L}} \leq 0 < S_{IJ}^{\mathscr{M}} \\ \boldsymbol{F}_{IJ}^{\mathscr{R}} & \text{if } S_{IJ}^{\mathscr{M}} \leq 0 \leq S_{IJ}^{\mathscr{R}} \\ \boldsymbol{F}(\boldsymbol{Q}_J) & \text{if } S_{IJ}^{\mathscr{R}} < 0 \end{cases} \tag{4.5}$$

where $\boldsymbol{F}_{IJ}^{\mathscr{L}}$ and $\boldsymbol{F}_{IJ}^{\mathscr{R}}$ are evaluated as

$$\boldsymbol{F}_{IJ}^{\mathscr{L}} = \boldsymbol{F}(\boldsymbol{Q}_I) + S_{IJ}^{\mathscr{L}}(\boldsymbol{Q}_{IJ}^{\mathscr{L}} - \boldsymbol{Q}_I) \qquad \boldsymbol{F}_{IJ}^{\mathscr{R}} = \boldsymbol{F}(\boldsymbol{Q}_J) + S_{IJ}^{\mathscr{R}}(\boldsymbol{Q}_{IJ}^{\mathscr{R}} - \boldsymbol{Q}_J) \tag{4.6}$$

using the Rankine–Hugoniot conditions.

4.1. Acoustic wave speeds

Usually, in the standard implementation of the HLLC numerical flux function, the acoustic wave speeds are determined in the form [28]

$$S_{IJ}^{\mathscr{L}} = \min\left(q_{IJ}^{\mathscr{L}} - c_{IJ}^{\mathscr{L}}, \tilde{q}_{IJ} - \tilde{c}_{IJ}\right) \qquad S_{IJ}^{\mathscr{R}} = \max\left(q_{IJ}^{\mathscr{R}} + c_{IJ}^{\mathscr{R}}, \tilde{q}_{IJ} + \tilde{c}_{IJ}\right) \tag{4.7}$$

where

$$(\tilde{u}_\alpha)_{IJ} = \frac{((\tilde{u}_\alpha)_I + (\tilde{u}_\alpha)_J R_{IJ})}{(1 + R_{IJ})} \qquad \tilde{q}_{IJ} = (\tilde{u}_\alpha)_{IJ}\, n_{IJ}^\alpha \qquad R_{IJ} = \sqrt{\rho_I \rho_J} \tag{4.8}$$

and

$$\tilde{H}_{IJ} = \frac{(H_I + H_J R_{IJ})}{(1 + R_{IJ})} \qquad \tilde{c}_{IJ}^2 = (\gamma - 1)(\tilde{H}_{IJ} - 0.5\,(\tilde{u}_\alpha)_{IJ}\,(\tilde{u}_\alpha)_{IJ}) \tag{4.9}$$

Here, H denotes the enthalpy. With this definition, at a transition from a subsonic state to a supersonic state in the direction of the flow, i.e. $S_{IJ}^{\mathscr{L}} < 0$ and $S_{IJ}^{\mathscr{R}} > 0$, the HLLC flux function is given by equation (4.6). However, when the transition is from a supersonic state to a subsonic state, i.e. $S_{IJ}^{\mathscr{L}} > 0$ and $S_{IJ}^{\mathscr{R}} > 0$, the HLLC flux function is taken as the left value, which represents a propagation of the discontinuity in the initial value to the right. In this case, $q_{IJ}^{\mathscr{R}} - c_{IJ}^{\mathscr{R}} < 0$, which corresponds to a positive acoustic wave on the left and a negative acoustic wave on the right, so that the initial discontinuity does not propagate. It is apparent that, in this case, an average value gives a better representation of the flux at the origin. To achieve this, in the current implementation, the approximations adopted for the acoustic wave speeds are replaced by the expressions

$$\tilde{S}_{IJ}^{\mathscr{L}} = \min\left(S_{IJ}^{\mathscr{L}}, q_{IJ}^{\mathscr{R}} - c_{IJ}^{\mathscr{R}}\right) \qquad \tilde{R}_{IJ}^{\mathscr{R}} = \max\left(S_{IJ}^{\mathscr{R}}, q_{IJ}^{\mathscr{L}} + c_{IJ}^{\mathscr{L}}\right) \tag{4.10}$$

where $S_{IJ}^{\mathscr{L}}$ and $S_{IJ}^{\mathscr{R}}$ are the original values for the acoustic wave speeds given by equation (4.7).

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

4.2. Higher order extension

The formulation that has been described leads to a first order finite volume scheme for the compressible flow equations. A higher order extension may be constructed in the standard form [8], by using reconstructed interface values in the HLLC solver. The method adopted works with a reconstruction for the primitive variables, i.e. for the density, the velocity components and the energy. This means that, for a typical primitive scalar variable, Q, the quantities $Q_{IJ}^{\mathscr{L}}$ and $Q_{IJ}^{\mathscr{R}}$, which are employed in the HLLC formulation of the Riemann problem, are replaced by the expressions

$$\tilde{Q}_{IJ}^{\mathscr{L}} = Q_{IJ}^{\mathscr{L}} + \delta Q_{IJ} \qquad \tilde{Q}_{IJ}^{\mathscr{R}} = Q_{IJ}^{\mathscr{R}} + \delta Q_{JI} \tag{4.11}$$

where δQ_{IJ} and δQ_{JI} denote appropriate approximations that depend upon the computed gradients of Q within the respective cells. To ensure stability of the resulting solution procedure, for the simulation of flows involving discontinuities or steep gradients, these computed gradients must be limited in some fashion [8].

4.2.1. Gradient approximation

The gradient of the primitive variables may be computed in a variety of ways. The Green–Gauss method, which is based upon a contour integral around a control volume, has been widely employed [12]. However, this is generally not exact even for linear functions and its accuracy is highly dependent on shapes of the mesh cells. Methods based upon a least squares procedure are generally better and produce more accurate approximations to the solution gradients on distorted meshes[15].

The objective is to construct an approximation to the gradient of a typical scalar variable Q, at a given vertex I, from the requirement that

$$\boldsymbol{h}_{IJ} \cdot (\nabla Q)_I = Q_J - Q_I \qquad \text{for each} \qquad J \in \Lambda_I \tag{4.12}$$

where $\boldsymbol{h}_{IJ}$ is the edge vector connecting vertex I to vertex J. This requirement may be expressed in the equivalent matrix form

$$\boldsymbol{A}_I \, (\nabla Q)_I = \boldsymbol{b}_I \tag{4.13}$$

where $\boldsymbol{A}_I$ is a $\Lambda_I \times 3$ matrix and $\boldsymbol{b}_I$ is a $\Lambda_I \times 1$ vector, with typical entries defined by $(\boldsymbol{A}_I)_{JK} = (\boldsymbol{h}_{IJ})_K$ and $(\boldsymbol{b}_I)_J = Q_J - Q_I$. As the number of unknowns is not generally equal to the number of equations, a solution to the problem may be obtained, in a least squares sense, as the vector (∇Q) that minimizes the norm defined by

$$\| \boldsymbol{A}_I \nabla Q - \boldsymbol{b}_I \|_2^2 = \sum_{J \in \Lambda_I} \left(\boldsymbol{h}_I \cdot \nabla Q - (Q_J - Q_I) \right)^2 \tag{4.14}$$

This problem is classical in optimization theory and the solution can be written as

$$(\nabla Q)_I = \left(\boldsymbol{A}_I^T \boldsymbol{A}_I \right)^{-1} \boldsymbol{A}_I^T \boldsymbol{b}_I \tag{4.15}$$

where the matrix $\left(\boldsymbol{A}_I^T \boldsymbol{A}_I \right)^{-1} \boldsymbol{A}_I^T$ is the pseudo–inverse of the matrix $\boldsymbol{A}_I$.

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

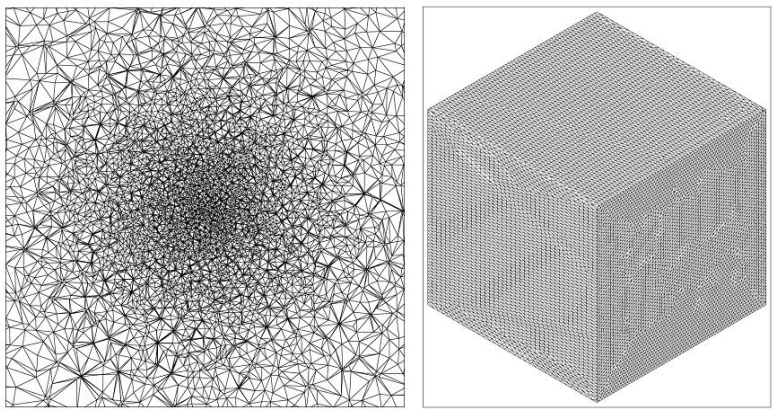


Figure 4: A cut through the isotropic tetrahedral domain mesh used to discretise a cube and a view of the corresponding surface triangulation.

An alternative procedure for the estimation of the gradient can be obtained by re–writing the requirement of equation (4.12) in the form

$$\frac{\boldsymbol{h}_{IJ}}{|\boldsymbol{h}_{IJ}|} \cdot (\nabla Q)_I = \frac{Q_J - Q_I}{|\boldsymbol{h}_{IJ}|} \qquad \text{for each} \qquad J \in \Lambda_I \tag{4.16}$$

Note now that the right hand side here can be regarded as an approximation to the derivative along the edge, whereas previously the right hand side was just an edge difference. Equations (4.12) and (4.16) are, of course, mathematically equivalent, but the corresponding least squares solutions are different, with the formulation of equation (4.16) leading to a weighted least squares approximation. In this case, the weighted norm is defined as

$$\|\boldsymbol{A}_I \nabla Q - \boldsymbol{b}_I\|_{2,w}^2 = \sum_{J \in \Lambda_I} w_{IJ}^2 \left(\boldsymbol{h}_{IJ} \cdot \nabla Q - (Q_J - Q_I)\right)^2$$

Since the square of the unweighted norm for equation (4.16) can be expressed as

$$\|\boldsymbol{A}_I \left(\nabla Q\right) - \boldsymbol{b}_I\|_2^2 = \sum_{J \in \Lambda_I} \frac{1}{|\boldsymbol{h}_{IJ}|} \left(\boldsymbol{h}_{IJ} \cdot (\nabla Q) - (Q_J - Q_I)\right)^2$$

it is apparent that the unweighted least squares solution of equation (4.16) is obtained from the weighted least squares solution of equation (4.13) with weights $w_{IJ} = 1/|\boldsymbol{h}_{IJ}|$, i.e. the weights are the inverse of the edge lengths. This LS formulation for estimation of the nodal gradients was originally proposed by Mavriplis [15].

In an attempt to improve this estimation of the solution gradient, equation (4.16) is used as the basis for an iterative procedure and the resulting method can be termed an iteratively corrected least squares method. For this process, it is convenient at this stage to write equation (4.16) in the matrix form of equation (4.13), noting that now this requires that $\boldsymbol{A}_I$ and $\boldsymbol{b}_I$ are to be re–defined as

$$(\boldsymbol{A}_I)_{JK} = \frac{(\boldsymbol{h}_{IJ})_K}{|\boldsymbol{h}_{IJ}|} \qquad (\boldsymbol{b}_I)_J = \frac{Q_J - Q_I}{|\boldsymbol{h}_{IJ}|} \tag{4.17}$$

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

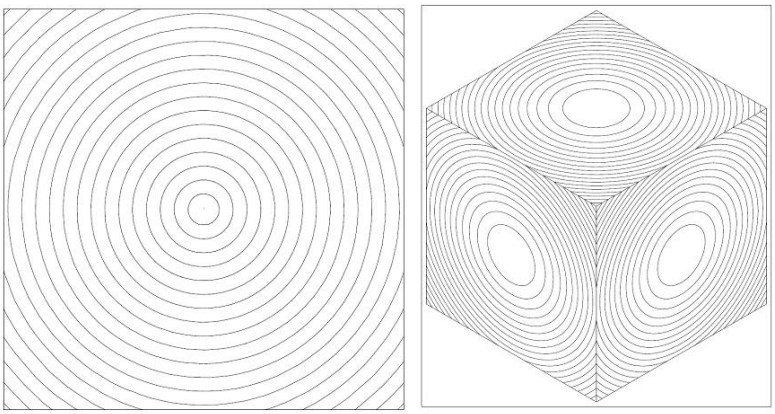


Figure 5: Contours of the norm of the gradient of the exact solution on a cut through the cube and on the surface, when the function is a general quadratic.

Using Taylor's Theorem, it follows that

$$\frac{Q_J - Q_I}{|\boldsymbol{h}_{IJ}|} = \frac{\boldsymbol{h}_{IJ}}{|\boldsymbol{h}_{IJ}|} (\nabla Q)_I + \frac{1}{2|\boldsymbol{h}_{IJ}|} \boldsymbol{h}_{IJ}^T \boldsymbol{H}(\boldsymbol{\eta}) \boldsymbol{h}_{IJ} \tag{4.18}$$

where $\boldsymbol{H}(\boldsymbol{\eta})$ denotes the Hessian evaluated at some point $\boldsymbol{\eta}$ on the edge $\boldsymbol{h}_{IJ}$. By taking $\boldsymbol{\eta}$ to be located at the midpoint of $\boldsymbol{h}_{IJ}$, the approximation

$$\langle H(\eta) \boldsymbol{h}_{IJ}, \boldsymbol{h}_{IJ} \rangle = \boldsymbol{h}_{IJ} \cdot (\nabla Q)_J - \boldsymbol{h}_{IJ} \cdot (\nabla Q)_I \tag{4.19}$$

can be adopted and, in this case, equation (4.18) reduces to

$$\frac{Q_J - Q_I}{|\boldsymbol{h}_{IJ}|} = \frac{\boldsymbol{h}_{IJ}}{|\boldsymbol{h}_{IJ}|} (\nabla Q)_I + \frac{1}{2|\boldsymbol{h}_{IJ}|} \left(\boldsymbol{h}_{IJ} \cdot (\nabla Q)_J - \boldsymbol{h}_{IJ} \cdot (\nabla Q)_I \right) \tag{4.20}$$

In this equation, $(\nabla Q)_I$ and $(\nabla Q)_J$ are unknown and the approach advocated here is to obtain the solution using an iterative process. The least squares estimate of the gradient, $(\nabla Q)_I^0$, is computed first using equation (4.15) and the improved estimate, $(\nabla Q)_I^n$ for $n = 1, 2, \ldots$, is obtained iteratively using the equation

$$\frac{\boldsymbol{h}_{IJ}}{|\boldsymbol{h}_{IJ}|} \cdot (\nabla Q)_I^n = \frac{Q_J^n - Q_I^n}{|\boldsymbol{h}_{IJ}|} - \frac{1}{2|\boldsymbol{h}_{IJ}|} \left(\boldsymbol{h}_{IJ} \cdot (\nabla Q)_J^{n-1} - \boldsymbol{h}_{IJ} \cdot (\nabla Q)_I^{n-1} \right) \tag{4.21}$$

In practice, the solution is computed using the algorithmic steps:

1. compute the least squares estimate of the gradient as

$$(\nabla Q)_I^0 = \left(\boldsymbol{A}_I^T \boldsymbol{A}_I \right)^{-1} \boldsymbol{A}_I^T \triangle Q_I \tag{4.22}$$

2. for $n = 1, 2, \ldots$, determine the improved gradient estimate $(\nabla Q)_I^n$ as

$$(\nabla Q)_I^n = \left(\boldsymbol{A}_I^T \boldsymbol{A}_I \right)^{-1} \boldsymbol{A}_I^T \left(\triangle Q_I^n - \frac{1}{2} \triangle (\nabla Q)_I^{n-1} \right) \tag{4.23}$$

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

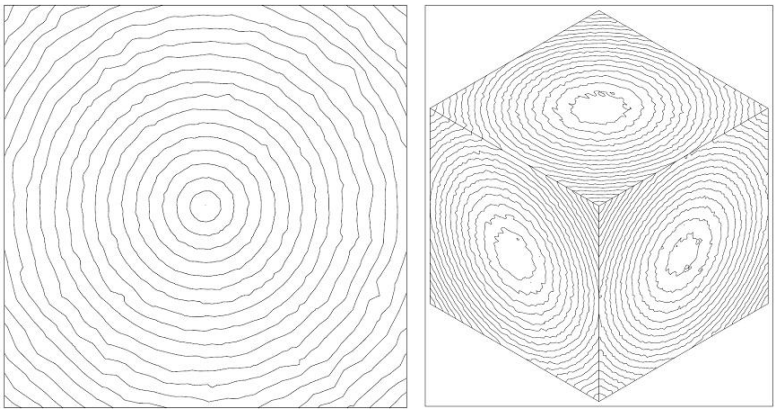


Figure 6: Contours of the norm of the gradient computed by the basic least squares method on a cut through the cube and on the surface, when the function is a general quadratic.

It should be noted that the pseudo–inverse matrix depends only on the mesh, which means that it need only be computed once and stored, which improves the computational efficiency of the algorithm. Note also that each iteration increases the size of the stencil of points that are employed, which improves the accuracy of the process at each step.

It has been shown [15] that the classical least squares method can produce accurate estimation of the solution gradient on isotropic meshes, but a significant loss of accuracy is observed in highly stretched meshes. In this case, the accuracy may be improved by using the weighted least squares approach, although the results obtained for the nodal gradients are only exact for linear functions. At convergence, the iterative method described here produces results which are exact for quadratic functions.

4.2.2. Numerical demonstration

To demonstrate the numerical performance of the iteratively corrected least squares method, consider the problems of reconstructing initially a general quadratic polynomial and then a general quartic polynomial on a domain in the shape of a cube. The domain is discretised using a mesh of isotropic tetrahedral elements and a view of a cut through the domain mesh and the corresponding surface discretisation is given in Figure 4. In each case, the norms of the gradient obtained by the basic least squares method and by the iteratively corrected least squares method are compared with the known exact distribution. Figure 5 shows the distribution of contours of the norm of the gradient of the exact solution, for the case of the quadratic function, on a cut through the domain and on the surface. The results obtained from the least squares method are shown in Figure 6 and the corresponding results, after 100 iterations, of the iteratively corrected least squares solution are shown in Figure 7. It is apparent that the gradient is accurately reconstructed by the iteratively corrected process, especially on boundaries where the gradient is much more difficult to estimate, and convergence of the logarithm of the L^2 error norm, as the number of iterations is increased, is shown in Figure 8. This example demonstrates that the iteratively corrected least squares method produces the exact gradient for quadratic functions, while the classical least squares method gives

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

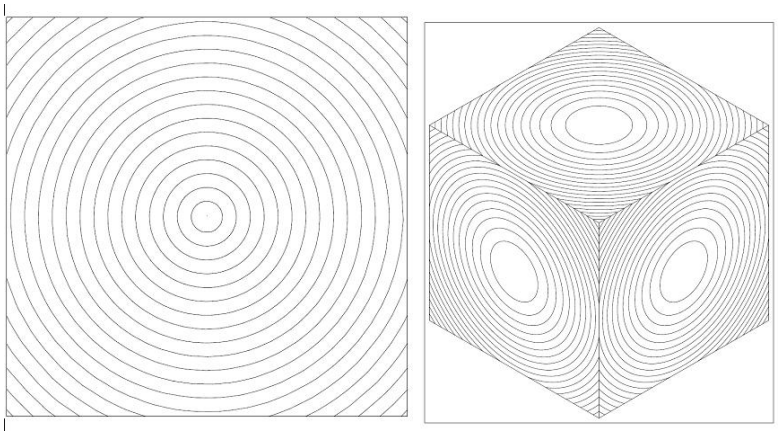


Figure 7: Contours of the norm of the gradient computed by the iteratively corrected least squares method on a cut through the cube and on the surface, when the function is a general quadratic.

results that are only exact for linear functions. Figures 9 to 12 show exactly the same results for the case of a general quartic funtion. From the variation in the logarithm of the L^2 error norm, as the number of iterations is increased, shown in Figure 12, it is apparent most of the improvement in the computed gradient values occurs during the few first iterations. This suggests that a small number of iterations can be sensibly used in practice.

4.3. Limiter design

Any slope limiter that is employed, to prevent solution over/under shooting, must possess certain properties [19, 8]. As the higher order correction of equation (4.11) changes the value of the variables on each side of a dual mesh interface, it is necessary to ensure that the integral of each variable over the volume defined by the interface and the vertices of the adjacent cells is conserved, i.e.

$$\int_{V_{IJ}} \bar{Q}_{IJ} = \int_{V_{IJ}} Q_{IJ} \tag{4.24}$$

where Q and $\bar{Q}$ refer to the values of any variable before and after application of the limiting. Here, V_{IJ} is the shaded volume indicated schematically, for a two dimensional case, in Figure 13. This requirement will be satisfied provided that

$$\bar{Q}^{\mathscr{L}}_{IJ} + \bar{Q}^{\mathscr{R}}_{IJ} = Q^{\mathscr{L}}_{IJ} + Q^{\mathscr{R}}_{IJ} \tag{4.25}$$

which implies that

$$\delta Q_{IJ} = -\delta Q_{JI} \tag{4.26}$$

To ensure satisfaction of this conservation condition, the minmod function defined by

$$\text{minmod}(a,b) = \begin{cases} 0 & \text{if } ab < 0 \\ \text{else} & \\ a & \text{if } |a| \leq |b| \\ b & \text{if } |a| \geq |b| \end{cases} \tag{4.27}$$

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

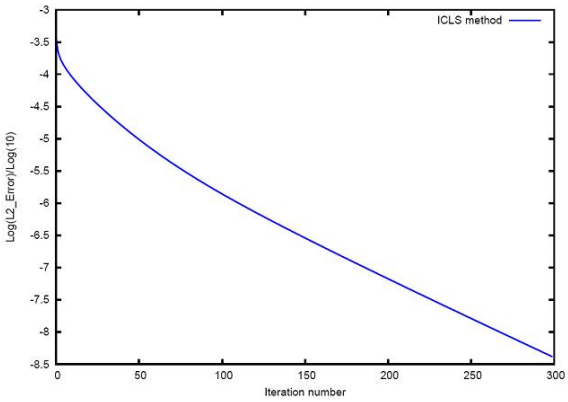


Figure 8: Convergence of the error curve for the norm of the gradient when using the iteratively corrected least squares solution for the quadratic function.

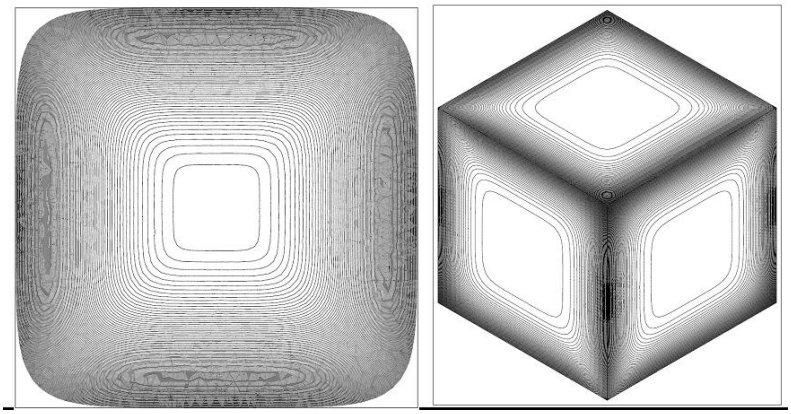


Figure 9: Contours of the norm of the gradient of the exact solution on a cut through the cube and on the surface, when the function is a general quartic.

is employed. When this function is used with arguments δQ_{IJ} and δQ_{JI}, both derivatives are set to zero, if the derivatives at nodes I and J have opposite sign, and are set to the same value if the derivatives at these nodes have the same sign. Higher order schemes are oscillatory not only in the vicinity of discontinuities but also in regions showing steep gradients in the solution. To ensure a non–oscillatory scheme, the requirement

$$\min_{J\in\Lambda_I}(Q_J - Q_J) \leq \boldsymbol{h}_{IK}\cdot\nabla Q_I \leq \max_{J\in\Lambda_I}(Q_J - Q_J) \tag{4.28}$$

is imposed. Using the local extremum diminishing (LED) concept [9] this condition may be satisifed using the Barth and Jespersen limiter [2]. This was originally devised for use with a cell centered scheme but is employed here in the cell vertex context. For each edge $\boldsymbol{h}_{IK}$, the

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

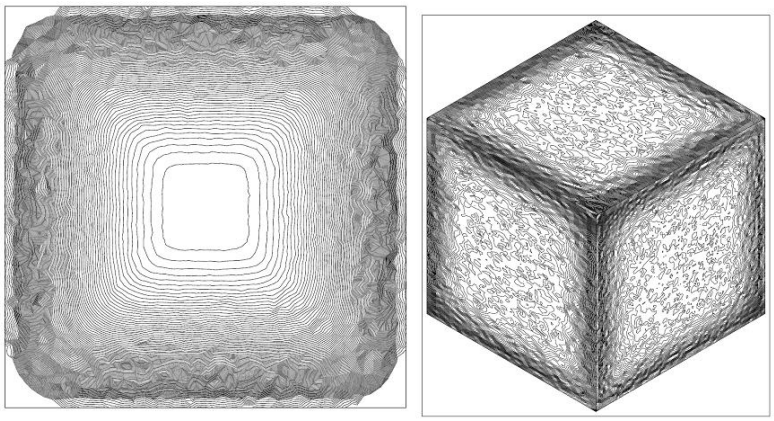


Figure 10: Contours of the norm of the gradient computed by the basic least squares method on a cut through the cube and on the surface, when the function is a general quartic.

quantity Φ_{IK} is defined as

$$\Phi_{IK} = \begin{cases} \min\left(1, \dfrac{\max_{J\in\Lambda_I}(Q_J - Q_I)}{\boldsymbol{h}_{IK}\cdot(\nabla Q)_I}\right) & \text{if } \boldsymbol{h}_{IK}\cdot\nabla Q_I > 0 \\ \min\left(1, \dfrac{\min_{J\in\Lambda_I}(Q_J - Q_I)}{\boldsymbol{h}_{IK}\cdot\nabla Q_I}\right) & \text{if } \boldsymbol{h}_{IK}\cdot\nabla Q_I < 0 \\ 1 & \text{otherwise} \end{cases} \tag{4.29}$$

and the limiter on the edges connected to node I is then obtained as

$$\Phi_I = \min_{K\in\Lambda_I} \Phi_{IK} \tag{4.30}$$

With this definition, the computed solution gradient $\boldsymbol{h}_{IK}\cdot\nabla Q_I$ is then replaced by $\Phi_I \boldsymbol{h}_{IK}\cdot\nabla Q_I$. In practice, this limiter shows an excessive smoothing of the gradient, especially for transonic flows, due to the fact that, when the condition of equation (4.28) is violated, the gradient in all directions is normalized in terms of the dominant gradient. This causes the relatively small gradients to be reduced to almost zero. This limiter is also non–differentiable and this is known to cause problems with the convergence of high order finite volume schemes [17]. In practice, therefore, the definition of equation (4.29) is replaced by

$$\Phi_{IK} = \begin{cases} sg_n\left(\dfrac{\max_{J\in\Lambda_I}(Q_J - Q_I)}{\boldsymbol{h}_{IK}\cdot(\nabla Q)_I}\right) & \text{if } \boldsymbol{h}_{IK}\cdot\nabla Q_I > 0 \\ sg_n\left(\dfrac{\min_{J\in\Lambda_I}(Q_J - Q_I)}{\boldsymbol{h}_{IK}\cdot\nabla Q_I}\right) & \text{if } \boldsymbol{h}_{IK}\cdot\nabla Q_I < 0 \\ 1 & \text{otherwise} \end{cases} \tag{4.31}$$

where $sg_n(t) = t/(1+t^n)^{\frac{1}{n}}$ denotes a family of functions derived from a sigmoid function [18]. In this form, the limiter is differentiable and provides a gradual gradient normalization. This limiter

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

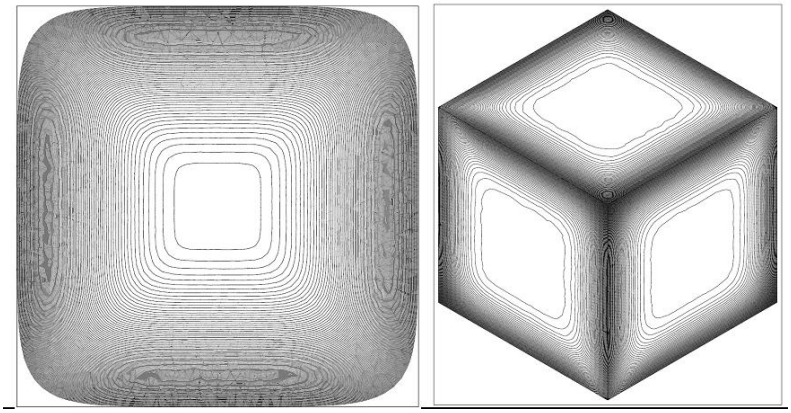


Figure 11: Contours of the norm of the gradient computed by the iteratively corrected least squares method, after 100 iterations, on a cut through the cube and on the surface, when the function is a general quartic.

is used without taking the minimum over the index K, which may reduce excessively the gradient. In addition, the sigmoid function has the attractive properties that $sg_n(t) \leq t$ and $sg_n(t) \leq 1$ for $t \geq 0$. This property guaranties that the condition of equation (4.28) will be satisfied and the non–amplification of local gradients. Furthermore, $sg_n(t)$ is a good approximation of the minimum function, since

$$\lim_{n\to\infty} (sg_n(t)) = \min(1, t) \qquad \text{for } t \geq 0 \tag{4.32}$$

The value $n = 4$ is adopted for inviscid flows, while the value $n = 2$ is chosen for viscous flows.

5. Numerical Results

The solution is advanced in time to steady state using an explicit multi-stage Runge Kutta [11] procedure and the convergence is accelerated by the use of local time stepping. To demonstrate the effectiveness of the resulting flow solver, simulations are presented of viscous transonic flow over an ONERA M6 wing and of inviscid supersonic flow over a generic F15 configuration.

5.1. Flow over an ONERA M6 wing

For the analysis of viscous transonic flow over an ONERA M6 wing, the wing surface is discretised using 110 286 triangles and 55 145 nodes. For the domain discretisation, a region consisting of 924 990 prismatic elements is generated first in the vicinity of the wing surface, by the advancing layers method, with the first layer at a distance of 1×10^{-4} from the surface. The remainder of the domain is filled with tetrahedral elements. The complete mesh contains 1 292 204 nodes and 4 880 340 tetrahedra. The imposed flow conditions are a free stream Mach number of 0.8395, an angle of attack of 3.06^0 and and a Reynolds number of 11×10^7. The solution is advanced until both the lift and the drag are converged to engineering accuracy. Figure 14 provides a detail of the

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

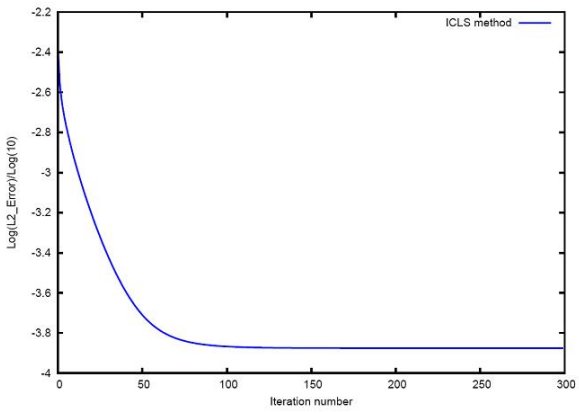


Figure 12: Convergence of the error curve for the norm of the gradient when using the iteratively corrected least squares solution for the quartic function.

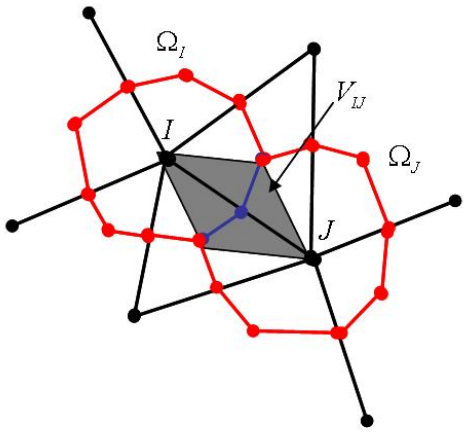


Figure 13: Illustration of the definition of the region V_{IJ} for a two dimensional triangulation.

mesh on the wing surface and on the symmetry plane and a corresponding view of the computed distribution of the pressure coefficient, C_p, contours. The convergence behaviour is demonstrated in Figure 15, which displays the variation of the logarithm of L^2 norm of the residual with the number of iterations. The chordwise profiles of the pressure coefficient are compared to experimental results at different spanwise stations in Figure 16. These results show a very good agreement with experiment, especially at the station corresponding to 80% of wing semi–span. Agreement of this quality is often difficult to obtain, since the station is located very close to the intersection point of λ shock that occurs on the upper surface of the wing. This is illustrated in Figure 17 which shows the comparison, at this station, between the distribution computed with the approach described in this paper and the centered two–parameter approach employed previously [24, 25]. The quality of the previous results can certainly be improved by tuning the free parameters, but this requirement is avoided with the upwind HLLC implementation. It should be noted, however, that the cpu

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

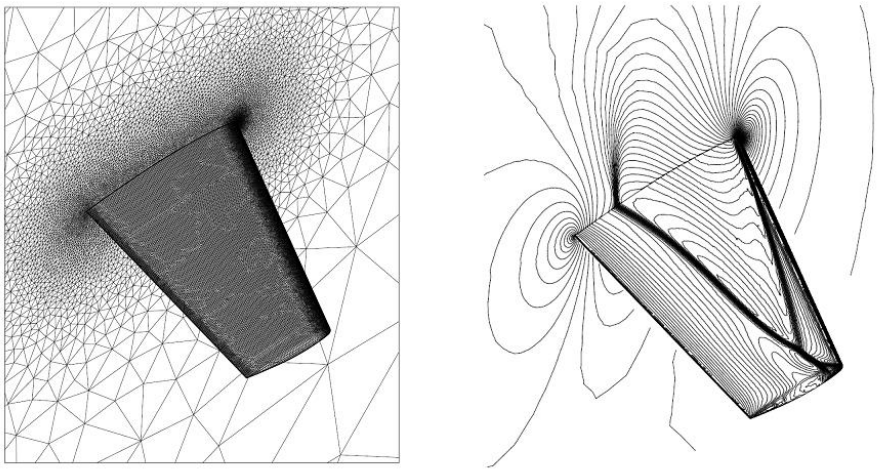


Figure 14: Steady turbulent transonic flow over an ONERA M6 wing, showing a view of the mesh, on the surface and on the symmetry plane, and a detail of the corresponding computed distributions of the C_p contours.

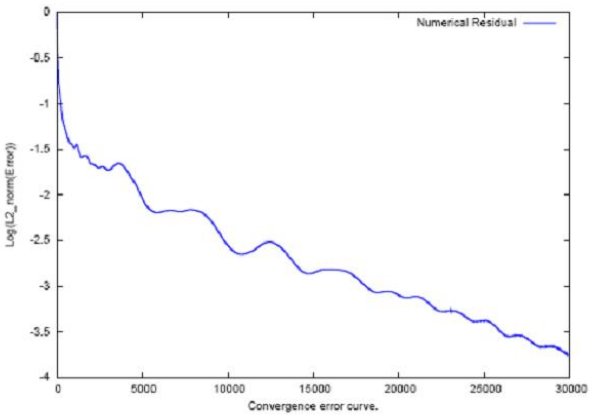


Figure 15: The convergence of the L^2 norm of the residual in the simulation of steady turbulent transonic flow over an ONERA M6 wing.

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

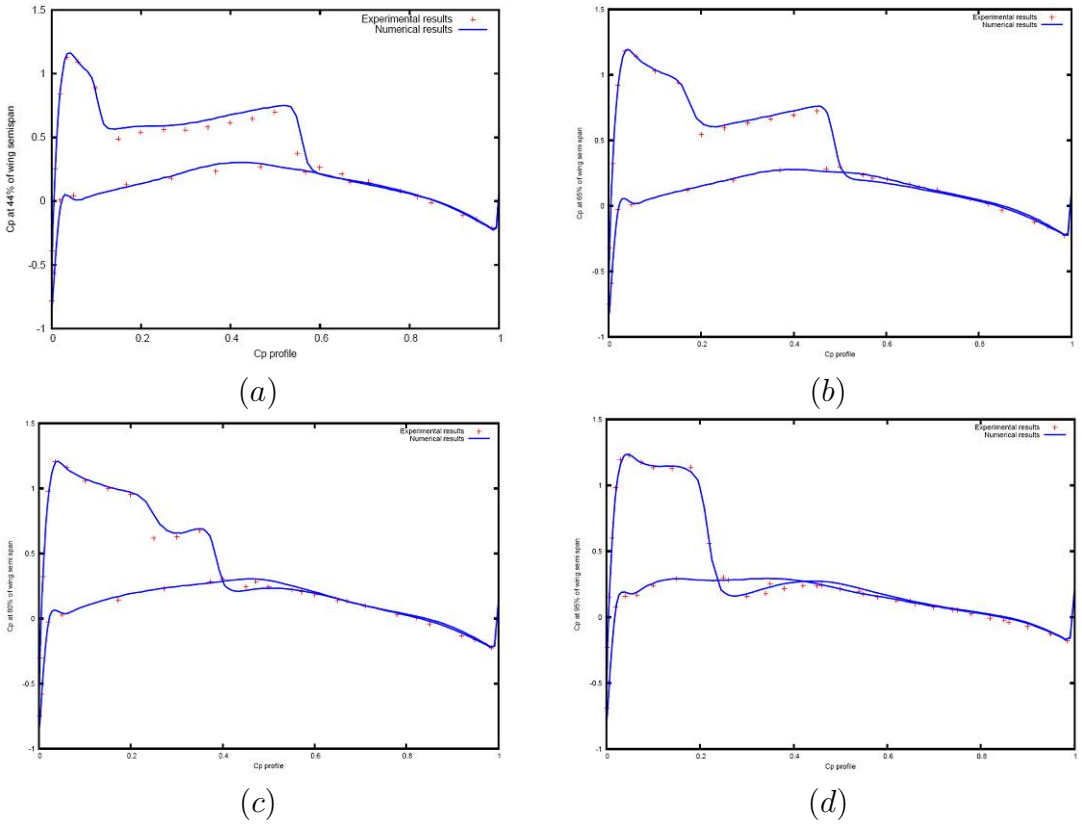


Figure 16: Steady turbulent transonic flow over an ONERA M6 wing showing a comparison between the computed C_p chordwise profile and the experimental observations at different spanwise locations.

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

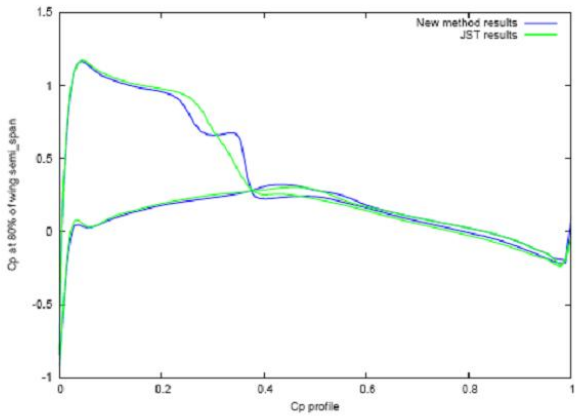


Figure 17: Steady turbulent transonic flow over an ONERA M6 wing showing a comparison between the C_p chordwise profiles produced by a centered two parameter algorithm [24, 25] and by the method described in this paper at 80% of the wing semi–span.

requirement for this test case is around 15% greater than that of the previous scheme.

5.2. Flow over an F15 configuration

To demonstrate the robustness of the proposed scheme, it is now applied,without any modification, to the simulation of an inviscid supersonic flow over a generic F15 military configuration. The aircraft surface is discretized using 423 194 triangles with 211 599 nodes and the domain is discretised with a mesh of 7 599 995 tetrahedral elements and 1 299 923 nodes. For the first simulations, the free stream Mach number is set to 2 and the angle of attack is 3^0. Figure 18 shows a cut through the domain mesh and a view of the discretised surface. The corresponding computed pressure coefficient contour distributions are also shown in this figure and it is apparent that the shocks are cleanly captured. The convergence behaviour is demonstrated in Figure 19, which displays the variation of the logarithm of L^2 norm of the residual with the number of iterations. To further demonstrate the robustness of the scheme, the simulation is repeated with the free stream Mach number increased to 3, but at the same angle of attack. The corresponding results are shown in Figure 20 and in Figure 21 and demonstrate the robustness of the method with respect to a range of flow speeds and geometric complexity.

6. Conclusions

The paper has described the implementation of an HLLC finite volume method for the solution of the compressible Navier Stokes equations on unstructured meshes. The main contributions include the improvement in the calculation of the acoustic wave speed within the HLLC Riemann solver, a

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

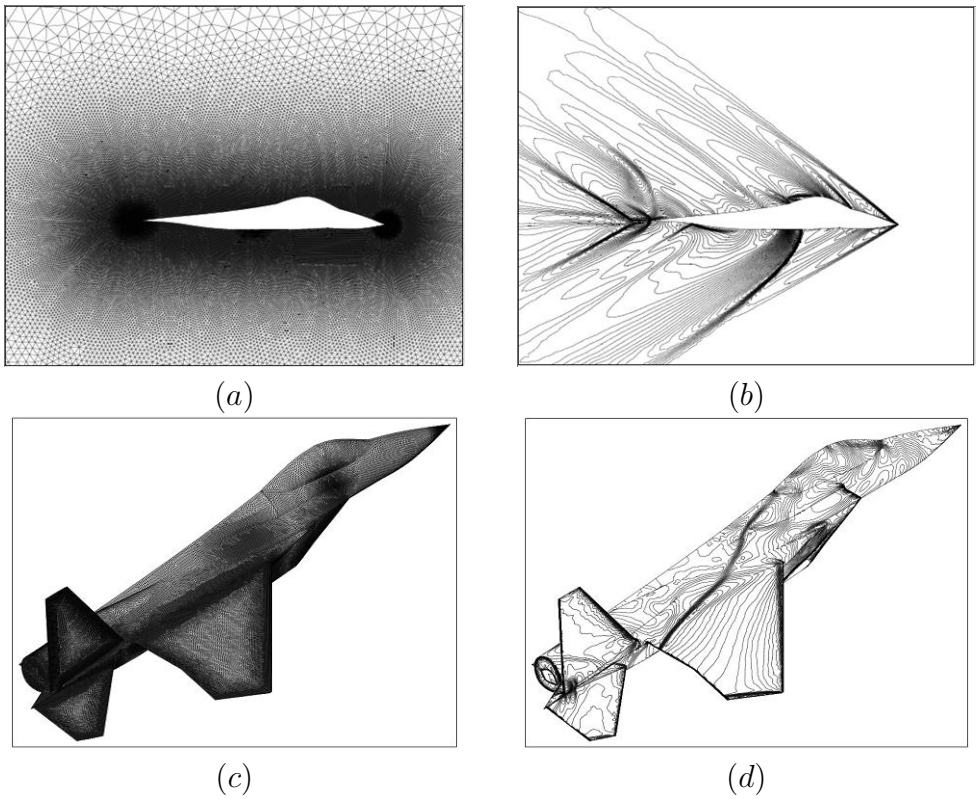


Figure 18: Steady inviscid flow over a generic F15 configuration when $M_\infty = 2$: (a) detail of a cut through the domain mesh; (b) computed distribution of C_p contours on this cut; (c) view of the discretised surface; (d) view of the computed distribution of C_p contours on the surface.

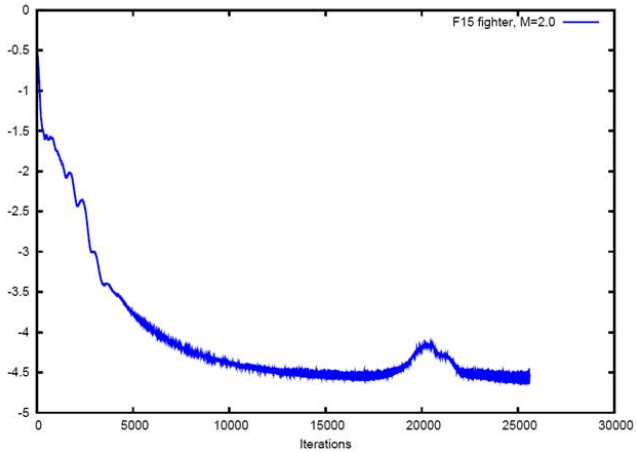
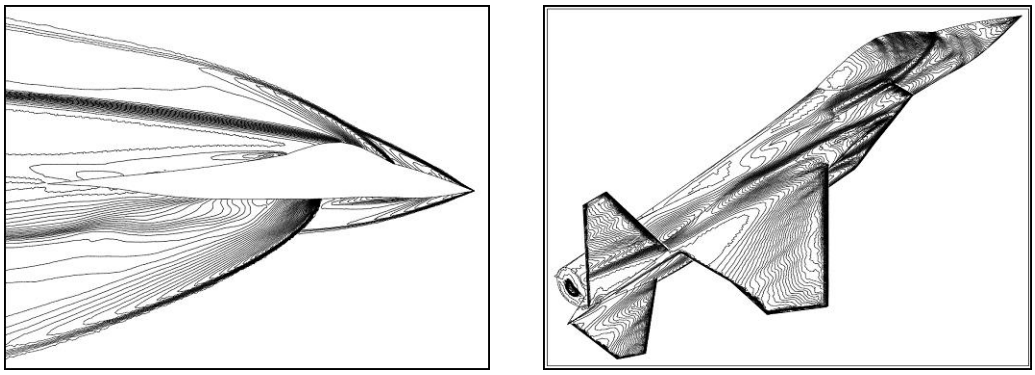


Figure 19: Convergence of the L^2 norm of the residual in the simulation of steady inviscid flow over a generic F15 configuration when $M_\infty = 2$.

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press



(a) (b)

Figure 20: Steady inviscid flow over a generic F15 configuration when $M_\infty = 3$: (a) computed distribution of C_p contours on a cut through the domain mesh; (b) view of the computed distribution of C_p contours on the surface.

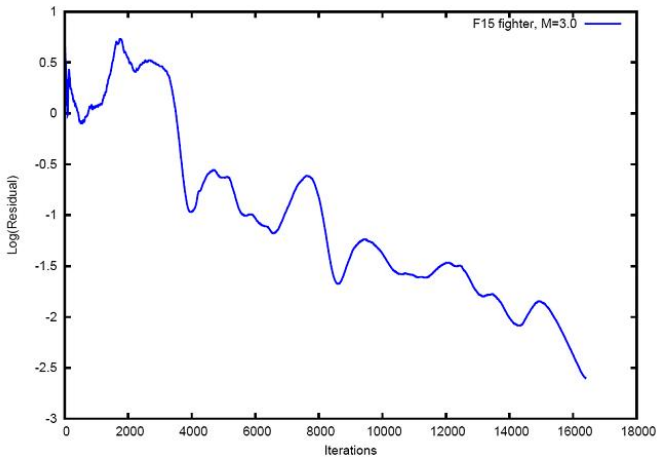


Figure 21: Convergence of the L^2 norm of the residual in the simulation of steady inviscid flow over a generic F15 configuration when $M_\infty = 3$.

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

new limiter design based on a stability analysis and the use of the iteratively corrected least squares method to improve the approximation of the gradient reconstruction. The results of the simulation of a turbulent transonic flow over an ONERA M6 wing and of a supersonic inviscid flow over a military aircraft configuration are included. The results obtained demonstrate the accuracy and robustness of the proposed approach, in terms of convergence and stability, for Mach numbers ranging from 0.8395 to 3, without the requirement of tuning any free parameters.

References

[1] T. Barth. *A 3–D upwind Euler solver for unstructured meshes*. AIAA–91–1548–CP, 1991.

[2] T. Barth, D. Jespersen. *The design and application of upwind schemes on unstructured meshes*. AIAA Paper 89–0366, 1989.

[3] P–H. Cournède, C. Debiez, A. Dervieux. *A positive MUSCL scheme for triangulations*. INRIA Report 3465, 1998.

[4] P. Geuzaine. *An implicit upwind finire volume method for compressible turbulent flows on unstructured meshes*. PhD Thesis, Université de Liège, 1999.

[5] A. Harten, P.D. Lax, B. Van Leer. *On upstream differencing and Godunov-type schemes for hyperbolic conservation laws*. SIAM Review, 25 (1983), 35–61.

[6] R. Hartmann, J. Held, T. Leicht, F. Prill. *Discontinuous Galerkin methods for computational aerodynamics–3D adaptive flow simulation with the DLR PADGE code*. Aerosp. Sci. Tech., in press (2010), DOI: 10.1016/j.ast.2010.04.002.

[7] O. Hassan, K. Morgan, E. J. Probert, J. Peraire. *Unstructured tetrahedral mesh generation for three–dimensional viscous flows*. Int. J. Num. Meth. Engg., 39 (1996), 549–567.

[8] C. Hirsch. Numerical Computation of Internal and External Flows. Volume 2 John Wiley and Sons, Chichester, 1990.

[9] A. Jameson. *Analysis and design of numerical schemes for gas dynamics. 1: artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence*. Int. J. CFD, 4 (1995), 171–218.

[10] A. Jameson, T. J. Baker, N. P. Weatherill. *Calculation of Inviscid transonic flow over a complete aircraft*. AIAA Paper–86–0103, 1986.

[11] A. Jameson, W. Schmidt, E. Turkel. *Numerial solution of the Euler equations by finite volume methods using Runge–Kutta time stepping schemes*. AIAA Paper 81–1259, 1981.

[12] R. Löhner. Applied CFD Techniques. John Wiley and Sons, Chichester, 2001.

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

[13] H. Luo, J. D. Baum, R. Löhner. *Edge–based finite element scheme for the Euler equations*. AIAA J. 32 (1994), 1183–1190.

[14] P. R. M. Lyra. *Unstructured grid adaptive algorithms for fluid dynamics and heat conduction*. PhD Thesis, University of Wales, Swansea, 1994.

[15] D. J. Mavriplis. *Revisiting the least–squares procedure for gradient reconstruction on unstructured meshes*. AIAA Paper 2003–3986, 2003.

[16] D. J. Mavriplis, V. Venkatakrishnan. *A 3D agglomeration multigrid solver for the Reynolds–averaged Navier–Stokes equations on unstructured meshes*. Int. J. Num. Meth. Fluids, 23 (1996), 527–544.

[17] C. Michalak, C. Ollivier–Gooch. *Accuracy preserving limiter for the high–order accurate solution of the Euler equations*. J. Comput. Phys., 228 (2009), 8693–8711.

[18] T. M. Mitchell. Machine Learning. WCB–McGraw–Hill, 1997.

[19] K. Morgan, J. Peraire. *Unstructured grid finite element methods for fluid mechanics*. Rep. Prog. Phys., 61 (1998), 569–638.

[20] K. Morgan, J. Peraire, J. Peiró, O. Hassan. *The computation of three dimensional flows using unstructured grids*. Comp. Meth. Appl. Mech. Engg, 87 (1991), 335–352.

[21] J. Peiró, J. Peraire, K. Morgan. *The generation of triangular meshes on surfaces*. in C. Creasy and C. Craggs (eds), Applied Surface Modelling, Ellis–Horwood, Chichester, 25–33, 1989.

[22] J. Peraire, J. Peiró, K. Morgan. *Finite element multigrid solution of Euler flows past installed aero-engines*. Comp. Mech., 11 (1993), 433–451.

[23] S. Pirzadeh. *Viscous unstructured three–dimensional grids by the advancing–layers method*. AIAA–94–0417, 1994.

[24] K. A. Sørensen. *A multigrid accelerated procedure for the solution of compressible fluid flows on unstructured hybrid meshes*. PhD Thesis, University of Wales, Swansea, 2002.

[25] K. A. Sørensen, O. Hassan, K. Morgan, N. P. Weatherill. *A multigrid accelerated hybrid unstructured mesh method for 3D compressible turbulent flow*. Comp. Mech., 31 (2003), 101–114.

[26] P. R. Spalart, S. R. Allmaras. *A one–equation turbulent model for aerodynamic flows*. AIAA Paper 92–0439, 1992.

[27] T. E. Tezduyar. *Finite element methods for flow problems with moving boundaries and interfaces*. Arch. Comp. Meth. Engg., 8 (2001) 83–130.

[28] E. F. Toro. Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction (2nd edn), Springer, Berlin, 1999.

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press

[29] E. F Toro, M. Spruce, W. Speares. *Restoration of the contact surface in the HLL–Riemann Solver*. Shock Waves, 4 (1994), 25–34.

[30] M. Vahdati, K. Morgan, J. Peraire. *The computation of viscous compressible flows using an upwind algorithm and unstructured meshes*. in S. N. Atluri (ed), Computational Nonlinear Mechanics in Aerospace Engineering, AIAA Progress in Aeronautics and Astronautics Series, AIAA, Washington, 479–505, 1992.

[31] N. P. Weatherill, O. Hassan. *Efficient three–dimensional Delaunay triangulation with automatic boundary point creation and imposed boundary constraints*. Int. J. Num. Meth. Engg., 37 (1994), 2003–2039.

[32] F. M. White. Viscous Fluid Flow (3rd edn). McGraw Hill, Boston, 2006.

https://doi.org/10.1051/mmnp/20116308 Published online by Cambridge University Press