INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2009; **61**:970–986
Published online 12 January 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1992

A local domain-free discretization method to simulate three-dimensional compressible inviscid flows

C. H. Zhou[1] and C. Shu[1,2,*,†]

[1]*Department of Aerodynamics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, People's Republic of China*
[2]*Department of Mechanical Engineering, National University of Singapore, Singapore 119260, Singapore*

SUMMARY

In this paper, the domain-free discretization method (DFD) is extended to simulate the three-dimensional compressible inviscid flows governed by Euler equations. The discretization strategy of DFD is that the discrete form of governing equations at an interior point may involve some points outside the solution domain. The functional values at the exterior-dependent points are updated at each time step by extrapolation along the wall normal direction in conjunction with the wall boundary conditions and the simplified momentum equation in the vicinity of the wall. Spatial discretization is achieved with the help of the finite element Galerkin approximation. The concept of 'osculating plane' is adopted, with which the local DFD can be easily implemented for the three-dimensional case. Geometry-adaptive tetrahedral mesh is employed for three-dimensional calculations. Finally, we validate the DFD method for three-dimensional compressible inviscid flow simulations by computing transonic flows over the ONERA M6 wing. Comparison with the reference experimental data and numerical results on boundary-conforming grid was displayed and the results show that the present DFD results compare very well with the reference data. Copyright © 2009 John Wiley & Sons, Ltd.

Received 25 August 2008; Revised 25 November 2008; Accepted 29 November 2008

KEY WORDS: domain-free discretization; boundary conditions; non-boundary-conforming methods; compressible inviscid flows

1. INTRODUCTION

In [1], Shu and Fan proposed a domain-free discretization (DFD) method to solve partial differential equations (PDEs) on irregular domains. So far, the DFD method has been successfully applied to

*Correspondence to: C. Shu, Department of Mechanical Engineering, National University of Singapore, Singapore 119260, Singapore.
†E-mail: mpeshuc@nus.edu.sg

Contract/grant sponsor: National Science Foundation of China; contract/grant number: 10772083

simulate two-dimensional incompressible, compressible, inviscid, and viscous flows around fixed and moving bodies [2–5]. This paper is the first endeavor to present the local DFD method for the simulation of three-dimensional inviscid flows.

DFD is inspired from the analytical method. It is well known that the analytical method takes two separate steps to get the closed-form solution of a PDE. In the first step, a general solution is pursued, which depends only on the given PDE. Then in the second step, the expression of the general solution is substituted into boundary conditions to determine the unknown coefficients in the general solution. Clearly, the solution domain is only involved in the second step when boundary conditions are implemented. In contrast, the conventional numerical method solves the PDE in just one step, in which the PDE is discretized on the solution domain with proper implementation of boundary conditions. We can see clearly that the discretization of the PDE in a numerical method is directly coupled with boundary conditions and thus is problem dependent. To overcome the drawbacks of conventional numerical methods that strongly couple the PDE with the solution domain, the DFD method was developed from the hint of the analytical method.

In the DFD, the implementation of boundary conditions and the discretization of the PDE are treated separately as in the analytical method. The discrete form of the PDE at a point inside the solution domain may involve some points outside the domain, which serves as the role to implement the boundary condition. The key process in the DFD method is to evaluate the functional values at the points outside the solution domain. The evaluation can be done from the ‘solution-extension’. As we know, the smooth solution of a PDE inside the solution domain satisfies the PDE not only at the interior points, but also at the exterior points. Thus, we can simply substitute the coordinates of exterior points into the closed-form solution to obtain the functional values. The closed-form solution is usually unknown for the case of doing numerical computations, but it is possible to get some approximate form of the solution in the local region, for example, along a mesh line.

In the earlier applications of DFD [1–3], the approximate form of solution is pursued along the whole mesh line that only involves two boundary points. This way is not suitable for more complex domains. To make the method more general, the local DFD was developed in [4, 5]. In the local DFD, the low-order schemes are adopted for spatial discretization and also for approximate form of the solution near the wall boundary. In [5], the local DFD method was applied to two-dimensional compressible Euler and Navier–Stokes equations in conservative form. The functional values at the exterior-dependent points are updated at each time step by the extrapolation along the boundary normal direction in conjunction with wall boundary conditions and the simplified momentum equation in the vicinity of the wall. The functional values at the exterior points are computed from the current values of flow variables at some interior associated points or some values of the known body movement.

As described above, the DFD method belongs to non-boundary-conforming methods since a solid wall can be immersed on the grid, usually the Cartesian grid. In recent years, non-boundary-conforming numerical methods are attracting a lot of attention because they eliminate the tedious task of mesh generation for complex geometry required by classical boundary-conforming methods and can simulate flows around multi-bodies with large deformations and arbitrary movements in a straightforward manner. For compressible flow simulations, the Cartesian cut-cell method and the ghost-cell method are the widely used two approaches among the non-boundary-conforming methods.

In the Cartesian cut-cell approach, the Cartesian grid cells that intersect with the immersed body are reconstructed so that the local boundary conformity is achieved. This approach allows boundary conditions on a solid wall to be imposed in a manner similar to that used in traditional boundary-conforming methods. A difficulty encountered in the implementation arises from the large number of possible intersections between the fixed grid and body surface, leading to the formation of various irregular cells. Furthermore, the generation of irregular cells with very small volume can impact the conservation and stability properties of the method. This is the well-known small-cell problem for cut-cell method, which has been addressed by merging the small control volumes with nearby larger ones proposed by Quirk [6] or hybridizing the conservative discretization with a stable non-conservative discretization proposed by Collela *et al.* [7]. Recent work on successful applications of the Cartesian cut-cell method can be found in [7–10].

The crux of ghost-cell method proposed by Dadone and Grossman in [11] is the curvature-corrected symmetry technique developed for body-fitted grids. The method introduces ghost cells near the wall boundary, the flow variables at the centers of which are evaluated according to an assumed flow-field model in the vicinity of the wall. The assumed model consists of a vortex flow that satisfies the normal momentum equation and the non-penetration condition. This flow-field model locally enforces symmetry conditions for entropy and total enthalpy along a normal to the body surface. In the ghost-cell methodology, flow variables at all centers exterior to the body are computed with fluxes at the surrounding cell edges and there is no need for special treatment corresponding to cut cells. To avoid grid clustering near the body to be maintained to the far-field boundary, a far-field coarsening based on *iblanking* approach was introduced to preserve the structured nature of the Cartesian grid [12]. Recently, the ghost-cell method was applied for the calculation of three-dimensional inviscid flows around fixed boundaries [12].

Apart from the non-boundary-conforming methods, the so-called mesh free or meshless methods should also be mentioned. This class of methods reduces the high cost of mesh generation by completely discarding the pre-specified mesh connectivity in the process of spatial discretization. The mesh-free methods can be roughly grouped into two categories. One is based on the polynomial approximation [13, 14], and the other is to use the radial basis functions (RBFs) as interpolants [15–17]. The difference between the DFD method and a mesh-free method is very clear. The DFD method provides a strategy for spatial discretization. It still needs a numerical method to discretize the spatial derivatives. The key in the DFD method is that the discrete form of PDE may involve some points outside the solution domain, where the functional values are evaluated by the local solution forms. In contrast, the mesh-free method constructs functional approximation or interpolation from information at a set of scattered nodes within the solution domain.

The DFD method has already been successfully applied to solve various two-dimensional flows. It can be easily applied for problems with complex geometry and moving boundary. In this paper, we extend this method to simulate the three-dimensional compressible flows governed by Euler equations. We adopt the concept of 'osculating plane' introduced in [18] and modified by Dadone and Grossman in [12]. With this concept, the two-dimensional DFD can be easily extended to the three-dimensional case. Owing to their flexibility, tetrahedral meshes are employed for three-dimensional calculations. At the beginning, we generate an initial tetrahedral mesh from hexahedral cells of a coarse Cartesian mesh that is independent of the geometry. The finally used tetrahedral mesh is obtained by refining the initial mesh gradually from the body surface to the farfield. To validate the local DFD method for three-dimensional compressible inviscid flow simulations, the transonic flows around the ONERA M6 wing were simulated, and the obtained numerical results were compared with available data in the literature.

2. GOVERNING EQUATIONS

Using a Cartesian coordinate system, the three-dimensional Euler equations in conservative form for compressible inviscid flows can be written as

$$\frac{\partial \mathbf{w}}{\partial t}+\frac{\partial \mathbf{f}}{\partial x}+\frac{\partial \mathbf{g}}{\partial y}+\frac{\partial \mathbf{h}}{\partial z}=0 \tag{1}$$

where **w** is the vector of conservative variables, and **f**, **g**, **h** are the convective flux vectors. **w**, **f**, **g**, and **h** are given by

$$\mathbf{w}=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{bmatrix},\quad \mathbf{f}=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ u(\rho E+p)\end{bmatrix},\quad \mathbf{g}=\begin{bmatrix}\rho v\\ \rho vu\\ \rho v^2+p\\ \rho vw\\ v(\rho E+p)\end{bmatrix},\quad \mathbf{h}=\begin{bmatrix}\rho w\\ \rho wu\\ \rho wv\\ \rho w^2+p\\ w(\rho E+p)\end{bmatrix} \tag{2}$$

In (2), ρ represents the fluid density, u, v, and w are the Cartesian components of the velocity, E is the total energy, and p is the pressure which can be calculated from the following equation of state for a perfect gas:

$$p=\rho(\kappa-1)\left(E-\frac{u^2+v^2+w^2}{2}\right) \tag{3}$$

where κ is the ratio of specific heats of fluid and taken as 1.4 for air.

3. SPATIAL DISCRETIZATION EMPLOYED IN DFD

As described in the introduction, in the DFD method, the wall boundary can be superimposed upon the computational mesh. We suppose that $\Omega \subset R^3$ is a connected open set containing a body ω, and denote the boundary of Ω by Γ. With h a space discretization step, a tetrahedrization T_h of $\bar{\Omega}$ is introduced.

DFD is a discretization strategy. Its essence is that the discrete form of governing equation can involve some points outside the solution domain. It still needs a numerical approach to do discretization and transfer the differential equation into a discrete form. The spatial discretization employed here is the direct three-dimensional extension of the two-dimensional Galerkin finite element approach proposed by Mavriplis and Jameson in [19]. The procedure begins by storing flow variables at the mesh vertices, and piecewise linear flux functions are used over the individual tetrahedra. Let **F** denote the convective flux tensor, the Cartesian components of which are **f**, **g**, and **h**. The Euler equations can be rewritten in the vector notation as

$$\frac{\partial \mathbf{w}}{\partial t}+\nabla\cdot\mathbf{F}=0 \tag{4}$$

Multiplying by a test function ϕ and integrating over the domain yields the following Galerkin formulation:

$$\frac{\partial}{\partial t}\iiint_{\Omega}\phi\mathbf{w}\,\mathrm{d}\Omega+\iiint_{\Omega}\phi\nabla\cdot\mathbf{F}\,\mathrm{d}\Omega=0 \tag{5}$$

Integrating the flux integral by parts gives

$$\frac{\partial}{\partial t}\iiint_{\Omega}\phi\mathbf{w}\,\mathrm{d}\Omega=\iiint_{\Omega}\mathbf{F}\cdot\nabla\phi\,\mathrm{d}\Omega-\iint_{\Gamma}\mathbf{n}\cdot\mathbf{F}\phi\,\mathrm{d}\Gamma \tag{6}$$

where $\mathbf{n}$ is the outward normal unit vector at Γ. In order to evaluate the flux balance at each vertex P, ϕ is taken as a piecewise linear function which is equal to 1 at P and vanishes at all other vertices. Therefore, the integrals in the above equation are nonzero only over the tetrahedrons that contain the vertex P and thus define the influence domain of node P. Evaluating the flux integral and employing the concept of a lumped mass matrix while integrating the term on the left-hand side (LHS) of (6), one obtains

$$\Omega_P\frac{\partial\mathbf{w}_P}{\partial t}=\sum_{e=1}^{n}\frac{\mathbf{F}_A+\mathbf{F}_B+\mathbf{F}_C}{3}\cdot\Delta\mathbf{S}_{ABC} \tag{7}$$

In the above formulation, the summation is over all the tetrahedra in the influence domain of P and Ω_p represents the volume of the domain. As illustrated in Figure 1, $\Delta\mathbf{S}_{ABC}$ represents the directed (normal) triangle area of the face of each tetrahedron on the outer boundary of the influence domain. $\mathbf{F}_A$, $\mathbf{F}_B$, and $\mathbf{F}_C$ are the convective fluxes at the three vertices of this triangle.

The domain Ω is an auxiliary domain and the solution domain is $\Omega\backslash\omega$. In this work, the boundary conditions at Γ are classical and a Stegger–Warming flux splitting scheme [20] is used for in- and out-flow boundaries.

Here, we should indicate that there is no imposition of wall boundary conditions in the spatial discretization and the discrete form of the governing equations is irrelevant to the solution domain according to the concept of DFD. The wall boundary conditions will be implemented via evaluating the flow variables at exterior-dependent points as discussed in Section 5. This is the main difference from the conventional approaches.

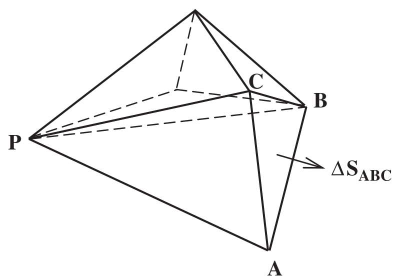

Figure 1. Directed area of influence-domain-boundary face of a tetrahedron.

4. ARTIFICIAL DISSIPATION AND TIME MARCHING

The spatial discretization based on Galerkin finite element approach corresponds to central differencing in structured mesh topology. The artificial dissipation operator proposed by Jameson *et al.* in [21] is adopted to prevent oscillations in the vicinity of a shock and damp high frequency errors. For completeness, a brief description of the construction of this operator is given below. It is a blend of undivided Laplacian and biharmonic operators in the flow field. At the node i, the undivided Laplacian of $\mathbf{w}$ can be approximated as the summation of the differences of $\mathbf{w}$ along all edges meeting at i

$$(\nabla^2 \mathbf{w})_i = \sum_{k=1}^{n} (\mathbf{w}_k - \mathbf{w}_i) \tag{8}$$

where n represents the number of edges meeting at i. Since the biharmonic operator can be viewed as a Laplacian of another Laplacian, the artificial dissipation term can be expressed as

$$D_i(\mathbf{w}) = \sum_{k=1}^{n} \alpha_k \{\varepsilon_{2k}(\mathbf{w}_i - \mathbf{w}_k) - \varepsilon_{4k}(\nabla^2 \mathbf{w}_i - \nabla^2 \mathbf{w}_k)\} \tag{9}$$

where ε_2 and ε_4 are adaptive coefficients designed to switch on enough dissipation where it is needed, and α is a factor proportional to the maximum eigenvalue of the Euler equations. In the summation of Equation (9), the coefficient ε_4 is set be zero for all the edges intersecting with the solid wall. As presented in the next section, this cancellation of the action of biharmonic operator for the wall-intersected edges will simplify the evaluation of flow variables at exterior-dependent points. Numerical experiments show that the cancellation does not affect the stability of the scheme.

The spatial discretization transforms (1) into the following set of coupled ordinary differential equations:

$$\Omega_i \frac{\mathrm{d}\mathbf{w}_i}{\mathrm{d}t} + Q_i(\mathbf{w}) - D_i(\mathbf{w}) = 0, \quad i = 1, 2, 3, \ldots, N \tag{10}$$

where N is the number of the computational mesh nodes, $Q_i(\mathbf{w})$ represents the discrete approximation to the convective fluxes, and $D_i(\mathbf{w})$ represents the artificial dissipation terms. These equations are integrated in time using a five-stage, hybrid, time-stepping scheme proposed in [19].

5. EVALUATION OF FLOW VARIABLES AT EXTERIOR-DEPENDENT POINTS AND IMPLEMENTATION OF WALL BOUNDARY CONDITIONS BY LOCAL DFD

According to spatial discretization and introduction of the artificial dissipation term presented in Sections 3 and 4, the calculation of the vector of conservative variables at any mesh point inside the solution domain depends on the conservative vector and its undivided Laplacian at each 'surrounding' point connected to this point by an edge of tetrahedron. For an edge intersecting with the wall boundary, the interior end point is a computed node at which all flow variables are obtained by solving the governing equations, and the exterior end point is one of the dependent points of this computed node. All these kinds of interior computed nodes near the wall boundary have at least one dependent point outside the solution domain as shown in Figure 2. Setting

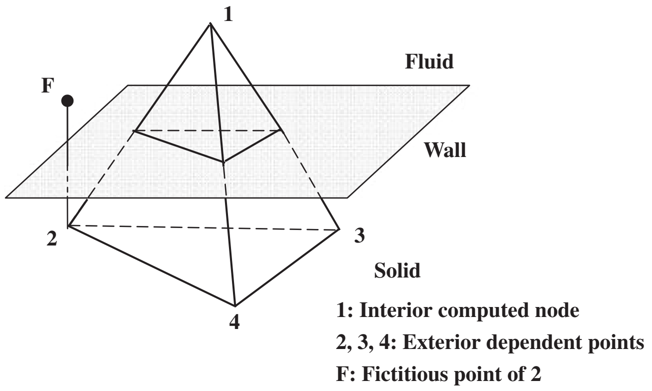

Figure 2. Interior computed node, exterior-dependent points and fictitious point.

$\varepsilon_4 = 0$ in (9) for wall-intersected edges means that only evaluation of the conservative vector at the exterior-dependent point is required in calculations, and there is no need to compute its undivided Laplacian. The key process of the DFD method is how to evaluate the functional values at the exterior-dependent points, and in this process the wall boundary conditions will be imposed.

In the present work, the values of flow variables at the exterior-dependent points are extrapolated from the flow field or determined by the local simplified flow equations, along the normal direction to the wall boundary. Since all flow variables are approximated by piecewise linear functions in the spatial discretization, the linear extrapolation will be reasonable. Therefore, some points on the normal to the boundary and inside the solution domain should be constructed for the extrapolation and the solution of the local simplified flow equations. These points may not be the mesh points; hence, we call them fictitious points.

For a given exterior-dependent point, the mirror image to wall is chosen as its fictitious point as illustrated in Figure 2. Suppose that this fictitious point locates in a tetrahedron $\tau \in T_h$. If the four vertices of τ are all inside the solution domain, it can be used to evaluate the flow variables at the fictitious point by the linear interpolation. This interpolation tetrahedron is denoted by τ_{I}. With such a definition of fictitious point, in the treatment of an impermeable wall for inviscid flow computations, the normal component of the velocity at the exterior-dependent point can be set to be antisymmetric with respect to the normal component of the velocity at the fictitious point in the reference frame. This prescription avoids the overshoot of an extrapolated velocity when the wall boundary is very close to the interior computed node.

If the four vertices of τ_{I} are not all inside the solution domain, the values of the flow variables at the exterior vertex are not known currently. This case appears frequently in the DFD computations.

To evaluate the normal component of velocity at the fictitious point, a new interpolation tetrahedron $\tau_{\mathrm{I}} \notin T_h$ should be constructed locally. A simple example is illustrated in Figure 3. The construction can be done easily by linking the interior vertices of τ to one of the intersection points of edges with the wall boundary so that the formed tetrahedron contains the fictitious point and has a maximum volume value. The normal component of velocity at intersection point can be obtained directly from the known body movement. This division of some wall-intersected cells is only for the construction of interpolation tetrahedrons to evaluate the normal velocity at some fictitious points, and not applicable for solving governing equations.

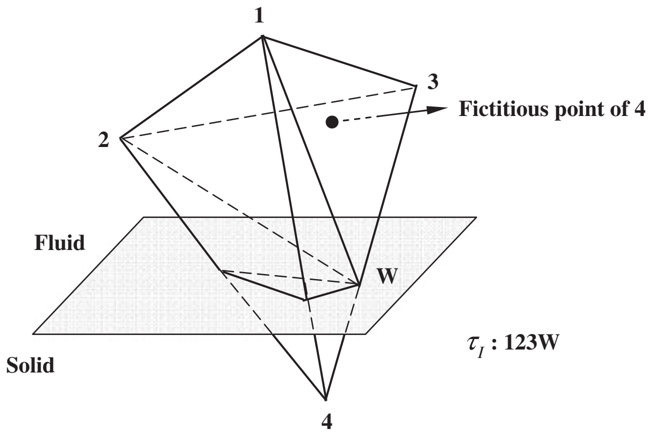

Figure 3. Local construction of the interpolation tetrahedron $\tau_{\mathrm{I}} \notin T_h$.

In the case of $\tau_{\mathrm{I}} \notin T_h$, the values of pressure, density, and tangent components of velocity at the wall-intersection points are not known, thus the values of these variables at the mirroring fictitious point cannot be determined. To overcome this difficulty, we construct another fictitious point for a given exterior-dependent point. In this case, the point that satisfies: (1) on the normal to the wall that passes the exterior point; (2) the four vertices of the tetrahedron containing this point (i.e. the interpolation tetrahedron) are all inside the solution domain; (3) closest to the wall is defined as the additional fictitious point for the evaluation of pressure, density, and tangent components of velocity at the exterior-dependent point.

Thus, the flow variables at the four vertices of the interpolation tetrahedron τ_{I} either take the current computed values or are determined from the body movement. By the linear interpolation over τ_{I}, the values of flow variables at the fictitious point can be obtained. For example, the pressure at the fictitious point is computed by

$$p_f = \sum_{k=1}^{4} \phi_k(x_f, y_f, z_f) p_k \tag{11}$$

where the summation over k refers to the four vertices of the interpolation tetrahedron τ_{I}, and $\phi_k(x_f, y_f, z_f)$ is the test function for each vertex at the fictitious point.

The values of flow variables at the fictitious points, at each time step, are used in conjunction with the wall boundary conditions and the simplified flow equations near the wall to update continually the values of flow variables at the exterior-dependent points.

Now, we discuss the evaluation of flow variables at an exterior-dependent point in detail. For three dimensions, an intrinsic coordinate system proposed by Serrin [18] can be taken at first. This local system is composed of the streamwise direction, the normal direction, and the binormal direction. The normal direction is the direction normal to the streamwise direction and parallel to the plane containing the streamwise direction and the gradient of pressure. The plane containing the streamwise direction and the normal direction is the so-called ‘osculating plane’ in which three-dimensional flow behaves locally like an axisymmetric flow [22]. A direct application of this system in DFD is difficult because the normal direction is generally not the wall normal.

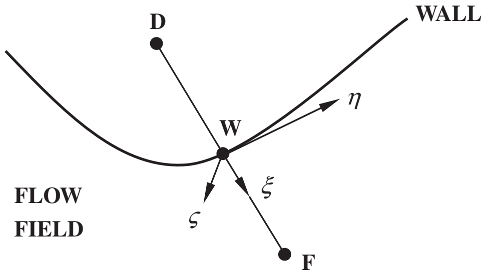

Figure 4. Local system for the evaluation of flow variables at an exterior-dependent point. ξ: wall normal direction; η: surface streamwise direction; ς: direction perpendicular to the ξ–η plane; **D**: exterior-dependent point; **F**: fictitious point; **W**: intersection point.

To remove this difficulty, we introduce a modified local system, used by Dadone and Grossman in the ghost-cell method [12], as shown in Figure 4. This local system is composed of the wall normal direction ξ, the surface streamwise direction η, and the direction ς perpendicular to both ξ and η. Moreover, the surface streamwise direction is approximated by the projection of the streamwise direction at fictitious point to the body surface. With such a local coordinate system, the local DFD can be implemented easily in three dimensions.

For inviscid flows, u_d, v_d, and w_d, the Cartesian components of velocity at the exterior-dependent point D are determined in such a way that the locally total velocity $\mathbf{q}$ at the intersection point W is tangent to the wall (i.e. the non-penetration condition). Because there is no shear stress in the inviscid flows, we can assume that in the small region near the wall, the tangent components of velocity do not change in the normal direction. With the non-penetration condition and the assumption, the normal and tangent components of the velocity at an exterior-dependent point can be obtained by

$$q_d^{\xi} = -q_f^{\xi} \tag{12}$$

$$q_d^{\eta} = q_w^{\eta} = q_f^{\eta} \tag{13}$$

$$q_d^{\varsigma} = q_w^{\varsigma} = q_f^{\varsigma} = 0 \tag{14}$$

In (12)–(14), q_d^{ξ} represents the normal component of the velocity at the exterior-dependent point D, and the other variables, such as q_d^{η} and q_f^{η}, have a similar meaning as q_d^{ξ}. With (12)–(14), the Cartesian components of the velocity at an exterior-dependent point can be computed by

$$u_d = q_d^{\xi}\xi_x + q_d^{\eta}\eta_x \tag{15}$$

$$v_d = q_d^{\xi}\xi_y + q_d^{\eta}\eta_y \tag{16}$$

$$w_d = q_d^{\xi}\xi_z + q_d^{\eta}\eta_z \tag{17}$$

where ξ_x, ξ_y, and ξ_z are the Cartesian components of the direction vector ξ, η_x, η_y and η_z the Cartesian components of direction vector η.

The pressure at the exterior-dependent point, p_d, is determined from the simplified normal momentum equation in the local $\xi-\eta$ plane, as suggested in [12] for steady flows and here extended to unsteady flows:

$$\left(\frac{\partial p}{\partial \xi}\right)_w = -\left(\frac{\partial q^{\xi}}{\partial t}\right)_w + \rho_w \frac{(q_w^{\eta})^2}{R_w^{\eta}} \tag{18}$$

In the above equation, R_w^{η} is the radius at the intersection point W of the curvature of the line that is formed by the intersection of the wall surface with the plane $\xi-\eta$. For steady flow computations, the temporal derivative $(\partial q^{\xi}/\partial t)_w$ is equal to zero. ρ_w, the density at W, is approximated by the following interpolation:

$$\rho_w = \frac{\Delta\xi_{wd}\rho_f + \Delta\xi_{fw}\rho_d}{\Delta\xi_{fd}} \tag{19}$$

where $\Delta\xi_{wd}$ represents the distance between W and D, $\Delta\xi_{fw}$ and $\Delta\xi_{fd}$ have a similar meaning. By discretizing the partial derivative in the LHS of (18) and using (13) and (19), we obtain

$$\frac{p_f - p_d}{\Delta\xi_{fd}} = \frac{(\Delta\xi_{wd}\rho_f + \Delta\xi_{fw}\rho_d)(q_f^{\eta})^2}{\Delta\xi_{fd} R_w^{\eta}} \tag{20}$$

With the assumption of an adiabatic wall $(\partial T/\partial\xi)_w = 0$, i.e. $[\partial(p/\rho)/\partial\xi]_w = 0$, we have

$$\frac{p_d}{\rho_d} = \frac{p_f}{\rho_f} \tag{21}$$

Substituting (21) into (20), p_d and ρ_d are given by

$$p_d = Bp_f, \quad \rho_d = B\rho_f \tag{22}$$

with

$$B = \frac{R_w^{\eta} p_f - \rho_f \Delta\xi_{wd}(q_f^{\eta})^2}{R_w^{\eta} p_f + \rho_f \Delta\xi_{fw}(q_f^{\eta})^2} \tag{23}$$

There exist some special exterior-dependent points at which the flow variables may be multi valued as outlined in [11, 12]. These points are associated with the thin body, the width of which is smaller than two or one grid interval. The technique for the treatment of these multi-valued points has been first suggested by Dadone and Grossman [12] and can also be considered an extension of that used in two-dimensional flow conditions [5]. The difference is that the body may be thin in several directions in three-dimensional conditions.

The first case is that the flow variables at an exterior-dependent point inside the solid body may have several values corresponding to the computed nodes locating on different sides of the body to implement the wall boundary conditions on the different sides. This case happened usually when the body width is smaller than two grid intervals. The second case is that there exist some exterior-dependent points inside the solution domain. When the body width is smaller than one grid interval, a computed node near the wall may also be the exterior-dependent point of another computed node on the opposite side of the wall to reflect the corresponding boundary effect.

Conversely, the latter computed point is also the exterior-dependent point of the former. Thus, each flow variable at such an interior point has two values: the real value is obtained by solving governing equations, and the other is obtained by the extrapolation from the flow field to reflect the opposite boundary effect.

In the present work, the values of flow variables at a fictitious point are obtained by the linear interpolation over a tetrahedron and the basis function is the piecewise linear function. Evaluating flow variables at fictitious points in this way is consistent with spatial discretization of the governing equations. However, as we have discussed at the beginning of this section, in the DFD computations, the mirroring fictitious point may locate in a wall-intersected tetrahedron. Linear extrapolation from one of the neighboring interior tetrahedrons of this wall-intersected tetrahedron is not consistent with spatial discretization of the governing equations, and the resultant errors will be larger than those with spatial approximation for the governing equations. Higher-order extrapolation may improve the accuracy but it is more complicated. Therefore, in such a situation, we have to construct a local tetrahedron only for the evaluation of normal velocity at the mirroring fictitious point, and choose another fictitious point (not the mirror image) for the evaluation of density, tangential velocity, and pressure at the corresponding exterior-dependent point. Alternatively, we may use RBFs [15–17, 23] as interpolants to evaluate flow variables at fictitious points by using a local cloud of scattered nodes surrounding the fictitious points.

6. MESH REFINEMENT

In the present work, tetrahedral meshes are obtained by refining the hexahedral cells of Cartesian meshes. Cartesian meshes require that any grid clustering near the body must be maintained to the farfield and the direct application of Cartesian meshes will lead a very large node number in the calculation of flows around a body. In order to address this problem, we generate a coarse Cartesian mesh at the beginning, and then an initial uniform tetrahedral mesh is constructed by refining hexahedral cells of this coarse Cartesian mesh. The final tetrahedral mesh used in calculations is obtained by refining the initial tetrahedral mesh gradually from the body surface to the farfield.

To refine the initial mesh, new nodes are placed within the volume of the original tetrahedral cell. The child cells resulting from such a division should be topologically similar to the parent cell, and increase in grid stretching and skewness compared with the parent cell should be avoided as much as possible. In this work, we adopt the adaptive algorithm for tetrahedral grids proposed by Vijayan and Kallinderis in [24]. Slight improvements have been made in eliminating the hanging nodes that appear on the edges of the interface between the divided and undivided cells. We flag the tetrahedrons for refinement in terms of their distances to the body surface, and therefore the refined mesh is geometrically adaptive.

In the mesh refinement, new nodes are introduced in the middle of edges of the flagged tetrahedrons. Each edge is divided into two edges and each face into four faces after introducing three edges in the interior of the divided face, as shown in Figure 5. This results in four corner child cells. Then, the interior octahedron is divided into four tetrahedrons by the shortest diagonal of the octahedron, which results in the formation of cells with the lowest aspect ratio.

After the division of the flagged cells, the resulting grid contains a number of cells that are left with hanging mid-edge nodes on some of their six edges due to the refinement of the neighboring cells. These interface cells constitute the border between the divided and the undivided cells. The interface cells can be eliminated by the following two ways.

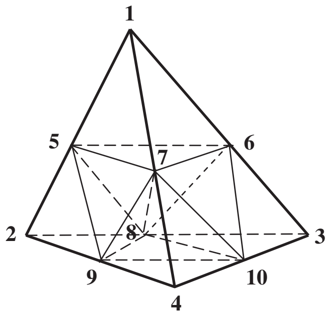

Figure 5. Division of a tetrahedron into eight sub-cells.

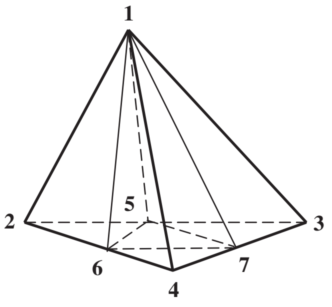

Figure 6. Division of a tetrahedron with all three hanging nodes on the same face.

6.1. *Directional division*

Two interface configurations appear frequently. Those two cases are treated separately with a simple technique. In the case where all of the hanging nodes appear on the edges of the same face, the interface cell is directionally divided into four children as shown in Figure 6. In the case of a hanging node appearing on only one of the six edges, the interface cell is henceforth divided into two children as shown in Figure 7. Directional cell division results in a significant reduction in the number of interface cells.

6.2. *Centroidal node division*

By a simple method, all the other different interface configurations can be treated in a general way. A new node is inserted at the center of the interface cell. Then, all of the four corner nodes are connected to the added node and the cell is now divided into four sub-cells as illustrated in Figure 8. The four sub-cells, on the edges of which there are still some hanging nodes, have three configurations. There are three, one, or two hanging nodes on the edges of the same face, respectively. The first two configurations are the same as those in the case of directional division shown in Figures 6 and 7, and can be divided in the same way. The sub-cell of the last configuration,

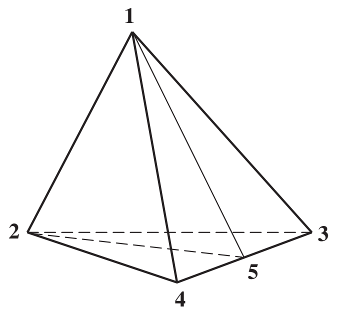

Figure 7. Division of a tetrahedron with only one hanging node.

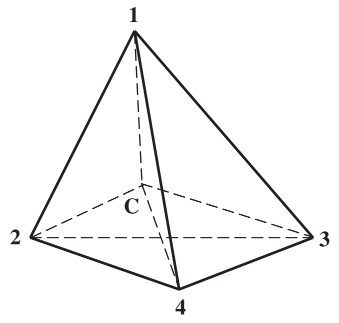

Figure 8. Division of a tetrahedron by the added centroidal node.

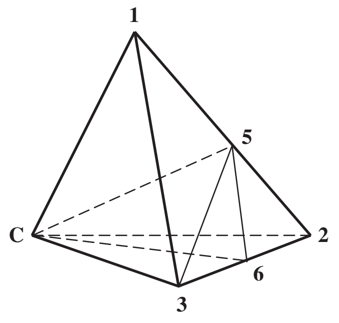

Figure 9. Division of a sub-cell.

which has two hanging nodes on the same face, can be divided into three sub-sub-cells as illustrated in Figure 9. Whether 1 and 6 or 3 and 5 are connected depends on the distances between them. The two points with shorter distance are connected.

7. NUMERICAL EXPERIMENT

Here, we present some numerical results for an ONERA M6 wing and display the comparison of the present results with experimental data and numerical results in the literature to provide a validation to the local DFD method described in this paper for the three-dimensional compressible Euler equations. This configuration has been widely used as a benchmark case for evaluating the accuracy of Euler solvers.

This wing has a leading-edge sweep angle of 30°, an aspect ratio of 3.8, and a taper ratio of 0.562. The airfoil section of the wing is the ONERA 'D' airfoil, which is a symmetrical section with 10% maximum thickness-to-chord ratio. The test case we took is the transonic flow over the wing configuration delimited by a symmetry plane, at a Mach number of free stream $M_\infty = 0.84$ and an angle of attack $\alpha = 3.06°$. In Reference [25], Schmitt and Charpin have given the experimental data (Reynolds number, $Re = 1.814\times 10^7$). Under the specified flow conditions, Euler results can be compared with Navier–Stokes results or experimental results.

The computational domain is bounded by a rectangle box with boundaries at $-5\leqslant x\leqslant 7$, $-6\leqslant y\leqslant 6$, $0\leqslant z\leqslant 6$, where x is the streamwise direction, z is the spanwise direction, and y is the direction perpendicular to x–z plane (the wing planform). The characteristic length, namely the semispan, is 1.0. The geometrically adaptive mesh has 157 304 computational nodes. The mesh size near the wing surface is $\frac{1}{80}$.

The lift and drag coefficients obtained by the current local DFD Euler solver are $C_\mathrm{L} = 0.291$, $C_\mathrm{D} = 0.014$. These values agree well with the Euler results reported by Mavriplis in [26], where the values $C_\mathrm{L} = 0.290$, $C_\mathrm{D} = 0.013$ are obtained on an adaptive unstructured mesh with 173 412 nodes. Figure 10 depicts the computed pressure contours on the wing upper surface, and the λ-shock pattern has been well captured.

In Figure 11, the computed pressure coefficients on the wing surface at six spanwise sections are presented and compared with other Euler results [24], Navier–Stokes results [27], and the experimental data [25]. Overall, the present results are close to the experimental data and reference numerical results. There is a slight difference of suck peaks between the present DFD results and the

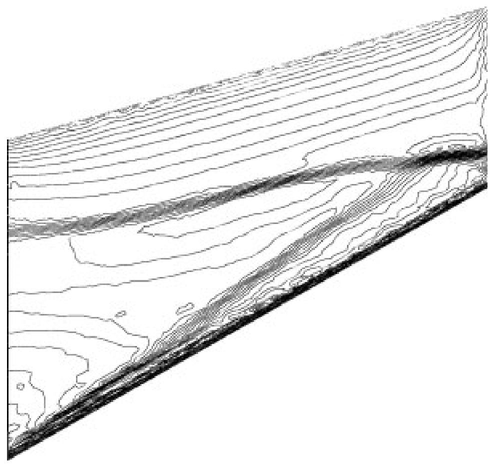

Figure 10. Computed pressure contours over the M6 wing upper surface.

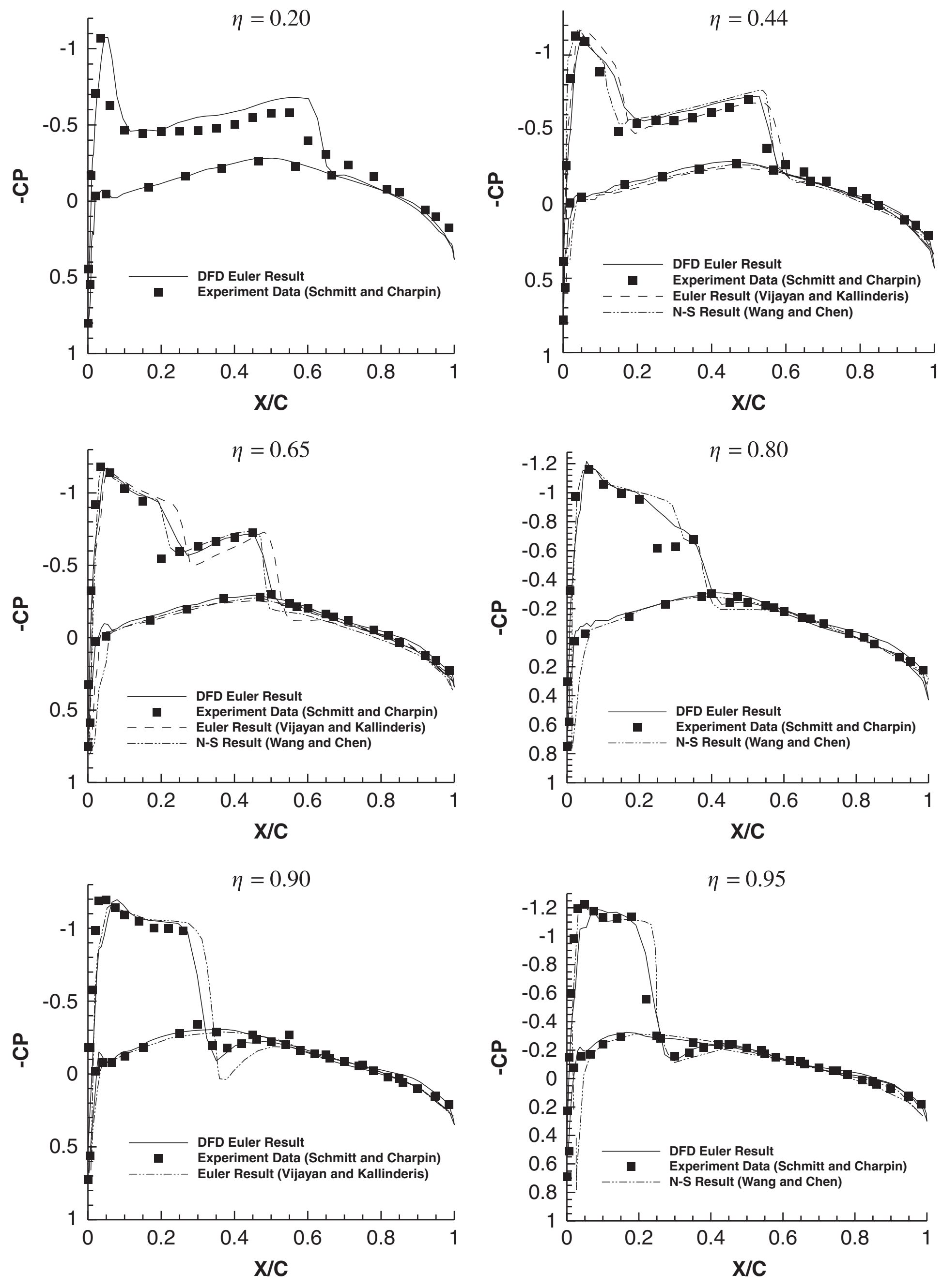

Figure 11. Comparison of pressure coefficients on the M6 wing surface.

experimental data. It may be attributed to two facts. One is that our mesh is not fine enough in the region near the leading edge. The mesh we used is obtained by refining a uniform Cartesian mesh gradually from the wing surface. Another fact is that the present governing equations are Euler equations and the viscosity effect has not been considered in the numerical simulation. By checking Figure 11 carefully, we can find that the pressure peak difference also exists in the referenced Euler results on the solution-adaptive tetrahedral grid, but disappears in the Navier–Stokes results. On the top surface of 44% spanwise section, the Navier–Stokes result has a better agreement with the experimental data, but both the Navier–Stokes solver and DFD Euler solver over-predict the expansion region. On the top surface of 65% spanwise section, the positions of the foreshock predicted by all solvers are later the experimental result. On the top surface of 80% spanwise section, both solvers do not capture the foreshock well. These differences of shock location and strength between numerical results and experimental data, which can also be frequently found in other papers, should mainly be attributed to the interaction of shock wave with turbulent boundary layer. The discussion about them exceeds the range of the present work.

8. SUMMARY

In this paper, we extend the local DFD method developed in [5] to the simulation of three-dimensional compressible flows governed by Euler equations in conservative form. According to [12], the concept of the so-called ‘modified osculating plane’ is adopted, with which the local DFD can be easily implemented in three dimensions. Geometrically adaptive tetrahedral mesh was employed for three-dimensional calculations to avoid grid clustering near the body to be maintained to the farfield. Since the boundary can be superimposed upon computational meshes, this method has all the advantages of non-boundary-conforming methods.

The local DFD method for three-dimensional compressible inviscid flows was validated by its application to simulate transonic flows over the ONERA M6 wing. The computed surface pressure distributions at different spanwise locations compare favorably well with the published experimental data and numerical results using body-fitted grid. This demonstrates the ability of the DFD method to simulate practical three-dimensional flows.

ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation of China under Grant number 10772083.

REFERENCES

1. Shu C, Fan LF. A new discretization method and its application to solve incompressible Navier–Stokes equations. *Computational Mechanics* 2001; **27**:292–301.
2. Shu C, Wu YL. Domain-free discretization method for doubly connected domain and its application to simulate natural convection in eccentric annuli. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**: 1827–1841.
3. Wu YL, Shu C, Qiu J, Tani J. Implementation of multi-grid approach in domain-free discretization method to speed up convergence. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:2425–2438.
4. Shu C, Wu YL. Adaptive mesh refinement-enhanced local DFD method and its application to solve Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 2006; **51**:897–912.
5. Zhou CH, Shu C, Wu YZ. Extension of domain-free discretization method to simulate compressible flows over fixed and moving bodies. *International Journal for Numerical Methods in Fluids* 2007; **53**:175–199.

6. Quirk JJ. An alternative to unstructured grids for computing gas dynamics flows around arbitrary complex two-dimensional bodies. *Computers and Fluids* 1994; **23**:125–142.
7. Colella P, Graves DT, Keen BJ, Modiano D. A Cartesian grid embedded boundary method for hyperbolic conservation laws. *Journal of Computational Physics* 2006; **211**:347–366.
8. Hu XY, Khoo BC, Adams NA, Huang FL. A conservative interface method for compressible flows. *Journal of Computational Physics* 2006; **219**:553–578.
9. Popinet S. Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries. *Journal of Computational Physics* 2003; **190**:572–600.
10. Ye T, Mittal R, Udaykumar HS, Shyy W. An accurate Cartesian method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics* 1999; **156**:209–240.
11. Dadone A, Grossman B. Ghost-cell method for inviscid two-dimensional flows on Cartesian grids. *AIAA Journal* 2004; **42**:2499–2507.
12. Dadone A, Grossman B. Ghost-cell method for analysis of inviscid three-dimensional flows on Cartesian-grids. *Computers and Fluids* 2007; **36**:1513–1528.
13. Liu W, Jun S, Zhang Y. Reproducing kernel particle methods. *International Journal for Numerical Methods in Fluids* 1995; **20**:1081–1106.
14. Ding H, Shu C, Yeo KS, Xu D. Development of least square-based two-dimensional finite difference schemes and their application to simulate natural convection in a cavity. *Computers and Fluids* 2004; **33**:137–154.
15. Shu C, Ding H, Yeo KS. Computation of incompressible Navier–Stokes equations by local RBF-based differential quadrature method. *CMES—Computer Modeling in Engineering and Sciences* 2005; **7**:195–205.
16. Chen W. *New RBF Collocation Schemes and Kernel RBFs with Applications*. Lecture Notes in Computational Science and Engineering, vol. 26. Springer: Berlin, 2002; 75–86.
17. Shu C, Ding H, You KS. Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:941–954.
18. Serrin J. Mathematical principles of classical fluid mechanics. In *Handbuch Der Physik*, Flugge S, Truesdell C (eds), vol. VIII. Springer: Berlin, 1959; 125–262.
19. Mavriplis DJ, Jameson A. Multigrid solution of the Navier–Stokes equations on triangular meshes. *AIAA Journal* 1990; **28**(8):1415–1425.
20. Steger J, Warming RF. Flux splitting for the inviscid gas dynamic with applications to finite-difference methods. *Journal of Computational Physics* 1983; **40**:263–293.
21. Jameson A, Baker TJ, Weatherill NP. Calculations of transonic flows over a complete aircraft. *AIAA Paper 86-0103*, 1986.
22. Frohn A. An analytic characteristic method for steady three-dimensional flows. *Journal of Fluid Mechanics* 1974; **63**(part I):81–96.
23. Chen W, Tanaka M. A study on time schemes for DRBEM analysis of scalar impact wave. *Computational Mechanics* 2002; **28**:331–338.
24. Vijayan P, Kallinderis Y. A 3D-finite volume scheme for the Euler equations on adaptive tetrahedral grids. *Journal of Computational Physics* 1994; **113**:249–267.
25. Schmitt V, Charpin F. Pressure distributions on the ONERA M6-wing at transonic Mach numbers, experiment data base for computer program assessment. *AGARD AR-138*, 1979.
26. Mavriplis DJ. Three-dimensional unstructured multigrid for the Euler equations. *AIAA Journal* 1992; **30**(7): 1753–1761.
27. Wang ZJ, Chen RF. Anisotropic Cartesian grid method for viscous turbulent flow. *AIAA-2000-0395*, 2000.