INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2007; **72**:1332–1354
Published online 16 April 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.2048

A cut-cell non-conforming Cartesian mesh method for compressible and incompressible flow

J. Pattinson, A. G. Malan*,† and J. P. Meyer

Department of Mechanical and Aeronautical Engineering, University of Pretoria, Pretoria 0002, South Africa

SUMMARY

This paper details a multigrid-accelerated cut-cell non-conforming Cartesian mesh methodology for the modelling of inviscid compressible and incompressible flow. This is done *via* a single equation set that describes sub-, trans-, and supersonic flows. Cut-cell technology is developed to furnish body-fitted meshes with an overlapping mesh as starting point, and in a manner which is insensitive to surface definition inconsistencies. Spatial discretization is effected *via* an edge-based vertex-centred finite volume method. An alternative dual-mesh construction strategy, similar to the cell-centred method, is developed. Incompressibility is dealt with *via* an artificial compressibility algorithm, and stabilization achieved with artificial dissipation. In compressible flow, shocks are captured *via* pressure switch-activated upwinding. The solution process is accelerated with full approximation storage (FAS) multigrid where coarse meshes are generated automatically *via* a volume agglomeration methodology. This is the first time that the proposed discretization and solution methods are employed to solve a single compressible–incompressible equation set on cut-cell Cartesian meshes. The developed technology is validated by numerical experiments. The standard discretization and alternative methods were found equivalent in accuracy and computational cost. The multigrid implementation achieved decreases in CPU time of up to one order of magnitude. Copyright © 2007 John Wiley & Sons, Ltd.

Received 4 September 2006; Revised 30 December 2006; Accepted 19 February 2007

KEY WORDS: cut-cell non-conforming Cartesian meshes; inviscid compressible and incompressible flow; artificial compressibility; alternative dual-mesh construction strategy; agglomerated FAS multigrid

*Correspondence to: A. G. Malan, Department of Mechanical and Aeronautical Engineering, University of Pretoria, Pretoria 0002, South Africa.
†E-mail: a.g.malan@up.ac.za

Contract/grant sponsor: Denel Aerospace Systems
Contract/grant sponsor: South African National Research Foundation; contract/grant number: 2053287
Contract/grant sponsor: THRIP; contract/grant number: 3257

1. INTRODUCTION

Computational fluid dynamics (CFD) is today widely accepted as it plays a key role in the design of numerous fluid flow-intensive industrial systems. These range from hydrodynamics and low-speed aerodynamics to supersonic flows over missiles. Although CFD has advanced significantly over the last few decades, its effective use as a design tool is today still hampered by concerns such as long lead times and excessive operator and computational time cost [1, 2]. One of the main problem areas is the mesh generation task, which can be more time intensive than the actual flow modelling.

Although a variety of non-overlapping or body-fitted type mesh generation techniques is available as documented by Blazek [3], current methods are not truly automatic [4] and often require many hours of user-assisted grid generation [5], particularly where complex geometries are involved. This typically not only requires a high level of expertize from the operator but also much time is spent on tasks such as fixing of 'dirty' geometries (to rectify minuscule gaps, overlaps, or discontinuities between surfaces), overall domain decomposition into suitably meshable volumes, as well as the selection of mesh generation technique-related input parameters. The latter is of importance in order to furnish elements with suitable quality from a numerical discretization point of view.

When anisotropic meshes are not required, as is the case with inviscid flow, overlapping non-conforming Cartesian meshes almost completely circumvent all of the above mesh generation-related problems. Cartesian-type mesh generation is further known to be highly amenable to automation, as reported by several authors [5–9]. These meshes also naturally produce elements which are superior to unstructured types (such as triangles or tetrahedra) in terms of accuracy *versus* computational cost, and in all domain areas except at the boundary and where hanging nodes are present [10]. Aftosmis *et al.* [7] demonstrate that these types of meshes may in addition be generated with superb efficiency (the complexity of the method is $O(N \log N)$ where N is the number of cells) on complex geometries. The main drawback with using the overlapping Cartesian approach is the stair-step that it employs to approximate the fluid domain boundary. As Kirkpatrick *et al.* [11] indicate, a prohibitively fine mesh is required for accurate boundary field calculations. Fortunately, various techniques have been developed to circumvent the former. The method employed by most researchers is the so-called cut-cell approach [6, 12–15]. In recent work, Dawes [16] employs level sets as an efficient means of representing the geometry, which may then be used for the purpose of producing a cut-cell mesh. In this work, a mesh cutting procedure is developed, which is insensitive to surface definition inconsistencies ('dirty' geometries). It employs widely used geometry definition formats.

Having decomposed the fluid domain into non-overlapping sub-domains or elements, the governing equations are discretized over each. The methodology employed almost exclusively to date in the context of cut-cell Cartesian meshes is the cell-centred finite volume method [6, 8, 9, 13, 14, 17–22]. A reference to the edge-based median dual vertex-centred finite volume method does not currently feature. This method naturally offers greater accuracy in unstructured regions [3] and allows certain boundary conditions (such as density on symmetry boundaries or components of characteristic boundary conditions) to be enforced more naturally as compared to the aforementioned method (no extrapolation or ghost cell generation is required).

The two main approaches currently used to deal with fluid incompressibility are the so-called *pressure-based* (projection scheme) method proposed by Patankar [23], and the *density-based* (artificial compressibility) scheme introduced by Chorin [24]. The latter method has received attention only in recent years [25–28], in part due to its memory efficiency, suitability to massively parallel environments, and mutual technology transfer with compressible time-marching systems. In the context of modelling incompressible flow on cut-cell non-conforming Cartesian meshes, the projection-type schemes have featured exclusively to date [17, 20, 29].

The vertex-centred edged based finite volume method constitutes a notionally second-order accurate (central difference) type discretization method, which is known to exhibit a tendency for odd–even decoupling. To suppress this tendency, stabilization has to be added. Stabilizing such schemes without the loss of second-order accuracy is a major concern in the field of computational fluid dynamics [30]. This has led to the development of various methods by researchers such as Beam and Warming [31], Jameson *et al.* [32], and MacCormack and Baldwin [33]. One such class of scheme is termed *artificial viscosity* or *artificial dissipation*, and involves the addition of a biharmonic operator to the system of equations in the regions devoid of discontinuities. A harmonic operator (first-order accurate upwinding) is to be added in the vicinity of shocks in order to render a stable non-physical oscillation-free method [30]. Another class of method employed to suppress non-physical oscillations is the so-called *upwind* schemes (Godunov type/Riemann solver). In the context of Euler flow on unstructured Cartesian meshes, these schemes are the most popular [5, 10, 12–15, 19, 21, 34] and the *artificial dissipation* methods do not at present feature. Due to the latter, as well as to in addition evaluate its applicability to cut-cell Cartesian meshes, the Jameson, Schmidt and Turkey [32] (JST) variant of *artificial dissipation* will be employed in this work.

To complement the speed and efficiency of the Cartesian meshing methods, it is of importance that the solution process be similarly fast and efficient. Both implicit and explicit solution methods are currently employed with Cartesian meshes for this purpose. The former includes lower–upper symmetric Gauss–Seidel [8] used in conjunction with a variant of the full approximation storage (FAS) multigrid method, as well as the block lower–upper symmetric Gauss–Seidel method [10, 34].[‡] The aforementioned method requires less memory than a fully implicit scheme, while involving a comparable amount of iterations. Explicit temporal discretization techniques vary from multi-stage time-stepping [13] to employing Runge–Kutta relaxation in conjunction with FAS multigrid solution acceleration [14, 35].[§] The memory cost of these techniques is optimal ($O(N)$), while computational cost tends towards $O(N)$.

In this paper sub-, trans-, and supersonic inviscid compressible, as well as inviscid incompressible flows will be modelled on cut-cell non-conforming Cartesian meshes. This will be done using a single unified governing equation set. In the context of cut-cell meshes, spatial discretization will for the first time be effected *via* an edge-based vertex-centered finite volume method. The standard dual-mesh construction strategy will be employed and an alternative methodology in addition developed. The latter is equivalent to a cell-centred method, and in the interest of applicability, the standard edge-based averaging procedure is suitably extended. A recently developed generalized locally preconditioned artificial compressibility technique [27] will be implemented to deal with fluid incompressibility such that both compressible and incompressible flow may be solved with a single numerical scheme. Scalar-valued (JST) artificial dissipation is employed for the purpose of stabilization, while pressure switch-activated first-order upwinding will be used for shock

[‡] Although the cited work involves unstructured meshes in addition to non-conforming Cartesian meshes, it is deemed of direct relevance to the work under consideration.

[§] In the cited work an implicit method is employed to advance real time and an explicit method is used to converge the inner pseudo-time loop. This is known as dual-time-stepping.

capturing. The solution is accelerated *via* an agglomerated FAS multigrid method. Multigrid mesh agglomeration is effected automatically through a generically applicable volume agglomeration strategy. The developed methodology will be evaluated by application to the solution of a number of inviscid flow benchmark test cases. The accuracy of the standard as well as alternative discretization strategies will be evaluated by comparison of predicted results to the analytical or benchmark solutions. The computational cost as well as multigrid speed-up obtained with the two aforementioned discretization methods will be considered to assess their computational efficiencies.

2. GOVERNING EQUATIONS

The partial differential equation set, which describes the inviscid flow of an incompressible or compressible fluid, consists of equations which enforce the principles of mass, momentum, and energy conservation. The governing system of equations may be written for a Cartesian co-ordinate system in the following non-dimensional conservative form

$$\frac{\partial \mathbf{W}}{\partial \mathbf{Q}}\frac{\partial \mathbf{Q}}{\partial t}+\frac{\partial \mathbf{F}^j}{\partial x_j}=\mathbf{0} \tag{1}$$

where

$$\mathbf{W}=\begin{pmatrix}\rho\\ \rho u_1\\ \rho u_2\\ \rho E\end{pmatrix},\quad \mathbf{Q}=\begin{pmatrix}p\\ u_1\\ u_2\\ \eta E\end{pmatrix},\quad \mathbf{F}^j=\begin{pmatrix}\rho u_j\\ \rho u_1 u_j+p\delta_{1j}\\ \rho u_2 u_j+p\delta_{2j}\\ \eta(\rho E+p)u_j\end{pmatrix} \tag{2}$$

and t is time, ρ is the density, u_j the velocity component in direction x_j, E is the specific *total energy* and p is the pressure. Further, δ_{ij} refers to the Kronecker delta function and η switches compressible flow-specific terms on and off. The latter is defined as

$$\eta=\begin{cases}1 & \text{for compressible flow}\\ 0 & \text{for incompressible flow}\end{cases} \tag{3}$$

Note that in the case of incompressible flow, the above definition results in the isothermal form of the equations in which the energy equation plays no role. Further,

$$\frac{\partial \mathbf{W}}{\partial \mathbf{Q}}=\begin{pmatrix}\dfrac{1}{c^2} & 0 & 0 & 0\\ \dfrac{a_u u_1}{c^2} & \rho & 0 & 0\\ \dfrac{a_u u_2}{c^2} & 0 & \rho & 0\\ \dfrac{E}{c^2} & \rho & \rho & \rho\end{pmatrix} \tag{4}$$

where a_u is 1 for compressible flow. For incompressible flow, a_u is calculated as proposed by Malan *et al.* [27]. The acoustic velocity c is calculated as follows:

$$c = \eta\sqrt{\gamma RT} + (1-\eta)c_\tau \tag{5}$$

where γ is the ratio of specific heats and R is the ideal gas constant. Also, c_τ is the pseudo-acoustic velocity as defined by Malan *et al.* [27]. In the above equations, non-dimensional quantities are related to their dimensional counterparts (depicted with superscript *) through the following relations

$$\begin{aligned} &t = \frac{t^* U^*_\infty}{L^*}, \quad u_j = \frac{u^*_j}{U^*_\infty}, \quad p = \frac{p^*}{\rho^*_\infty U^{*2}_\infty}, \quad T = \frac{T^*}{U^{*2}_\infty / C_p} \\ &\rho = \frac{\rho^*}{\rho^*_\infty}, \quad x_j = \frac{x^*_j}{L^*}, \quad E = \frac{E^*}{U^{*2}_\infty} \end{aligned} \tag{6}$$

where t, L, and T are time, characteristic length, and temperature, respectively. Further, the subscript ∞ denotes free-stream conditions with U_∞ being the free-stream velocity.

2.1. Boundary conditions

At the in- and outflow boundaries, characteristic-type conditions are employed. For the purpose of this work a generalization was required to ensure applicability to both compressible and incompressible flow systems. In all cases, the application of boundary conditions on the far field of a domain depends on the local Mach number (or pseudo-Mach number in the case of incompressible flow), as well as whether the flow is leaving or entering the domain. The four cases encountered in this work are as follows: supersonic inlet, supersonic outlet, incompressible as well as compressible subsonic in- and outlet conditions. It is implied that the latter is applied to transonic flow systems.

At the supersonic-inlet, free stream conditions are prescribed, while stream-up extrapolated values are applied to the outflow (an inverse distance weighting algorithm was employed for this purpose). The flow conditions at boundaries in the subsonic case are calculated *via* an altered version of a characteristic-type analysis based on one-dimensional Riemann invariants [18]. The alteration is required to extend the method's applicability to incompressible flow systems. The procedure is outlined below.

Riemann invariants, R, based on the free-stream and extrapolated values are calculated as

$$R_\infty = \mathbf{u}_\infty \cdot \mathbf{n} - \frac{2c_\infty}{\gamma - 1} \tag{7}$$

$$R_e = \mathbf{u}_e \cdot \mathbf{n} + \frac{2c_e}{\gamma - 1} \tag{8}$$

The subscript e denotes values determined from the extrapolated primitive variables and $\mathbf{n}$ denotes the boundary outward-pointing normal unit vector. The required sonic velocities for both compressible and incompressible flows are calculated as follows:

$$c_\infty = \eta\sqrt{(\gamma-1)T_\infty} + (1-\eta) \times \max(\sqrt{r\mathbf{u}_\infty \cdot \mathbf{u}_\infty}, c_\tau) \tag{9}$$

$$c_e = \eta\sqrt{(\gamma-1)T_e} + (1-\eta)c_\infty \tag{10}$$

Note that the above sonic velocities in the incompressible case were calculated as shown in the interest of consistency with the artificial compressibility solution method. Further, in this work the artificial compressibility-related parameter is set to $r = 1.2$. Finally, γ in Equations (7) and (8) is calculated as

$$\gamma = \eta\gamma_{\text{air}} + (\eta - 1)\left(\frac{c_\infty^2}{T_\infty} + 1\right) \tag{11}$$

where, in the case of incompressible flow, $T_\infty = 1.0$.

From the above, the sonic velocity at the boundary (c_b) is now calculated as follows

$$c_\text{b} = 0.25\eta(\gamma - 1)(R_e - R_\infty) + (1 - \eta)c_\infty \tag{12}$$

and the velocity normal to the boundary is found by

$$u_\text{n} = 0.5(R_e + R_\infty) \tag{13}$$

Using the normal velocity at the boundary, the velocity at the in- and outflow boundaries are calculated as

$$\mathbf{u}_\text{b} = \mathbf{u}_\infty + (u_\text{n} - \mathbf{u}_\infty \cdot \mathbf{n}) \cdot \mathbf{n} \quad \text{and} \quad \mathbf{u}_\text{b} = \mathbf{u}_e + (u_\text{n} - \mathbf{u}_e \cdot \mathbf{n}) \cdot \mathbf{n} \tag{14}$$

respectively. The temperature at the in- and outflow boundaries are found from

$$T_\text{b} = T_\infty\left(\frac{c_\text{b}^2}{c_\infty^2}\right) \quad \text{and} \quad T_\text{b} = T_e\left(\frac{c_\text{b}^2}{c_e^2}\right) \tag{15}$$

respectively.

The pressure at the boundary is determined by applying the ideal gas isentropy relations at the inflow and outflow, respectively, as

$$p_\text{b} = p_\infty\left(\frac{T_\text{b}}{T_\infty}\right)^{\gamma/(\gamma-1)} \quad \text{and} \quad p_\text{b} = p_e\left(\frac{T_\text{b}}{T_e}\right)^{\gamma/(\gamma-1)} \tag{16}$$

The above relations hold for both compressible and incompressible flow systems. Finally, symmetry or slip boundary conditions are employed at the surface of the aerodynamic object.

3. SPATIAL-DOMAIN DECOMPOSITION

Spatial-domain decomposition in this paper is effected *via* a cut-cell non-conforming Cartesian mesh method, which circumvents the majority of the shortcomings of body-fitted type techniques due to the following advantages:

1. It is naturally able to deal with ‘dirty’ geometries.
2. Mesh generation is exceptionally fast, with the complexity of the method being $O(N \log N)$.
3. Element quality is near optimal for the vast majority of the domain, and is a natural consequence of the mesh generation procedure.
4. The vast majority of elements found in the domain require less computational effort than the unstructured (triangles in two dimensions) counterpart.
5. The mesh is well suited to adaptive refinement.

In the following sections, the procedure followed to construct the body-fitted cut-cell non-conforming Cartesian meshes is detailed.

3.1. Cut-cell non-conforming Cartesian mesh procedure

The first stage in furnishing a suitable cut-cell mesh is the generation of an overlapping non-conforming Cartesian mesh. The commercial package Harpoon [36] was employed for this purpose. Examples of meshes generated with the latter package is shown in Figure 1 (top). Note that the mesh generator attempts to remove cells that are completely inside the geometry, leaving just the cells that overlap the boundary. In this, it has been found that these packages are not always entirely successful, often not removing cells that lie completely within the geometry and in some cases removing cells that overlap the boundary. This is depicted in the figure (bottom).

In order to circumvent the loss in accuracy resulting from a stair-step boundary definition, a preprocessor was developed which furnishes a body-fitted cut-cell mesh with the above generated overlapping mesh as starting point. This was further to be done in a manner which is insensitive to surface definition inconsistencies ('dirty' geometries). To effect the aforementioned, the preprocessor needs to have access to and be able to effectively interpret the geometry definition. In this work, the OpenCASCADE [37] open source software was employed. The latter was found to

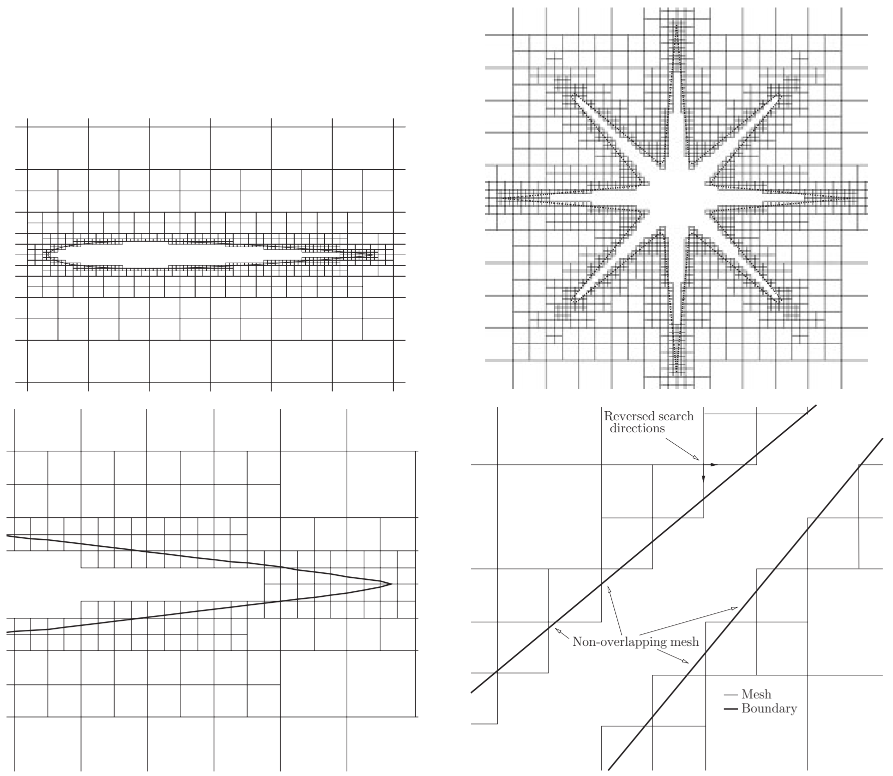

Figure 1. The non-conforming overlapping Cartesian mesh test cases (with body geometry definition) employed for testing of the developed preprocessor (cell-cutting and merging technology). Close-ups are shown below.

be industrially relevant (as a multitude of features are available) while it allows the interpretation of three widely used geometry definition formats, viz. STL[¶], STEP, and IGES.

3.2. Cell-cutting algorithm

The cell-cutting algorithm commences by visiting all boundary nodes and checking the attached edges for an intersection with the geometry. If no intersection is found, the checked edges are deleted (as these are located inside the geometry) and the boundary redefined. If an intersection is found, the edge is marked for cutting. The procedure is then repeated until all edges completely interior to the geometry have been deleted and all edges that intersect the boundary found (which is the case when at least one edge attached to each boundary node intersects the boundary). All boundary nodes are then deleted and the new boundary defined by creating nodes at edge-geometry intersections.

It is important to note that the above algorithm only works in isolation on meshes that completely overlap the boundary. However, the mesh generation process may produce areas where this is not the case i.e. containing 'voids'. An example of this is depicted in Figure 1 (bottom right). To address this, the preprocessor was extended to locate these voids and fill them with triangles or quadrilaterals.

3.3. Cell-merging operation

Pure cutting of the edges and filling of voids, as described above, may result in minuscule elements being created at the boundary. These elements are often many times smaller than their adjacent counterparts, which can seriously impair the accuracy of the solution scheme as well as solution time. To remedy this, such elements are merged into their neighbours.

Cell merging commences by performing a loop over all boundary elements. If an edge attached to a particular element is smaller than the largest edge of that specific element by more than a certain factor (in this work a factor of 2 was found to yield acceptable results for the cases tested), the edge is collapsed. Of the two nodes attached to the edge to be collapsed, the node that does not lie on the boundary is omitted. If both nodes lie on the boundary, then either is selected for deletion. Finally, a second loop is performed over the boundary elements and those that contain less than three nodes removed. An example of the resulting cell-merged mesh is shown in Figure 2 (right).

4. SOLUTION PROCEDURE: SPATIAL DISCRETIZATION

As mentioned previously, it is proposed that the edge-based vertex-centred dual-mesh finite volume spatial discretization method be applied to cut-cell non-conforming Cartesian meshes. This method is expected to be well suited for this purpose as it is naturally applicable to both structured- and unstructured-type meshes. The first step in discretizing the governing equations under consideration is the construction of the dual-mesh from the generated cut-cell mesh.

[¶]The .stl or stereo-lithography format is an ASCII or binary documenting system used in manufacturing digitally describe geometries. It employs a list of triangular surfaces to describe a computer-generated solid model surface. Note that this is the file-output format generated by most rapid prototyping machines.

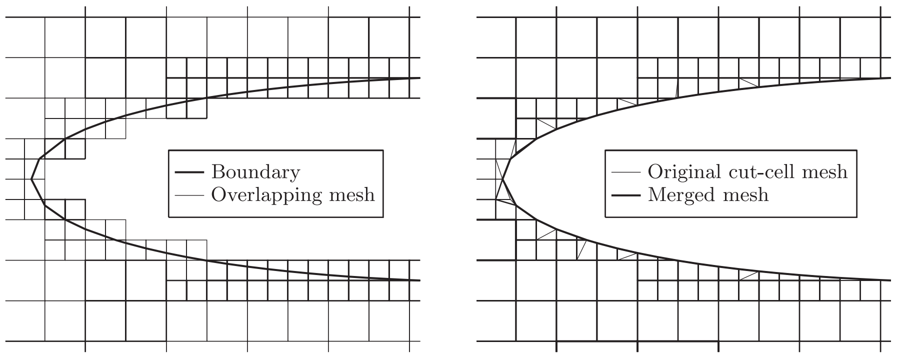

Figure 2. Shown is an example (left) of an overlapping non-conforming Cartesian mesh generated with the Harpoon software and (right) the mesh resulting after applying cell cutting (denoted 'original mesh') and cell merging.

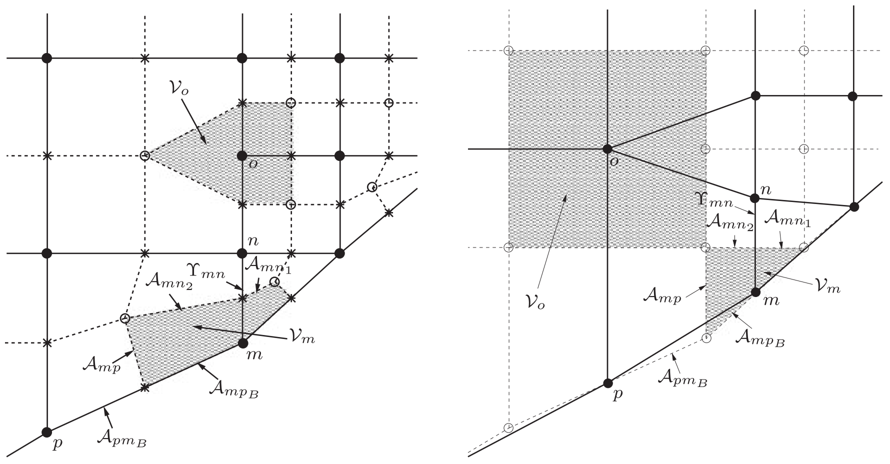

Figure 3. Schematic of the standard median-dual-mesh construction methodology (left) and a proposed alternative scheme (right). Solid lines represent the computational edges and filled circles depict the nodes. Note that both dual-mesh constructions are generated from the same mesh.

5. DUAL-MESH CONSTRUCTION

In this work both the standard dual-mesh construction of Vahdati *et al.* [38] as well as a new alternative method will be employed. The following sections describe these.

5.1. Standard dual-mesh construction

This procedure will be discussed with reference to Figure 3 (left). Dual-mesh bounding surface facets (lines) are constructed by connecting edge centres (stars in Figure 3) with element centroids (open circles). These facets are then joined to form an enclosed volume around a node (filled circles) as shown schematically for a node o in the figure. The volume associated with node o is designated $\mathscr{V}_o$.

To compute fluxes between nodes, the faces that make up the boundary surface of the control volume have to be expressed mathematically. This is effected in an edge-wise manner through the definition of so called *edge coefficients*. This is expressed for an edge between nodes m and n (denoted as Υ_{mn}) as

$$\mathbf{C}_{mn} = \sum_{\mathscr{A}_{mn_q} \in \mathscr{A}_{mn}} \mathbf{n}^{mn_q} \mathscr{A}_{mn_q} \tag{17}$$

where $\mathscr{A}_{mn_q}$ refers to a facet of this surface and $\mathbf{n}_{mn_q}$ is the outward pointing unit normal vector to $\mathscr{A}_{mn_q}$. The calculation of the domain-boundary-coefficient $\mathbf{B}_{mp}$ is similarly performed.

As $\mathbf{C}_{mn} = -\mathbf{C}_{nm}$, only one internal edge coefficient need be stored per edge. This is where edge-based schemes are considerably more efficient than element-based methods. With respect to the boundary edge coefficient, two values are stored per edge to ensure general applicability with regards to the dimension of the problem (in 3D $\mathbf{B}_{mp} \neq \mathbf{B}_{pm}$). The additional storage and computational cost is negligible as this is only needed for boundary edges.

5.2. Alternative dual-mesh construction

It is evident from the previous section that, at hanging nodes, the standard procedure results in dual-mesh volumes with geometric centres which are considerably removed from the node. In addition, the direction of the edge coefficients is significantly misaligned with that of the edge tangent. The aforementioned have serious accuracy implications. In an attempt to circumvent these, an alternative scheme was developed wherein nodes are placed in the centroids of the internal mesh elements, and the elements then constitute the dual-mesh or finite volumes. This is shown schematically in Figure 3 (right). The edges are then constructed between these nodes. The boundary elements are, however, to be given special treatment. As opposed to previously, a node is not placed in the centroid of the element, but rather on the boundary. More specifically, it is placed at the centre of the mesh boundary edge. This enables the boundary conditions to be applied as per the standard variant of the vertex-centred scheme.

Having defined the dual-mesh, edge coefficients are calculated as before. As with the standard scheme, internal edge coefficients are stored once per edge as $\mathbf{C}_{mn} = -\mathbf{C}_{nm}$. This is not possible for the boundary edge coefficients, as $\mathscr{A}_{mp_B} \neq \mathscr{A}_{pm_B}$ even in 2D. On this account, two boundary edge coefficients need to be stored. This, however, results in no alteration to the previously outlined methodology. Importantly, the furnished dual-mesh fundamentally differs from that of the standard vertex-centred scheme in that volume bounding surfaces may no longer be assumed to intersect edges midway between nodes (even by way of approximation). In order to ensure spatial accuracy on the new dual-mesh, the standard edge-based vertex-centred discretization scheme is to be accordingly generalized.

5.3. Spatial discretization

The convective term in Equation (1) viz. F^{ij} is now discretized in an edge-wise manner as

$$\int_{\mathscr{A}_m} F^{ij} n_j \, \mathrm{d}\mathscr{A} \approx \sum_{\Upsilon_{mn} \cap \mathscr{V}_m} \overline{F}^{ij}_{mn} C^j_{mn} + \sum_{\Upsilon^B_{mn} \cap \mathscr{V}_m} F^{ij}_m B^j_{mn} \tag{18}$$

where Υ_{mn} denotes the edge connecting nodes m and n, and $\overline{\mathbf{F}}_{mn}$ is the flux at the intersected dual-mesh face. In the case of the standard dual-mesh construction methodology, the latter may be calculated *via* the following relation:

$$\overline{F}^{ij}_{mn} = \overline{\rho}_{mn}\overline{u_i}_{mn}\overline{u_j}_{mn} \tag{19}$$

where the over-line quantities are calculated for a generic scalar field ϕ as

$$\overline{\phi}_{mn} = \frac{(\phi_m + \phi_n)}{2} \tag{20}$$

with ϕ_m and ϕ_n denoting the nodal (edge-vertex) values.

In the case of the proposed new dual-mesh construction procedure, the above is, however, to be generalized to account for dual-mesh bounding surfaces not in all cases being located at the edge centre:

$$\overline{\phi}_{mn} = (1 - r_{mn})\phi_m + r_{mn}\phi_n \tag{21}$$

where

$$r_{mn} = \frac{\sqrt{(x_{m_i} - x_{\mathscr{A}_i})^2\delta_{ii}}}{\sqrt{(x_{m_j} - x_{n_j})^2\delta_{jj}}} \tag{22}$$

Here, $x_{\mathscr{A}_i}$ denotes the ith Cartesian co-ordinate component of the spatial position where the bounding surface of finite volume $\mathscr{V}_m$ (namely $\mathscr{A}_m$) intersects edge $m - n$. Further, δ_{jj} is the Kronecker delta.

To ensure the overall computational efficiency of the numerical scheme, the above edge-based flux calculation procedure is employed in both forms shown viz. Equations (20) and (21). The latter, which is more costly to compute, is only employed in the irregular parts of the cut-cell Cartesian mesh, i.e. where $r_{mn} \neq \frac{1}{2}$.

5.4. Stabilization

The discretization method outlined in the previous section constitutes a second-order accurate central difference-type method, which is known to produce non-physical oscillations when solving hyperbolic partial differential equations. As pointed out previously, it is proposed to effect stabilization without the loss of second-order accuracy *via* scalar-valued dissipation (JST) developed by Jameson *et al.* [32]. This method, which does not yet feature in non-conforming Cartesian mesh solvers, has proven very effective when applied to complex flows on other types of meshes [39], while offering a balance between accuracy and computational efficiency [26].

As per Sørensen *et al.* [40], artificial dissipation is to be implemented in conjunction with first-order accurate upwinding so as to avoid non-physical spurious oscillations across discontinuities in field variables viz. across supersonic shocks. The resulting stabilizing term to be added to the right-hand side of the discretized governing equation is as follows:

$$\mathbf{D}_m = \eta\mathbf{D}^{\mathrm{sc}}_m + \mathbf{D}^{\mathrm{JST}}_m \tag{23}$$

where $\mathbf{D}^{\mathrm{sc}}_m$ and $\mathbf{D}^{\mathrm{JST}}_m$ denote the shock-capturing and artificial dissipation terms, respectively. These terms are calculated in a manner similar to Mavriplis [41], and such as to ensure applicability to both compressible and incompressible flow systems.

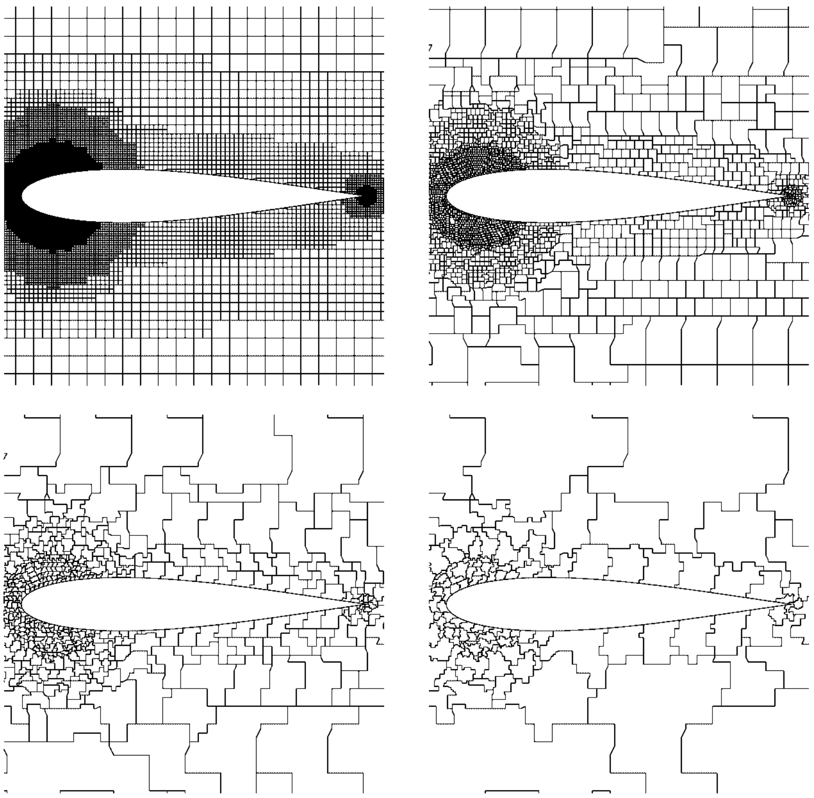

Figure 4. Van de Vooren test case—standard scheme: the fine mesh (top left) and corresponding first, second, and third coarse dual meshes.

5.5. *Temporal discretization and solution procedure*

The temporal term in Equation (1) is discretized using a four-step explicit multistage Runge–Kutta time-stepping method [27]. In the interest of computational efficiency, the stabilizing terms are only calculated in the first and third steps of the scheme.

6. SOLUTION PROCEDURE: FAS AGGLOMERATED MULTIGRID

The fast and efficient solution of the discretized equations necessitates the use of a solution acceleration algorithm. The latter is to be such that the matrix-free nature of the numerical methodology is preserved. An FAS multigrid method may be employed for this purpose. Although such a solution strategy has recently started featuring in the context of compressible flow modelling on non-conforming cut-cell Cartesian mesh methods [14, 35], a specific type of data structure is required to generate the course meshes. In this work, mesh agglomeration is based on the generic volume agglomeration strategy developed by Hannemann [42]. This is a new development in the context of the non-conforming cut-cell Cartesian meshes.

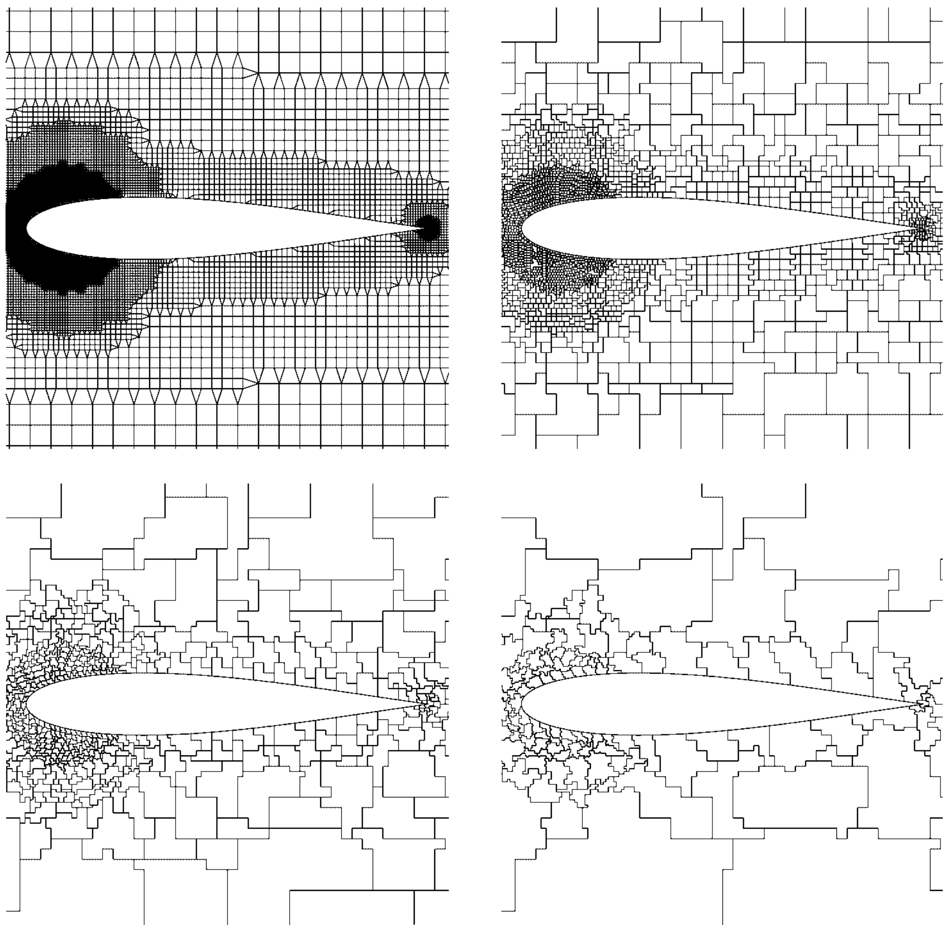

Figure 5. Van de Vooren test case—alternative scheme: the fine mesh (top left) and corresponding first, second, and third coarse dual meshes.

Similar to leading researchers in the field [3, 43–45], the FAS multigrid V-cycle methodology was employed. In the interest of flexibility from a programming point of view, a recursive form as documented by Briggs *et al.* [46] was implemented. Note that the optimal choice of the number of relaxation sweeps to be done on the upward and downward cycles had to be obtained through numerical experimentation for each test case (refer to *Numerical Tests* below). Further, the standard volume-weighted method was employed for the purposes of restriction while two methods were investigated by which to effect prolongation, viz. inverse distance weighting as per the method of Horton (as described by Watson [47]) and the linear least-squares operator. The latter was found to yield superior solver performance for all cases tested, and was consequently employed for the numerical tests.

When using the alternative discretization scheme in conjunction with FAS multigrid, r_{mn} (Equation (22)) is to be approximated for the coarse meshes. In this work, this is done in an

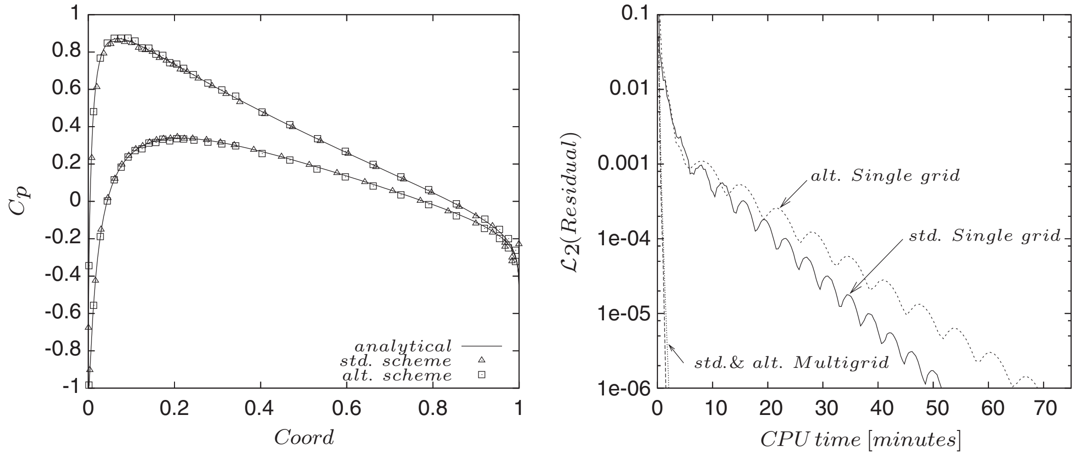

Figure 6. Van de Vooren test case—solution (left) and convergence plot (right). Here std. and alt. denote the standard and alternative discretization schemes, respectively.

edge-based manner *via* the following proposed relation:

$$r_{mn}^{2h} = \frac{\sum_{\Upsilon_{mn}\cap\mathscr{V}_m} r_{mn}^h |C_{mn}^h|}{\sum_{\Upsilon_{mn}\cap\mathscr{V}_m} |C_{mn}^h|} \tag{24}$$

where h refers to the finer mesh quantity and $2h$ to the coarser mesh.

7. NUMERICAL TESTS

The developed technology was evaluated *via* application to a number of benchmark flow problems. These included incompressible as well as compressible sub-, trans-, and supersonic systems. A mesh convergence study was performed for each case (involving at least three meshes), and when a solution was obtained to within a grid convergence index (GCI) [48] of 1–2%, the following comparisons were made:

- The predicted flow *via* both the standard and alternative schemes compared to that of a benchmark solution.
- The speed-ups in *CPU time* obtained from multigrid assessed for each discretization method.
- The two discretization methods compared in terms of accuracy, as well as computational cost.

In the interest of a rigorous investigation, the GCI error bound is in all cases quoted along with the ratio that indicates the asymptotic range of convergence. Note that the analyses were performed on a PC with Intel 3 GHz CPU and 400 MHz DDR RAM.

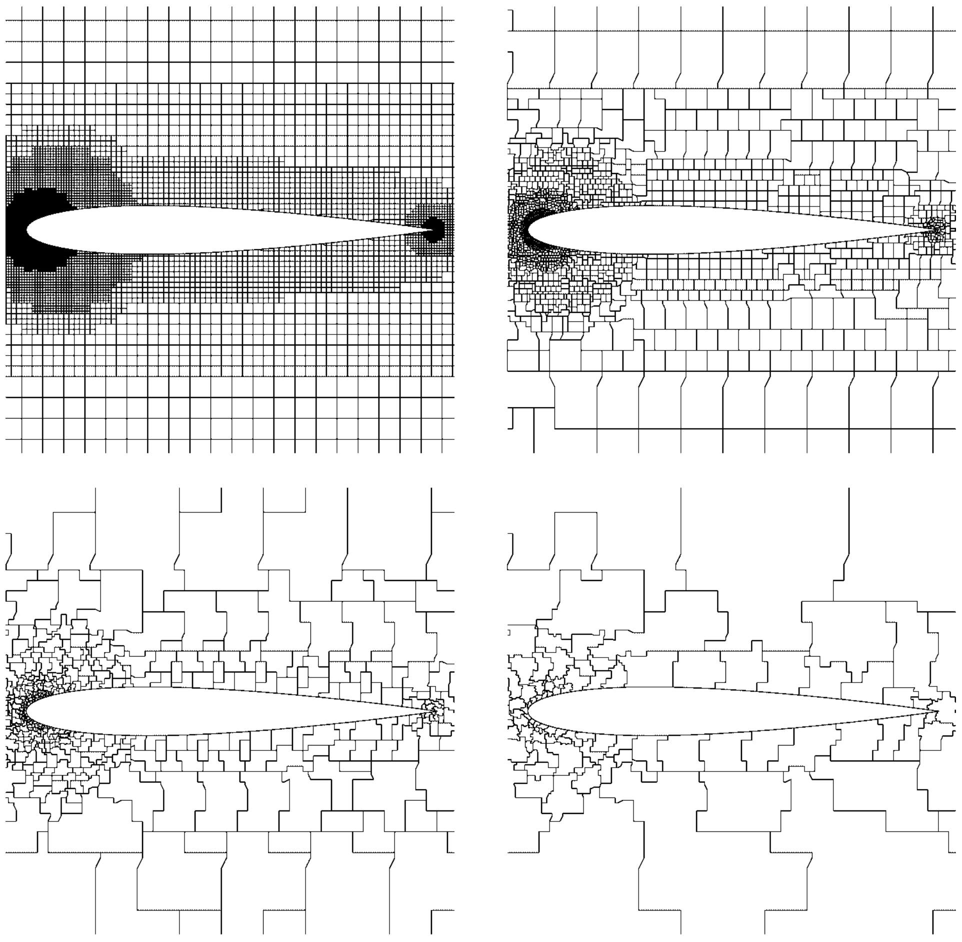
Figure 7. NACA0012 compressible flow test cases—standard scheme: the fine mesh (top left) and corresponding first, second, and fourth coarse dual meshes.

7.1. *Test case 1*: *Incompressible flow*

The first numerical example involves the incompressible flow over a *Van de Vooren* aerofoil with 15% thickness at 2° angle of attack. This problem is purely convective and contains two stagnation points, while an analytical solution is available. Slip-type boundary conditions were applied at the surface of the aerofoil while characteristic conditions were prescribed at the outer boundary (15 chord lengths from the profile). A part of the computational meshes employed for the standard and alternative dual-mesh construction strategies together with each set of multigrid agglomerated meshes are shown in Figures 4 and 5, respectively. The fine meshes in each case contain 15 420 and 14 748 computational points. Three coarse meshes were generated for the multigrid solution process, with achieved node-based coarsening ratios of between 3.8 and 4.6. The number of V-cycle relaxation iterations was set to 10 for both up- and downward sweeps.

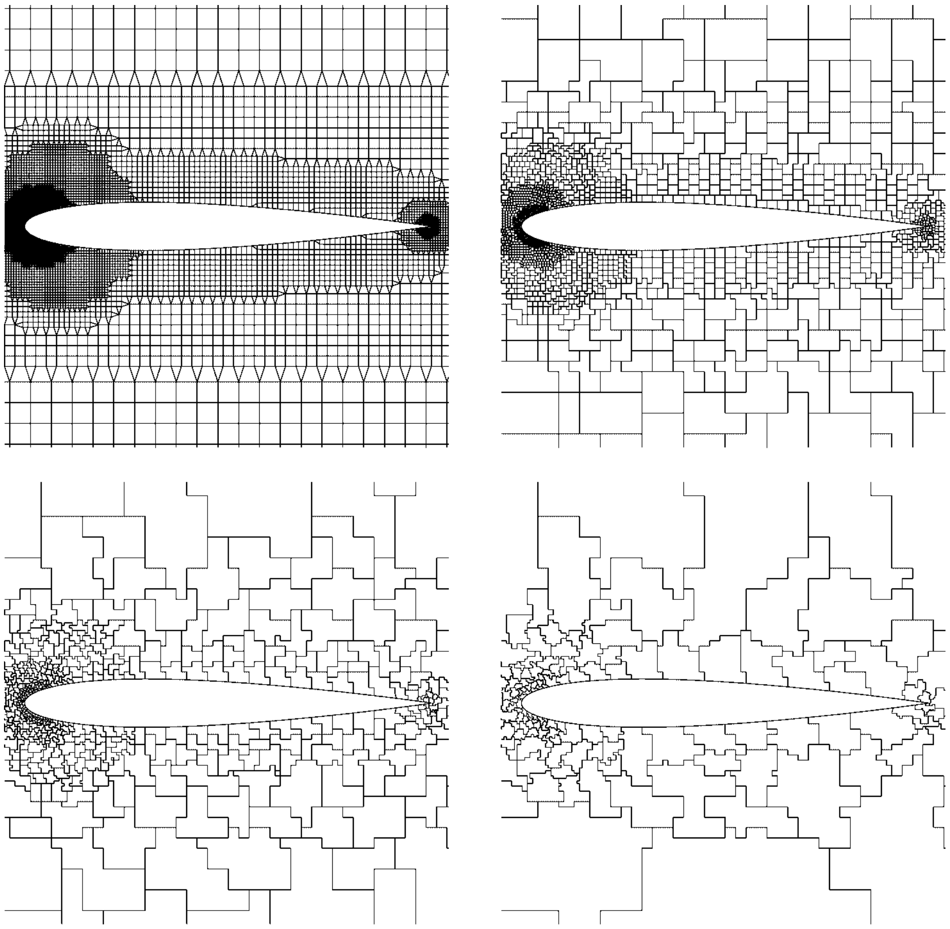

Figure 8. NACA0012 compressible flow test cases—alternative scheme: the fine mesh (top left) and corresponding first, second, and fourth coarse dual meshes.

In Figure 6, the aerofoil surface pressure coefficients predicted *via* the respective scheme variants are compared to that of the analytical solution. From the convergence plot in the figure it is clear that the alternative method only marginally out-performs the standard form in terms of computational cost. Multigrid speed-ups of 35 and 30 were achieved for the standard and alternative cases, respectively. The GCI error bound, based on the error between the computed and analytical solutions, is 0.22% for the standard case and 0.36% for the alternative case. Asymptotic convergence ratios of 0.996 and 1.11 were achieved for both cases, respectively.

7.2. Test case 2: Subsonic flow

The first compressible test case involves subsonic flow over a NACA0012 airfoil at $M = 0.63$ and 1° angle of attack. The cut-cell meshes in the vicinity of the airfoil resulting from the standard

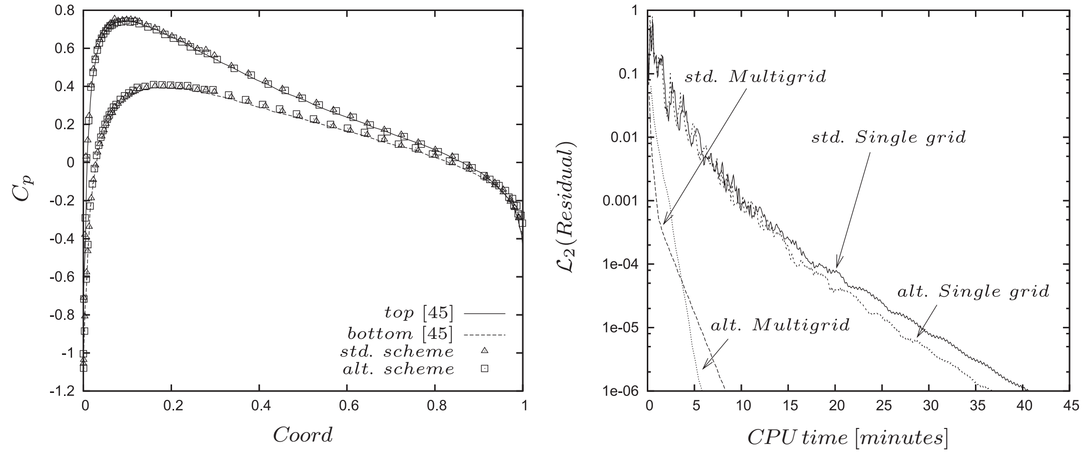

Figure 9. Solution (left) and convergence plot (right) for the subsonic test case. Here std. and alt denote the standard and alternative discretization schemes, respectively.

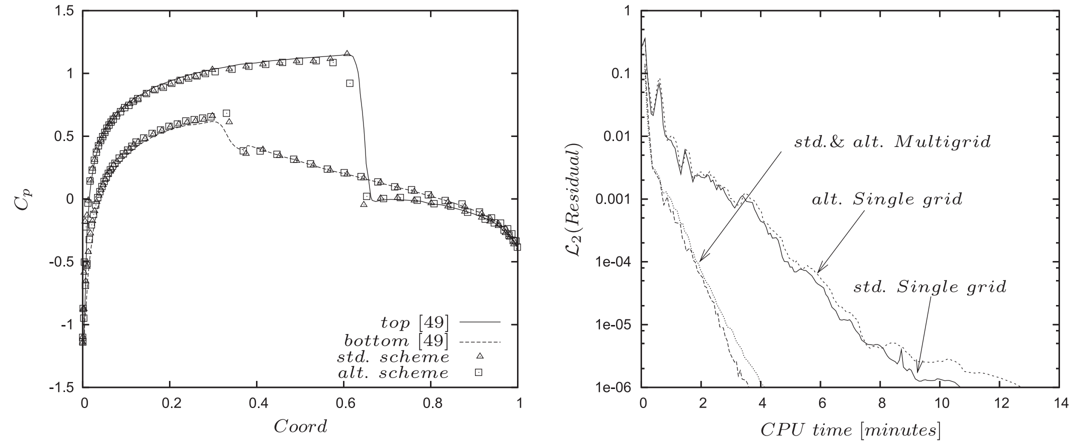

Figure 10. Solution (left) and convergence plot (right) for the transonic test case [49].

dual-mesh construction scheme, as well as a selection of the agglomerated coarse meshes, are depicted in Figure 7. A similar plot, Figure 8, is shown for the new dual-mesh construction scheme. The mesh extended 15 chord lengths away from the aerofoil surface. In the case of the standard dual-mesh construction strategy, it contains 12 819 computational points or nodes. Four coarse meshes were generated for this problem, with resulting coarsening ratios of between 4.0 and 4.8.

Figure 9 compares the solution of Sørensen [45] to the predicted pressure coefficient on the surface of the airfoil. For this test case, the standard and newly developed finite volume methodologies once again resulted in similar solutions, and at similar computational cost for both multigrid and explicit schemes. The GCI error bound is 0.41% for the standard scheme, and 0.42% for the

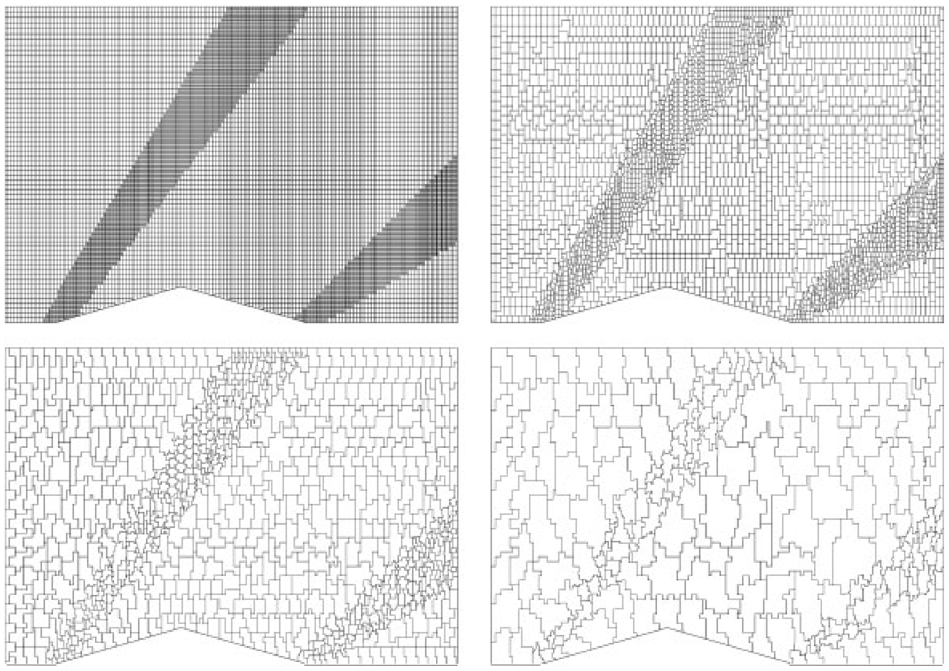
Figure 11. Supersonic test case—standard discretization scheme: fine and coarse dual meshes.

alternative method. Asymptotic convergence was achieved as indicated by GCI ratios of 1.001 and 0.9997 for the standard and alternative schemes, respectively. With regards to multigrid solution, the number of relaxation iterations in the up- and downward sweeps of the V-cycle found to produce best performance, was 10. The achieved multigrid speed-ups in computational time, based on a drop in residual of 6 orders of magnitude, were circa 19 for both discretization schemes.

7.3. *Test case 3*: *Transonic flow*

The second compressible example is concerned with the transonic flow over a NACA0012 airfoil at $M = 0.8$ and $1.25°$ angle of attack. The computational meshes used for the previous test case are once again employed (Figures 7 and 8). Figure 10 compares the airfoil surface pressure coefficients predicted *via* the two scheme variants to the benchmark solution. The GCI error bound for the standard scheme is 0.737%, and for the alternative scheme 0.835%. Asymptotic convergence was once again achieved. As shown in the figure, similar solutions were achieved and with similar computational cost and multigrid speed-up. The latter was circa 3 in CPU time for a drop in residual of 6 orders of magnitude. For this test case, the number of relaxation iterations was set to 4 on both up- and downward sweeps.

7.4. *Test case 4*: *Supersonic flow*

For the supersonic flow test case, the $10°$ double-wedge benchmark problem at $M = 2.0$ was used. The selected angle of attack is $0°$. The fine mesh for the standard and new dual-mesh construction

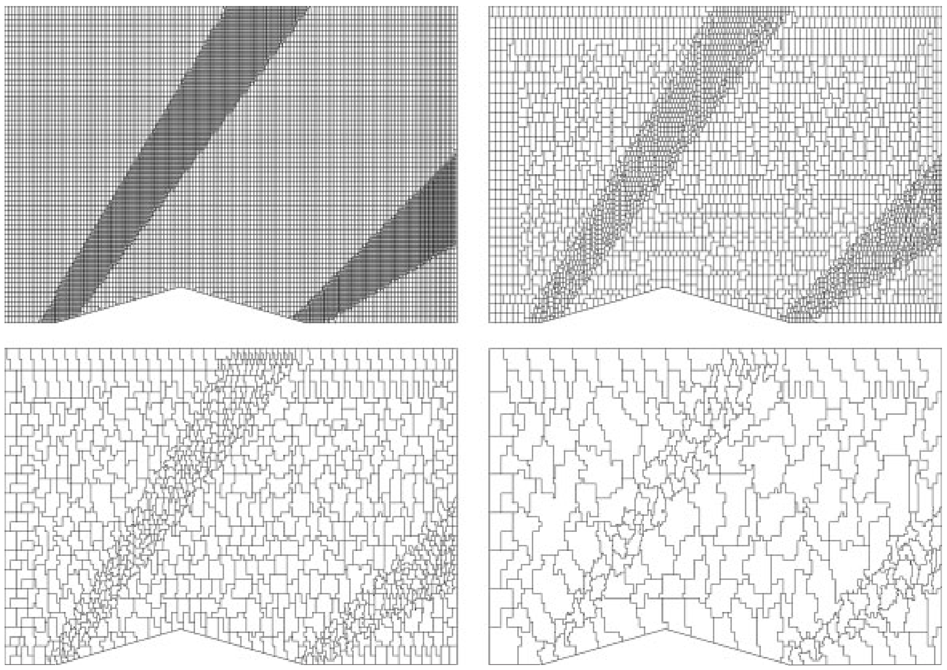

Figure 12. Supersonic test case—alternative discretization scheme: fine and coarse dual meshes.

schemes is shown together with their agglomerated coarse dual meshes in Figures 11 and 12. The standard scheme fine mesh contains 16 237 nodes with the alternative scheme containing 15 770 nodes. Node-based coarsening ratios of between 3.7 and 4.3 were achieved and a single relaxation sweep on each V-cycle mesh found to yield best convergence. The solution and convergence curves are shown in Figure 13. The GCI error bound for the standard scheme is 1.333% and the alternative scheme 1.264%, with GCI convergence ratios of 1.022 and 0.998 for the two schemes, respectively. As per the previous test cases, the two discretization schemes resulted in similarly accurate solutions and comparable computational cost and multigrid speed-ups.

8. SPATIAL CONVERGENCE OF ARTIFICIAL DISSIPATION

On consideration of the numerical test related results, it is surprising that the standard and alternative schemes resulted in such similar solution accuracies. Equation (21) thus resulted in little increased apparent accuracy. As the artificial dissipation term did not naturally lend itself to similar enhancement, it is therefore of interest to assess its influence. This will be done by, first, assessing if the dissipation terms show asymptotic convergence behaviour (these terms should tend towards zero with successive mesh refinements), and second, comparing the relative magnitude of these terms to that of the discretized spatial (convective) term.

For the purpose of this exercise, the subsonic case is considered as it does not contain any shock regions. The contribution of the artificial dissipation can therefore be assessed without the

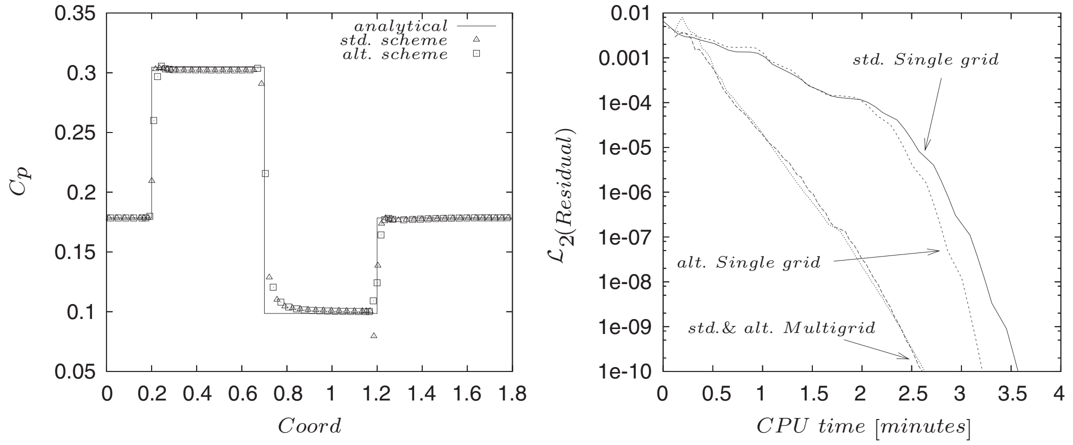

Figure 13. Supersonic test case: solution (left) and convergence plots (right).

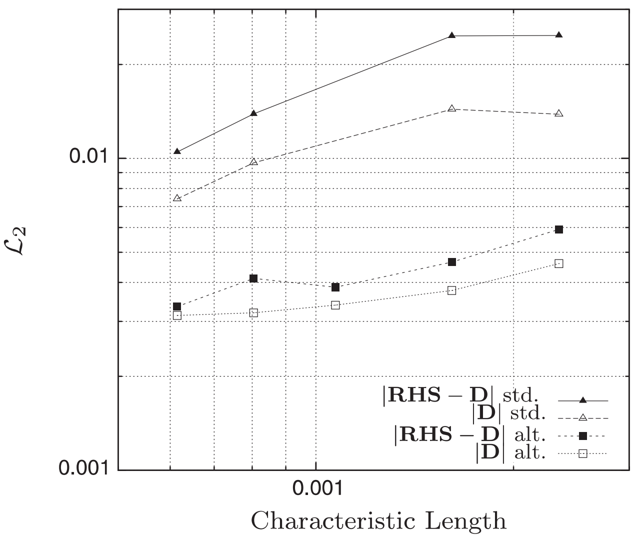

Figure 14. Spatial convergence of artificial dissipation terms on the NACA0012 subsonic test case. Note that **RHS** denotes the sum of the discretized spatial and stabilizing terms. The latter is denoted **D**.

influence of the shock capturing (first-order accurate) term. The Euclidean norm of the spatial contribution towards the residual as well as that of the dissipation terms, for various meshes, is shown in Figure 14. It can be seen from the figure that in the case of the standard scheme, the dissipation over the finest three meshes shows asymptotic-like convergence to the same extent as the discretized convective terms. Further, the dissipation is always considerably smaller than the spatial terms. However, in the case of the alternative scheme, the aforementioned convergence is not evident, which is to the detriment of solution accuracy. This is thought to be due to the off-centre face positions (with regard to placing on the edges) being accounted for in the discretization of the convective term, but not when calculating dissipation. As noted previously, this is as artificial dissipation does not naturally lend itself to account for off-centre face positions (apart from affecting the dissipation scaling parameter). It may therefore be advantageous to employ a high-resolution upwinding flux limiter in conjunction with the alternative discretization method (that does explicitly account for off-centre face positions). This is noted for future work.

9. CONCLUSION

This paper was concerned with the development of cut-cell non-conforming Cartesian mesh technology for the modelling of inviscid compressible as well as incompressible flow *via* a single governing equation set. A cut-cell method was developed with which to furnish body-fitted meshes with an overlapping non-conforming Cartesian mesh as starting point, and in a manner which is insensitive to surface definition inconsistencies. The vertex-centred edge-based finite volume scheme is used for spatial discretization purposes. An alternative dual-mesh construction strategy was developed specifically for application to non-conforming Cartesian meshes and the spatial discretization algorithm suitably enhanced. The solution was accelerated by the application of a volume-agglomerated FAS multigrid method. The developed modelling technology was validated by application to incompressible and compressible sub-, trans-, and supersonic flow problems. It is shown that the new dual-mesh construction strategy with enhanced discretization scheme does not offer a notable increase in accuracy as compared to that of the standard scheme. It is demonstrated that the former is in part due to the present implementation of the stabilization method. Further study is needed here. In terms of solver performance, the multigrid method showed speed-ups in CPU time ranging between a factor 2 and one order of magnitude.

ACKNOWLEDGEMENTS

We would like to give recognition and thanks to Dr C. P. Crosby for his foresight in being the original initiator of this work. Special recognition and thanks is due to Angus Brown, John O'Mahony, and Jonathan Fouche at Denel Aerospace Systems for their interest and continued support. Without you, the development of the intended capabilities would not have been possible.

REFERENCES

1. Charlton EF. An octree solution to conservation-laws over arbitrary regions (OSCAR) with applications to aircraft aerodynamics. *Doctor of Philosophy Thesis*, University of Michigan, Michigan, 1997.
2. Jameson A. A perspective on computational algorithms for aerodynamic analysis and design. *Progress in Aerospace Sciences* 2001; **37**(2):197–243.
3. Blazek J. *Computational Fluid Dynamics*: *Principles and Applications* (1st edn). Elsevier: Oxford, 2001.
4. Aftosmis MJ. Solution adaptive Cartesian grid methods for aerodynamic flows with complex geometries. *28th Computational Fluid Dynamics*, Volume Lecture Series 1997–2002, von Karman Institute for Fluid Dynamics, 1997.
5. Berger M, Aftosmis M, Adomavicius G. Parallel multigrid Cartesian meshes with complex geometry. *Proceedings of the Parallel CFD Conference*, Holland, 2000.
6. Yang G, Causon M, Ingram DM, Saunders R, Batten P. A Cartesian cut cell method for compressible flows part A: static body problems. *The Aeronautical Journal* 1997; **101**(1002):47–56.
7. Aftosmis MJ. Robust and efficient Cartesian mesh generation for component-based geometry. *AIAA Journal* 1998; **36**(6):952–960.

8. Ogawa T. Development of a flow solver using the adaptive Cartesian mesh algorithm for wind environment assessment. *Journal of Wind Engineering and Industrial Aerodynamics* 1999; **81**:377–389.
9. Murman SM, Aftosmis MJ, Berger MJ. Numerical simulation of rolling-airframes using a multi-level Cartesian method. *AIAA*, volume Paper 2002-2798, St. Louis, 2002.
10. Wang ZJ, Chen RF. Anisotropic solution-adaptive Cartesian grid method for turbulent flow simulation. *AIAA Journal* 2002; **40**:1969–1978.
11. Kirkpatrick MP, Armfield SW, Kent JH. A representation of curved boundaries for the solution of the Navier–Stokes equations on a staggered three dimensional Cartesian grid. *Journal of Computational Physics* 2003; **184**:1–36.
12. Murman SM, Aftosmis MJ, Berger MJ. Simulations of 6-DOF motion with a Cartesian method. *AIAA*, volume Paper 2003-1246, Reno, NV, 2003.
13. Coirier WJ, Powell KG. An accuracy assessment of Cartesian-mesh approaches for the Euler equations. *Journal of Computational Physics* 1995; **117**:121–131.
14. Aftosmis MJ, Berger MJ, Adomavicius G. A parallel multilevel method for adaptively refined Cartesian grids with embedded boundaries. *AIAA*, volume Paper 2000-0808, Reno, NV, 2000.
15. Ingram DM, Causon DM, Mingham CG. Developments in Cartesian cut cell methods. *Mathematics and Computers in Simulation* 2003; **61**:561–572.
16. Dawes WN. Building blocks towards VR-based flow sculpting. *AIAA*, Paper number AIAA 2005-1156, Reno, NV, 2005.
17. Popinet S. Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries. *Journal of Computational Physics* 2003; **190**:572–600.
18. Clarke DK, Salas MD, Hassan HA. Euler calculations for multi-element airfoils using Cartesian grids. *AIAA Journal* 1986; **24**(3):353–358.
19. Yang G, Causon DM, Ingram DM. Cartesian cut-cell method for axisymmetric separating body flows. *AIAA Journal* 1999; **37**(8):905–911.
20. Tucker PG, Pan Z. A Cartesian cut cell method for incompressible viscous flow. *Applied Mathematical Modelling* 2000; **24**:591–606.
21. Causon DM, Ingram DM, Mingham CG, Yang G, Pearson RV. Calculation of shallow water flows using a Cartesian cut cell approach. *Advances in Water Resources* 2000; **23**:545–562.
22. Ham FE, Lien FS, Strong AB. A Cartesian grid method with transient anisotropic adaptation. *Journal of Computational Physics* 2002; **179**:469–494.
23. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill: New York, 1980.
24. Chorin AJ. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics* 1967; **2**:12–26.
25. Tamamidis P, Zhang GQ, Assanis DN. Comparison of pressure-based and artificial compressibility methods for solving 3D steady incompressible viscous flows. *Journal of Computational Physics* 1996; **124**:1–13.
26. Lin FB, Sotiropoulos F. Assessment of artificial dissipation models for three-dimensional incompressible flow solutions. *Journal of Fluids Engineering* 1997; **119**:331–340.
27. Malan AG, Lewis RW, Nithiarasu P. An improved unsteady, unstructured, artificial compressibility, finite volume scheme for viscous incompressible flows: part I. Theory and implementation. *International Journal for Numerical Methods in Engineering* 2002; **54**(5):695–714.
28. Nithiarasu P. An efficient artificial compressibility (ac) scheme based on the characteristic based split (cbs) method for incompressible flow. *International Journal for Numerical Methods in Engineering* 2003; **56**(13):1815–1845.
29. Ye T, Mittal R, Udaykumar HS, Shyy W. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics* 1999; **156**:209–240.
30. Hirsch C. *Numerical Computation of Internal and External Flows*, *Computational Methods for Inviscid and Viscous Flows*, vol. 2. Wiley: New York, 1990.
31. Beam RM, Warming RF. An implicit finite difference algorithm for hyperbolic systems in conservation law form—application to Eulerian gas-dynamics equations. *Journal of Computational Physics* 1976; **22**:87–110.
32. Jameson A, Schmidt W, Turkel E. Numerical simulation of the Euler equations by finite volume methods using Runge–Kutta time-stepping schemes. *AIAA*, Paper 81-1259, *AIAA 5th Computational Fluid Dynamics Conference*, Palo Alto, California, 1981.
33. MacCormack RW, Baldwin BS. A numerical method for solving the Navier–Stokes equations with application to shock-boundary layer interactions. *AIAA*, Paper 1-75, 1975.
34. Zhang LP, Wang ZJ. A block LU-SGS implicit dual time-stepping algorithm for hybrid dynamic meshes. *Computers and Fluids* 2004; **33**:891–916.

35. Murman SM, Aftosmis MJ, Berger MJ. Implicit approaches for moving boundaries in a 3-D Cartesian method. *AIAA*, volume Paper 2003-1119, Reno, NV, 2003.
36. Harpoon. *Harpoon*. CEI, http://www.ceintl.com/products/ harpoon.html, version 1.4.0 a_2 edition, 2005.
37. OpenCASCADE. *OpenCASCADE*. http://www.opencascade.org/, version 5.2 edition, 2005.
38. Vahdati M, Morgan K, Peraire J, Hassan O. A cell-vertex upwind unstructured grid solution procedure for high-speed compressible viscous flow. *International Conference on Hypersonic Aerodynamics*, Royal Aeronautical Society: London, 1989; 12.1–12.22.
39. Swanson RC, Turkel E. On central-difference and upwind schemes. *Journal of Computational Physics* 1992; **101**:297–306.
40. Sørensen KA, Hassan O, Morgan K, Weatherill NP. Agglomerated multigrid on hybrid unstructured meshes for compressible flow. *International Journal for Numerical Methods in Fluids* 2002; **40**(3–4):593–603.
41. Mavriplis DJ. Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes. *AIAA Journal* 1990; **28**(2):213–221.
42. Hannemann V. Structured multigrid agglomeration on a data structure for unstructured meshes. *Numerical Methods for Fluid Dynamics*, vol. VII. Oxford University Press: Oxford, 2001; 329–337.
43. Lallemand MH, Steve H, Dervieux A. Unstructured multigridding by volume agglomeration: current status. *Computers and Fluids* 1992; **21**(3):397–433.
44. Mavriplis DJ, Venkatakrishnan V. Agglomeration multigrid for two-dimensional viscous flows. *Computers and Fluids* 1995; **24**(5):553–570.
45. Sørensen KA. A multigrid acceleration procedure for the solution of compressible fluid flows on unstructured hybrid meshes. *Doctor of Philosophy Thesis*, University of Wales Swansea, Swansea, 2002.
46. Briggs WL, Van Emden Henson, McCormick SF. *A Multigrid Tutorial* (2nd edn). Society for Applied and Industrial Mathematics: Philadelphia, PA, 2000.
47. Watson DF. *Contouring*, *A Guide to the Analysis and Display of Spatial Data*. Pergamon Press: New York, 1992.
48. Roache PJ. *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers: New Mexico, 1998.
49. AGARD Fluid Dynamics Panel. *Test Cases for Inviscid Flow Field Methods*. AGARD Advisory Report AR-211, 1985.