INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2007; **53**:175–199
Published online 19 June 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1254

Extension of domain-free discretization method to simulate compressible flows over fixed and moving bodies

C. H. Zhou[1,2], C. Shu[2,*,†] and Y. Z. Wu[1]

[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

This paper is the first endeavour to present the local domain-free discretization (DFD) method for the solution of compressible Navier–Stokes/Euler equations in conservative form. The discretization strategy of DFD is that for any complex geometry, there is no need to introduce coordinate transformation and 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 to impose the wall boundary condition by the approximate form of solution near the boundary. Some points inside the solution domain are constructed for the approximate form of solution, and the flow variables at constructed points are evaluated by the linear interpolation on triangles. The numerical schemes used in DFD are the finite element Galerkin method for spatial discretization and the dual-time scheme for temporal discretization. Some numerical results of compressible flows over fixed and moving bodies are presented to validate the local DFD method. Copyright © 2006 John Wiley & Sons, Ltd.

Received 16 December 2005; Revised 12 April 2006; Accepted 14 April 2006

KEY WORDS: domain-free discretization; boundary condition; moving bodies; compressible flows

1. INTRODUCTION

Recently, Shu *et al*. [1, 2] proposed a domain-free discretization (DFD) method to solve partial differential equations (PDEs) on irregular domains. 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 is only based on the given PDE. Then in the second step, the expression of the general solution is substituted into the boundary conditions to determine the unknown coefficients in the general solution. Clearly, the first step does not involve the solution domain. The solution domain (geometry of the problem) is

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

only involved in the second step when the boundary condition is implemented. So the analytical method can be well applied to both regular and irregular domain problems.

In contrast, the conventional numerical method solves the PDE by directly coupling it with the boundary condition. In other words, the numerical solution is obtained in just one step. In this step, the PDE is discretized on the solution domain with proper implementation of the boundary condition. We can see clearly that the discretization of the PDE in a numerical method is problem dependent. Due to this feature, some numerical methods such as finite difference schemes can only be applied to regular domain problems. When they are applied to solve irregular domain problems, the coordinate transformation is a must. In general, the process of coordinate transformation is very complicated, and 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 process of analytical method.

In the DFD, the implementation of the boundary condition and discretization of the governing equation are treated separately. The selected point for numerical discretization of PDE is inside the solution domain, but the discrete form of the PDE at the selected point may involve some points outside the solution domain, which serve as the role to implement the boundary condition. The key process in the DFD method is how to evaluate the functional values at the points outside the solution domain. This work can be done from the hint of the analytical method. As we know, for analytical methods, once the smooth solution of a PDE is obtained, it exactly satisfies the PDE not only at the interior points, but also at the exterior points outside the solution domain. So we can simply substitute the coordinates of exterior points into the closed-form solution to get the functional value. Although the hint is very useful, it is not applicable in the numerical computation. This is because for the case to do numerical computation, the closed-form solution is usually unknown. Otherwise, we do not need the numerical computation. On the other hand, although the closed-form solution in the whole solution domain is difficult to be obtained, it is still possible to get approximate form of the solution in the local region, for example, along a mesh line. The evaluation process of functional value at exterior point is termed as extrapolation. In the earlier applications of DFD [1, 2], the approximate form of solution is pursued along the whole mesh line, which only involves two boundary points. This way is very efficient for singly connected domain [1] and doubly connected domain [2] problems, but not so suitable for more complex domains since the mesh line may involve more than two boundary points. To remove this difficulty and make the method be more general, the local DFD was presented. In the local DFD, the Cartesian grid is used, and the low-order schemes are adopted for numerical discretization and approximate form of the solution. Since the Cartesian grid is usually used, the local DFD method becomes a kind of Cartesian grid solvers. Therefore, the mesh generation in the local DFD method is very simple and straightforward. Since one can use a mesh that is not fitted to the wall boundary in the DFD method, the mesh can stay fixed while the body is moving. In this process, only the status of mesh points, either inside the solution domain or outside the solution domain, is changed. So another advantage of the local DFD method is that we do not need to generate a new mesh or deform the previous mesh at each time step for a moving boundary problem. Compared with the mesh-deforming or mesh-moving methods [3, 4], there is no special difficulty for the local DFD method to simulate the flows over multi-moving-bodies.

On the other hand, we notice that there are a variety of Cartesian grid methods in the literature, which have different features from the local DFD method. One of them is the immersed boundary method (IBM) [5, 6]. IBM solves the governing equation in the whole domain including the region inside the solid boundary. It is actually an iterative process to satisfy the non-slip boundary condition

DOI: 10.1002/fld

by distributing the restoration force, which is resulted from unsatisfying of the boundary condition, to the Navier–Stokes equations. Thus, in the IBM, the boundary condition is approximately satisfied. In contrast, the DFD method discretizes the governing equation in the solution domain, but its discrete form may involve some points outside the solution domain, which are used to consider the effect of boundary conditions. In the DFD, the boundary condition is enforced directly. To the best of our knowledge, IBM is usually limited to the numerical simulation of incompressible viscous flows.

Another popular Cartesian grid method is the Cartesian grid embedded boundary method [7–9] presented by the group of Colella for heat equation and hyperbolic conservative equations. This approach is based on the finite volume discretization on a grid obtained from intersecting the domain with a Cartesian grid, and the finite volume grid contains the actual boundary. For irregular solution domains, there exist irregular cells cut by the embedded boundary, which are involved in the computation. In our approach, the cells are triangles obtained directly from the Cartesian grid, and the actual boundary does not appear in the computational mesh and the discrete form of PDEs does not involve any irregular cells. This is the main difference between our approach and the work of Colella's group. In Colella's method, the state variables are defined at the cell centre, even when the centre is outside the solution domain. Some extrapolations are used to compute the flux at the boundary faces and the covered faces. In this sense, the two methods are similar, but their extrapolation procedure is different. Colella's method [7–9] computes the flux at a covered face or a boundary face by solving a Riemann problem. In our evaluation of the state variables at the points outside the solution domain, the velocity extrapolation is similar to the local mirroring extrapolation used by Arienti *et al*. [10], and the density and pressure extrapolation is based on the local simplified flow equations which are different from constant extrapolation of Arienti *et al*. [10]. Arienti *et al*. [10] presented a ghost-fluid Eulerian–Lagrangian method for fluid–solid interaction problems. It was originated from the ghost-fluid method (GFM).

GFM was proposed by Fedkiw *et al*. [11] and originated as an algorithm for handling multi-fluid problems. The original GFM is designed to capture the discontinuous interfaces with an Eulerian solver on each side. Within a prescribed distance of an interface that depends exclusively on the stencil of the numerical scheme, an Eulerian grid point is a real node to one solver and a ghost node to the other. In GFM, the pressure and normal velocity of the ghost fluid are set identically equal to the pressure and normal velocity of the real fluid at each point, while the entropy and tangential velocity of the ghost fluid are extrapolated in the normal direction from the real fluid at another side of the interface. In physics, the definition of state variables at a ghost node in GFM is different from extension of the solution at interior points to the exterior dependent points in DFD. For the Poisson equation on irregular domains, using GFM [12], the value of solution variable at dependent point outside the boundary (ghost node) is defined by constant, linear, and quadratic extrapolation from the solution domain, while in our approach, the local mirroring extrapolation is taken to avoid the overshoot of an extrapolated variable when the boundary is very close to an interior point. The Cartesian grid method of Morton and Mayers [13] also uses a different extrapolation procedure from the DFD method. In DFD, the mirroring extrapolation is performed along the normal direction to the boundary so that the uniqueness can be ensured in the process. In Reference [13], the extrapolation to the exterior point is along the coordinate lines. So for different interior points, the extended solution at an exterior point may have multi-values.

So far, the DFD method has been successfully applied to simulate many incompressible flows [1, 2]. Its performance for the compressible flow simulation is still not clear as compressible flows involve the sharp shock waves. This paper is the first endeavour to present the local DFD

DOI: 10.1002/fld

method for the solution of compressible Navier–Stokes/Euler equations in the conservative form. The functional values at the exterior points are updated at each time step, in order to satisfy the wall boundary condition, by the approximate form of solution along the normal direction to the boundary. The accuracy of the DFD method depends on the numerical schemes used. In this paper, the second-order finite element Galerkin method is applied for spatial discretization. To show the accuracy of the local DFD method, the two-dimensional Poisson equation on a domain with curved boundary is solved on a series of meshes. Numerical experiments do confirm that the present local DFD method has the second-order accuracy. To show the capability of the local DFD method for simulation of compressible flows, flows around fixed and moving airfoils and circular cylinders are simulated and the present results are compared with available data in the literature.

2. GOVERNING EQUATIONS

The dimensionless, two-dimensional compressible Navier–Stokes equations in the conservative form can be written as

$$\frac{\partial w}{\partial t}+\frac{\partial f_{\mathrm{c}}}{\partial x}+\frac{\partial g_{\mathrm{c}}}{\partial y}=\frac{\sqrt{\kappa}M_{\infty}}{Re_{\infty}}\left[\frac{\partial f_{\mathrm{v}}}{\partial x}+\frac{\partial g_{\mathrm{v}}}{\partial y}\right] \tag{1}$$

where w is the vector of conservative variables, f_{c} and g_{c} are the convective flux vectors, f_{v} and g_{v} are the viscous flux vectors, M_{∞} is the Mach number of the free stream, Re_{∞} is the Reynolds number of the free stream, and κ is the ratio of specific heats of fluid and taken as 1.4 for air. When $Re_{\infty}\to\infty$, the above Navier–Stokes equations reduce to the Euler equations governing the inviscid compressible flows.

w, f_{c} and g_{c} are given by

$$w=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{bmatrix},\quad f_{\mathrm{c}}=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho u(E+p)\end{bmatrix},\quad g_{\mathrm{c}}=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho v(E+p)\end{bmatrix} \tag{2}$$

where ρ represents the fluid density, u and v are the x and y 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:

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

f_{v} and g_{v} are given by

$$f_{\mathrm{v}}=\begin{bmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ u\tau_{xx}+v\tau_{xy}-q_x\end{bmatrix},\quad g_{\mathrm{v}}=\begin{bmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ u\tau_{xy}+v\tau_{yy}-q_y\end{bmatrix} \tag{4}$$

DOI: 10.1002/fld

where τ represents the stress tensor and q is the heat flux vector

$$\tau_{xx} = 2\mu u_x - \tfrac{2}{3}\mu(u_x + v_y), \quad \tau_{yy} = 2\mu v_y - \tfrac{2}{3}\mu(u_x + v_y), \quad \tau_{xy} = \mu(u_y + v_x) \tag{5}$$

$$q_x = \frac{\kappa}{\kappa - 1}\left(\frac{\mu}{Pr}\right)\frac{\partial(p/\rho)}{\partial x}, \quad q_y = \frac{\kappa}{\kappa - 1}\left(\frac{\mu}{Pr}\right)\frac{\partial(p/\rho)}{\partial y} \tag{6}$$

In (5) and (6), Pr is the Prandtl number taken as 0.72 for air, μ is the molecular viscosity given by the Sutherland law.

3. LOCAL DFD DISCRETIZATION AND ARTIFICIAL DISSIPATION

As described in the introduction, in the local DFD method, the wall boundary can be superimposed upon the computational mesh. We suppose that $\Omega \subset R^2$ is a connected open set containing a body ω, as shown in Figure 1. For the flow around ω, the solution domain can be represented by $\Omega\backslash\omega$. We denote the boundary of Ω and ω by Γ and γ, respectively. With h a space discretization step, a triangulation T_h of Ω is introduced, as shown in Figure 2.

As indicated in the introduction, 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 similar to the Galerkin finite element approach proposed by Mavriplis and Jameson [14]. The procedure begins by storing flow variables at the vertices of triangles, and the flow variables are approximated by piecewise linear functions. Let F_C denote the convective flux tensor. Its Cartesian components are f_c and g_c. Similarly, F_V denotes the viscous flux tensor, and its Cartesian components are f_v and g_v. The Navier–Stokes equation

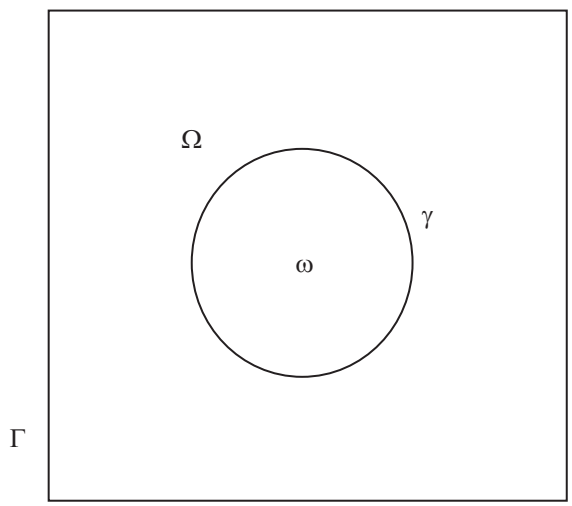

Figure 1. Configuration of solution domain.

DOI: 10.1002/fld

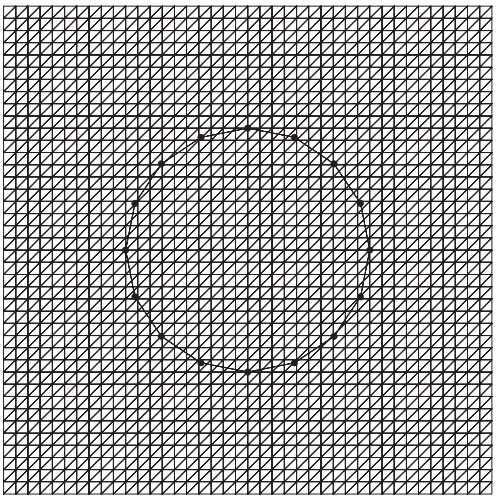

Figure 2. A triangulation T_h of Ω.

can then be rewritten in the following form:

$$\frac{\partial w}{\partial t} + \nabla \cdot F_{\mathrm{C}} = \frac{\sqrt{\kappa} M_\infty}{Re_\infty} \nabla \cdot F_{\mathrm{V}} \tag{7}$$

The Galerkin formulation of (7) is

$$\frac{\partial}{\partial t} \iint_\Omega \phi w \, \mathrm{d}x \, \mathrm{d}y + \iint_\Omega \phi \nabla \cdot F_{\mathrm{C}} \, \mathrm{d}x \, \mathrm{d}y = \frac{\sqrt{\kappa} M_\infty}{Re_\infty} \iint_\Omega \phi \nabla \cdot F_{\mathrm{V}} \, \mathrm{d}x \, \mathrm{d}y \tag{8}$$

where ϕ is a test function. Integrating the flux integrals in (8) by parts gives

$$\begin{aligned} \frac{\partial}{\partial t} \iint_\Omega \phi w \, \mathrm{d}x \, \mathrm{d}y = & \iint_\Omega F_{\mathrm{C}} \cdot \nabla \phi \, \mathrm{d}x \, \mathrm{d}y - \frac{\sqrt{\kappa} M_\infty}{Re_\infty} \iint_\Omega F_{\mathrm{V}} \cdot \nabla \phi \, \mathrm{d}x \, \mathrm{d}y \\ & - \int_\Gamma \mathbf{n} \cdot F_{\mathrm{C}} \phi \, \mathrm{d}\sigma + \frac{\sqrt{\kappa} M_\infty}{Re_\infty} \int_\Gamma \mathbf{n} \cdot F_{\mathrm{V}} \phi \, \mathrm{d}\sigma \end{aligned} \tag{9}$$

where $\mathbf{n}$ is the outward normal unit vector at Γ. In order to evaluate the flux at each vertex P, ϕ is taken as a piecewise linear function which is equal to 1 at node P and vanishes at all other vertices. Therefore, the integrals in the above equation are non-zero only over the triangles that contain the vertex P. As a result, the influence domain of P can be defined as shown in Figure 3.

The convective fluxes F_{C} can be taken as piecewise linear functions in space, and the viscous fluxes F_{V} can be taken as piecewise constants over each triangle since they are formed from the first-order derivatives of the flow variables. Evaluating the flux integrals with these assumptions and employing the concept of a lumped mass matrix while integrating the term on the left-hand side (LHS) of (9), one obtains

$$\Omega_P \frac{\partial w_P}{\partial t} = \sum_{e=1}^{n} \frac{F_{\mathrm{C}}^A + F_{\mathrm{C}}^B}{2} \cdot \Delta L_{AB} - \frac{\sqrt{\kappa} M_\infty}{Re_\infty} \sum_{e=1}^{n} \frac{3}{2} F_{\mathrm{V}}^e \cdot \Delta L_{AB} \tag{10}$$

DOI: 10.1002/fld

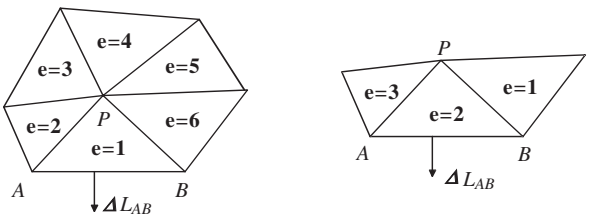

Figure 3. Influence domain of node P (left: for an interior node and right: for a node on the boundary Γ).

In the above formulation, the summations are over all the triangles in the influence domain of P. Ω_P represents the surface area of the domain. ΔL_{AB} represents the directed (outward normal) edge length of the face of each triangle e on the outer boundary of the influence domain; F_{C}^{A} and F_{C}^{B} are the convective fluxes at two vertices of the edge, and F_{V}^{e} is the viscous flux in triangle e.

For the numerical simulation of the flow around the body ω, the boundary Γ can be regarded as the outer boundary of the solution domain. In this work, the conventional boundary conditions at Γ are imposed and the Steger–Warming flux splitting scheme [15] is applied at the in- and outflow boundaries.

The domain Ω is an auxiliary domain and the boundary Γ can be arbitrary if it is far enough from ω. Here we should indicate that there is no imposition of wall boundary conditions in the spatial discretization and the discrete form of the Navier–Stokes equations is irrelevant of the solution domain according to the concept of DFD. This is the main difference from the original approach in Reference [14]. The implementation of wall boundary conditions will be discussed in Section 5.

The artificial dissipation operator proposed by Jameson *et al.* [16] is adopted to prevent oscillations near shocks and damp high-frequency errors. For completeness, a brief description on the construction of this operator is given below. It is a blend of undivided Laplacian and biharmonic operators in the flow field. At node i, the undivided Laplacian of w can be approximated by the summation of the differences of w along all edges meeting at i

$$(\nabla^2 w)_i = \sum_{k=1}^{n} (w_k - w_i) \tag{11}$$

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

$$D(w_i) = \sum_{k=1}^{n} \alpha_k \{\varepsilon_{2k}(w_i - w_k) - \varepsilon_{4k}(\nabla^2 w_i - \nabla^2 w_k)\} \tag{12}$$

In the above equation, ε_2 and ε_4 are adaptive coefficients designed to switch on sufficient dissipation when it is needed, and α is a factor proportional to the maximum eigenvalue of the Euler equations.

DOI: 10.1002/fld

4. TIME MARCHING

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

$$\Omega_i \frac{\mathrm{d}w_i}{\mathrm{d}t} + R(w_i) = 0, \quad i = 1, 2, 3, \ldots, N \tag{13}$$

where N is the number of the mesh points and

$$R(w) = Q(w) - D(w) \tag{14}$$

In the above equation, $Q(w)$ represents the discrete approximation of the convective fluxes, and $D(w)$ represents the dissipative term including the discrete approximation of the viscous fluxes and the artificial dissipation.

Following the work of Jameson [17], an implicit equation system can be obtained by approximating Equation (13) at the time level $(n+1)$

$$\Omega_i \frac{\mathrm{d}w_i^{n+1}}{\mathrm{d}t} + R(w_i^{n+1}) = 0 \tag{15}$$

When the second-order time discretization is used, Equation (15) becomes

$$\Omega_i \frac{3w_i^{n+1} - 4w_i^n + w_i^{n-1}}{2\Delta t} + R(w_i^{n+1}) = \overline{R}(w_i^{n+1}) = 0 \tag{16}$$

Adding a derivative with respect to a fictitious pseudo time τ to Equation (16) gives

$$\Omega_i \frac{\mathrm{d}w_i^{n+1}}{\mathrm{d}\tau} + \overline{R}(w_i^{n+1}) = 0 \tag{17}$$

The solution of equation system (17) can be obtained by marching in pseudo time to a steady state. In this work, the five-stage, hybrid, time-stepping scheme proposed by Mavriplis and Jameson [14] is adopted to march in the pseudo time. For each point, the allowable pseudo-time-step is chosen by

$$\Delta\tau_i = \min\left(\Delta\overline{\tau}_i, \frac{2\Delta t}{3}\right) \tag{18}$$

where $\Delta\overline{\tau}_i$ is the local pseudo-time-step considering the stability limitation due to both the convective and diffusive characters of the Navier–Stokes equations.

5. IMPLEMENTATION OF WALL BOUNDARY CONDITIONS

According to the spatial discretization presented in Section 3, the calculation of the conservative variables at any mesh point inside the solution domain depends on the flow variables at the points connected to this point by an edge of triangle, as shown in Figure 3. For an edge intersecting with the wall boundary, the interior end point is a computed node and the exterior end point is one of

DOI: 10.1002/fld

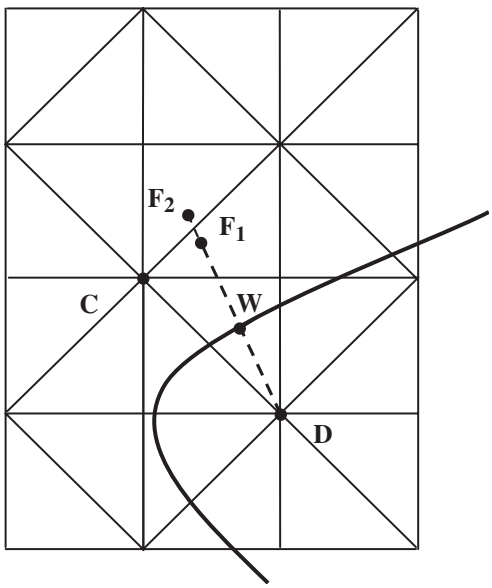

Figure 4. Interior computed point, exterior dependent point, and fictitious points.

dependent points of the interior end point. All this kind of interior computed nodes near the wall boundary has at least one dependent point outside the solution domain. An example is shown in Figure 4, where an interior computed node is denoted by C and an exterior dependent point is represented by D. The key process of the DFD method is how to evaluate the functional values at the exterior dependent points.

In this work, the values of flow variables at the exterior dependent points are extrapolated from the flow field along the normal direction to the wall boundary, or determined by the local simplified flow equations at the boundary. Since all flow variables are approximated by piecewise linear functions in the spatial discretization, the linear extrapolation will be reasonable. Therefore, some points 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, so they are termed as fictitious points.

The image point (to the wall) of a given exterior dependent point is chosen as its first fictitious point, and it can be found along the normal direction to the wall boundary, as shown in Figure 4. In this figure, the first fictitious point of the exterior dependent point D is denoted by F_1, and the intersection point of the line in the normal direction with the wall boundary is denoted by W. We suppose that this fictitious point is located in a triangle $\tau \in T_h$. If all the three vertices of τ are inside the solution domain, as shown in Figure 4, this triangle can be used to evaluate the flow variables at the fictitious point F_1 by the linear interpolation. So we denote this interpolation triangle by τ_I.

If the three vertices of τ are not all inside the solution domain, the values of the flow variables at the exterior vertex are not known. A new interpolation triangle $\tau_I \notin T_h$ should be constructed locally to evaluate the velocity components at the first fictitious point. For the example shown in

DOI: 10.1002/fld

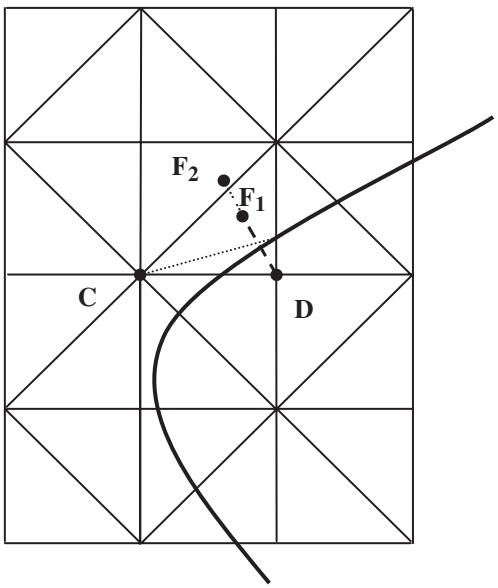

Figure 5. Local construction of the interpolation triangle $\tau_I \notin T_h$.

Figure 5, the construction can be done easily by linking one interior vertex of τ to the intersection point of the wall boundary with its opposite edge. In this case, the values of pressure, density, and tangential component of velocity (for inviscid flow computations) at the boundary vertex of τ_I are not known, so the values of these variables at the fictitious point F_1 cannot be determined. To overcome this difficulty, we construct the second fictitious point for the given exterior dependent point. The point which satisfies: (1) on the line normal to the wall boundary that passes through the exterior dependent point; (2) the three vertices of the triangle containing this point (i.e. the interpolation triangle) are all inside the field; and (3) closest to the wall boundary, is defined as the second fictitious point F_2 as shown in Figure 5.

So the flow variables at the three vertices of the interpolation triangle τ_I either take the current computed values or are determined directly from the known component values of velocity at the boundary-intersection points. By the linear interpolation over τ_I, the values of the flow variables at the fictitious point can be obtained. For example, the horizontal component of velocity at the fictitious point is computed by

$$u_f = \sum_{k=1}^{3} \phi_k(x_f, y_f) u_k \tag{19}$$

where the summation over k refers to the three vertices of the interpolation triangle τ_I.

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

Now, we discuss the evaluation of the flow variables at the exterior dependent point with reference to Figure 6.

DOI: 10.1002/fld

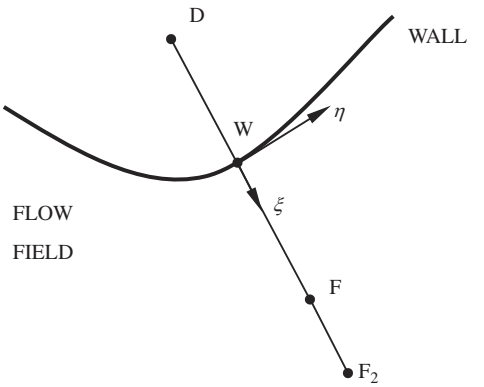

Figure 6. Evaluation of the flow variables at exterior dependent point: D: exterior dependent point, F: fictitious point, W: intersection point, F_2: additional fictitious point for approximation of the second-order derivative of q_ξ.

For inviscid flows, the Cartesian components of velocity at the exterior dependent point D, u_d and v_d, are determined in such a way that the normal velocity to the wall should be zero (i.e. no flow through the wall). Because there is no shear stress in inviscid flows, we can assume that in the small region near the wall, the tangential component of velocity does not change along the normal direction to the wall. With the wall boundary condition and the assumption, the tangential and normal components of the velocity at the dependent points can be obtained by

$$q_{\eta_d} = q_{\eta_f} = q_{\eta_w} \tag{20}$$

$$q_{\xi_d} = 2q_{\xi_w} - q_{\xi_f}, \quad q_{\xi_w} = V_{\xi_w} \tag{21}$$

In (20) and (21), ξ and η represents the normal and tangential direction to the wall boundary at W, V_{ξ_w} is the normal component of the velocity of body movement at W, q_{η_d} is the tangential component of the velocity at the exterior dependent point D. The other variables in (20) and (21), such as q_{ξ_d} and q_{η_f}, have a similar meaning as q_{η_d}. From (20) and (21), the Cartesian components of the velocity at the exterior dependent point D can be computed by

$$u_d = 2\xi_x V_{\xi_w} + u_f(\xi_y^2 - \xi_x^2) - 2v_f \xi_x \xi_y \tag{22}$$

$$v_d = 2\xi_y V_{\xi_w} + v_f(\xi_x^2 - \xi_y^2) - 2u_f \xi_x \xi_y \tag{23}$$

where ξ_x and ξ_y are the Cartesian components of the outward unit normal vector at the boundary point W.

The pressure at the exterior dependent point D, p_d, is determined from the following normal momentum equation in the local coordinates:

$$\left(\frac{\partial p}{\partial \xi}\right)_w = -\frac{\partial q_{\xi_w}}{\partial t} + \frac{\rho_w (q_{\eta_w})^2}{R_w} \tag{24}$$

DOI: 10.1002/fld

In the above equation, R_w is the local radius of the wall boundary at the intersection point W, and the density at W is approximated by the following interpolation:

$$\rho_w = \frac{\Delta\xi_{wd}\rho_f + \Delta\xi_{fw}\rho_d}{\Delta\xi_{fd}} \tag{25}$$

where $\Delta\xi_{wd}$ represents the distance in the normal direction between W and D, $\Delta\xi_{fw}$ and $\Delta\xi_{fd}$ have a similar meaning as $\Delta\xi_{wd}$. By discretizing the partial derivative on the left-hand side of Equation (24) and using Equations (20) and (21), we obtain

$$\frac{p_f - p_d}{\Delta\xi_{fd}} = -\frac{\partial V_{\xi_w}}{\partial t} + \frac{(\Delta\xi_{wd}\rho_f + \Delta\xi_{fw}\rho_d)(q_{\eta_f})^2}{R_w\Delta\xi_{fd}} \tag{26}$$

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

$$\frac{p_d}{\rho_d} = \frac{p_f}{\rho_f} \tag{27}$$

Substituting Equation (27) into Equation (26), p_d and ρ_d are given by

$$p_d = Bp_f, \quad \rho_d = B\rho_f \tag{28}$$

with

$$B = \frac{R_w p_f + R_w\Delta\xi_{fd}\dfrac{\partial V_{\xi_w}}{\partial t} - \rho_f\Delta\xi_{wd}(q_{\eta_f})^2}{R_w p_f + \rho_f\Delta\xi_{fw}(q_{\eta_f})^2} \tag{29}$$

For viscous flows, the Cartesian components of velocity at the exterior dependent point, u_d and v_d, are determined from the non-slip boundary condition at the wall, that is, the total velocity $\boldsymbol{q}$ at the point W is equal to the velocity of body movement. So u_d and v_d can be computed by

$$u_d = 2u_w - u_f, \quad u_w = V_{x_w} \tag{30}$$

$$v_d = 2v_w - v_f, \quad v_w = V_{y_w} \tag{31}$$

The tangential and normal components of velocity at the boundary point W are

$$q_{\eta_w} = V_{\eta_w}, \quad q_{\xi_w} = V_{\xi_w} \tag{32}$$

p_d, the pressure at the exterior dependent point D, is determined from the following simplified normal momentum equation:

$$\left(\frac{\partial p}{\partial\xi}\right)_w = -\frac{\partial q_{\xi_w}}{\partial t} + \frac{\rho_w(q_{\eta_w})^2}{R_w} + \frac{\mu_w\sqrt{\kappa}M_\infty}{Re_\infty}\left(\frac{\partial^2 q_\xi}{\partial\xi^2}\right)_w \tag{33}$$

For viscous flows, an adiabatic wall is also assumed, so the viscosity coefficient at the boundary point W can be obtained by $\mu_w = \mu_f$. In (33), ρ_w is also approximated by the linear interpolation (25). To approximate the second-order partial derivative on the right-hand side (RHS) of (33), the second fictitious point should be introduced as shown in Figure 4. This can be done in the

DOI: 10.1002/fld

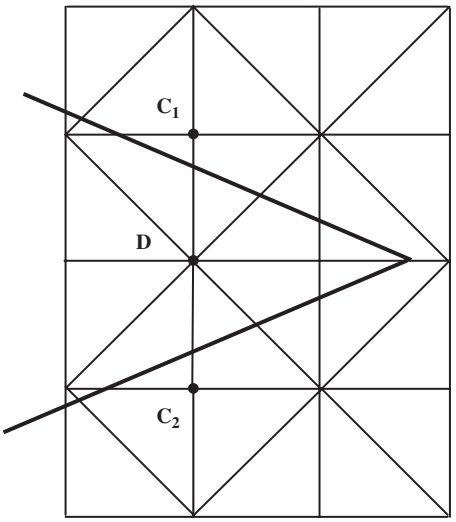

Figure 7. Case 1 of multi-value points.

similar way as we find the second fictitious point shown in Figure 5. This additional fictitious point is denoted by F_2 in Figure 6. After approximating the spatial derivatives in Equation (33) by the second-order difference schemes and using Equations (30) and (31) with the assumption of an adiabatic wall (27), we obtain

$$p_d = Cp_f, \quad \rho_d = C\rho_f \tag{34}$$

with

$$C = \frac{R_w p_f + R_w \Delta\xi_{fd}\left[\dfrac{\partial V_{\xi_w}}{\partial t} - \dfrac{\mu_f\sqrt{\kappa}M_\infty}{2Re_\infty}\left(\dfrac{q_{\xi_{f_2}} - q_{\xi_f}}{\Delta\xi_{ff_2}\Delta\xi_{f_2w}} - \dfrac{q_{\xi_f} - V_{\xi_w}}{\Delta\xi_{fw}\Delta\xi_{f_2w}}\right)\right] - \rho_f\Delta\xi_{wd}(q_{\eta_f})^2}{R_w p_f + \rho_f\Delta\xi_{fw}(q_{\eta_f})^2} \tag{35}$$

Now, we discuss some special cases where the flow variables at some points may have multi-values. These cases are associated with the thin wall boundary, the width of which is smaller than two or one grid interval.

The first case is shown in Figure 7. It is obvious that near the tailing edge of an airfoil, the exterior point D corresponds to two computed mesh points C_1 and C_2. The point C_1 is above the airfoil and C_2 is under the airfoil. So each flow variable at D has two values: one for the computation at C_1 is extrapolated along the normal direction to the upper wall from the flow field above the airfoil, and another for the computation at C_2 is extrapolated along the normal direction to the lower wall from the flow field under the airfoil.

The second case is that there exist some 'exterior' dependent points inside the solution domain. As shown in Figure 8, the point C_1 is, of course, a computed mesh point under the airfoil, but it is also the exterior (to the upper wall) dependent point of the computed point C_2 above the airfoil. So the point C_1 is also denoted by D_2 in Figure 8. Each flow variable at C_1 has two values:

DOI: 10.1002/fld

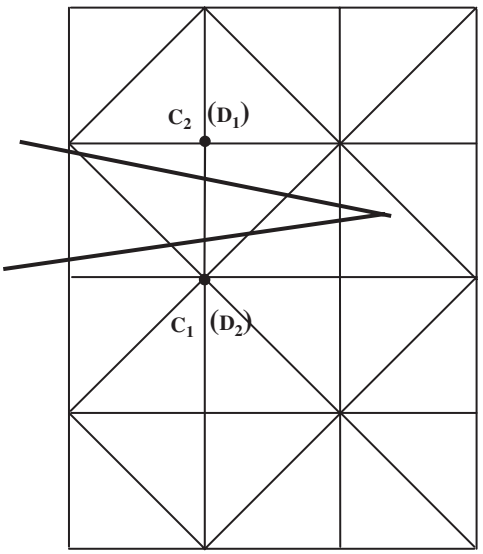

Figure 8. Case 2 of multi-value points.

the real one is computed from the governing equations, and the other is extrapolated along the normal direction to the upper wall from the flow field above the airfoil, which is used to reflect the boundary effect. Similarly, the computed point C_2 is also the exterior dependent point (also denoted by D_1) of C_1. Each flow variable at C_2 also has two values: the real one is obtained by solving the governing equations, and the other is obtained by the extrapolation from the flow field under the airfoil to consider the effect of boundary condition.

6. NUMERICAL EXPERIMENTS

Some numerical examples will be shown in this section to provide a validation to the local DFD method described above for the Navier–Stokes and the Euler equations in the conservative form.

6.1. Convergence rate of local DFD method for two-dimensional Poisson equation

To show the convergence rate of the local DFD method used in this work, the test problem of the two-dimensional Poisson equation is solved on the domain $\Omega\backslash\omega$ as shown in Figure 1

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x, y) \quad \text{in } \Omega\backslash\omega \tag{36}$$

Here Ω is the square domain $\Omega = [0, 1] \times [0, 1]$, and ω is a circle located at the centre of Ω with the radius of 0.25. For simplicity, the analytical solution u_{exact} is given to determine the source function $f(x, y)$ and to measure the numerical error. The Dirichlet boundary condition is given

DOI: 10.1002/fld

Table I. Convergence rate of local DFD method for two-dimensional Poisson equation.

	Function u_1		Function u_2	
Mesh spacing h	L^1-error	Order of accuracy	L^1-error	Order of accuracy
2.0×10^{-2}	1.18×10^{-4}	—	6.04×10^{-5}	—
1.0×10^{-2}	2.23×10^{-5}	2.34	1.46×10^{-5}	2.06
5.0×10^{-3}	6.10×10^{-6}	1.93	3.57×10^{-6}	2.03
2.5×10^{-3}	1.48×10^{-6}	2.04	9.52×10^{-7}	1.96

on the boundary, i.e.

$$u = u_{\text{exact}} \quad \text{on } \Gamma \cup \gamma \tag{37}$$

Two functions are selected as the analytical solutions of Equation (36). They are given as follows:

$$u_1 = \sin(\pi x)\sin(\pi y) \tag{38}$$

$$u_2 = \left(1-\frac{x}{2}\right)^6\left(1-\frac{y}{2}\right)^6 + 1000(1-x)^3x^3(1-y)^3y^3 + \left(1-\frac{x}{2}\right)^6 y^6 + \left(1-\frac{y}{2}\right)^6 x^6 \tag{39}$$

The functions u_1 and u_2 are taken from the work of Ding *et al.* [18] and Lyche *et al.* [19], respectively. In our computations, a uniform Cartesian mesh is generated at first, and then the triangulation is obtained by dividing one rectangle into two triangles. Table I shows the results of the numerical convergence test on a series of uniform meshes. It can be observed that in L^1 norm, the convergence rate or the accuracy of the local DFD method is about second order for this smooth problem.

6.2. Inviscid compressible flows past the fixed NACA0012 airfoil

The simulation of inviscid subsonic and transonic flows over a NACA0012 airfoil is chosen to demonstrate the accuracy of the Euler solutions obtained by the local DFD method. These two cases (subsonic and transonic) have been studied by many researchers.

To accurately simulate the flow past an airfoil, the triangle mesh is refined gradually in the region near the wall. The mesh used in the two cases is shown in Figure 9 and the number of computational points is 28 778. For comparison, the same calculations were also performed on the structured C-type mesh (70×320).

In the test case of subsonic flow, the Mach number of free stream is chosen as 0.63 and the angle of attack is 2°. The pressure coefficient distribution on the surface is given in Figure 10. The comparison of present result with the data given from the body-fitted mesh solver demonstrates good agreement. The lift and drag coefficients computed by the local DFD method are $C_L = 0.335$, $C_D = -6.4\times10^{-5}$, which compare very well with the results of the body-fitted mesh solver ($C_L = 0.336$, $C_D = 2.5\times10^{-5}$), the potential flow solver [20] ($C_L = 0.333$, $C_D = -2.0\times10^{-4}$), and the Euler solver [21] ($C_L = 0.334$, $C_D = -2.1\times10^{-5}$). Figure 11 presents the convergence history of residuals with the pseudo-time-step. The residual is defined as $\text{RES} = \|(\partial\rho/\partial t)^n\|_2 / \|(\partial\rho/\partial t)^1\|_2$, where the superscript of $\partial\rho/\partial t$ represents the time step.

DOI: 10.1002/fld

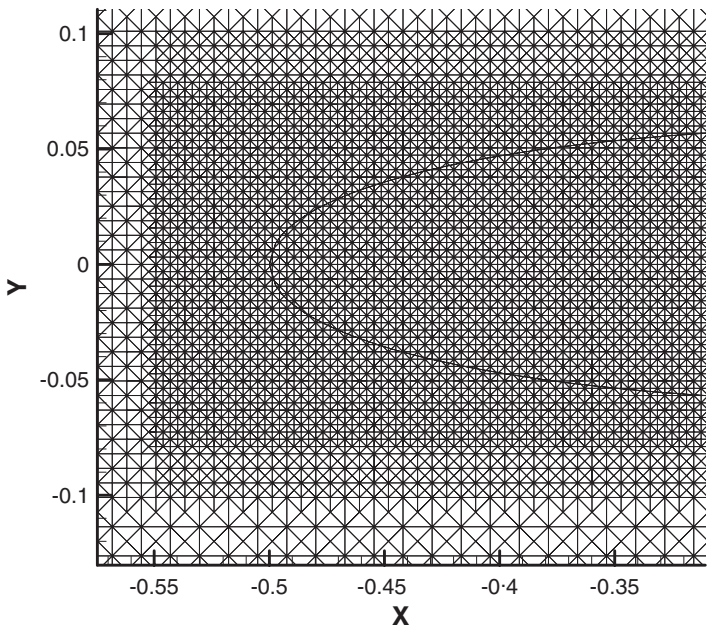

Figure 9. Partial view of the mesh used in the local DFD computations.

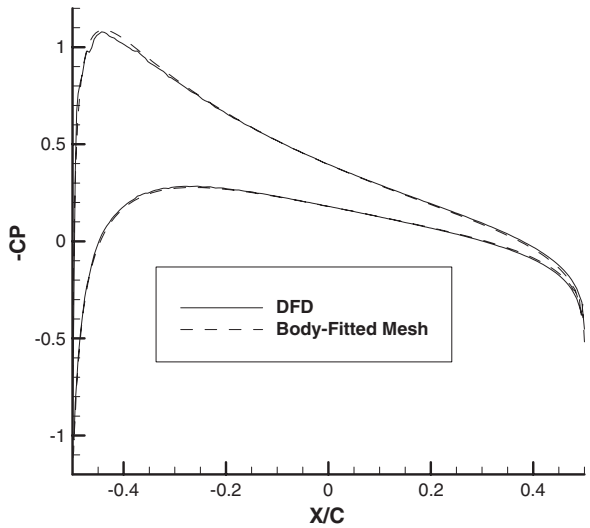

Figure 10. Pressure coefficient on the surface of NACA 0012 airfoil for $M_\infty = 0.63$, $\alpha = 2^\circ$.

In the test case of transonic flow, the Mach number of free stream is taken as 0.85 and the angle of attack is 1°. The pressure coefficient on the surface is given in Figure 12 and compared with the results obtained by the body-fitted mesh solver. The shocks captured by the local DFD method are a little bit sharper than those captured by the body-fitted mesh solver. This small difference is

DOI: 10.1002/fld

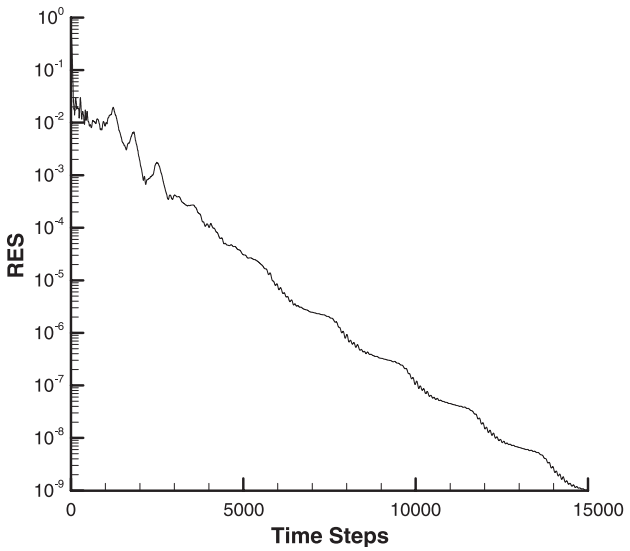

Figure 11. Convergence history of local DFD method for flow around NACA 0012 airfoil, $M_\infty = 0.63$, $\alpha = 2°$.

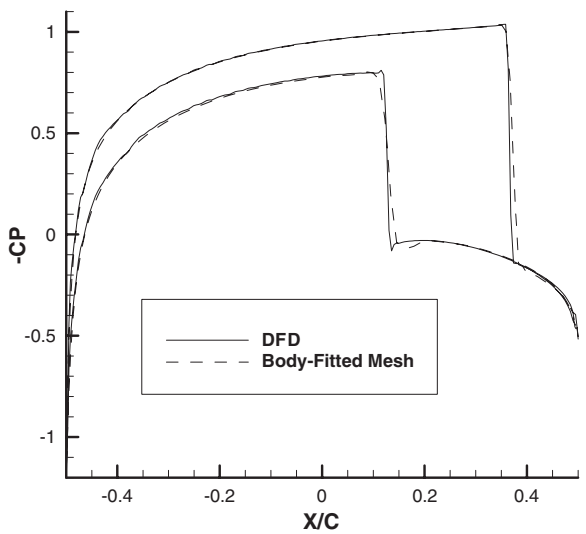

Figure 12. Pressure coefficient on the surface of NACA 0012 airfoil for $M_\infty = 0.85$, $\alpha = 1°$. DFD: $C_L = 0.373$, $C_D = 0.057$. Body-fitted mesh: $C_L = 0.380$, $C_D = 0.059$. Results of Reference [21]: $C_L = 0.377$, $C_D = 0.058$.

attributed to the geometry-adaptation employed in the mesh generation for the DFD method. The computed lift and drag coefficients by DFD and the body-fitted mesh solver are very close and also agree well with other Euler solvers [21, 22].

DOI: 10.1002/fld

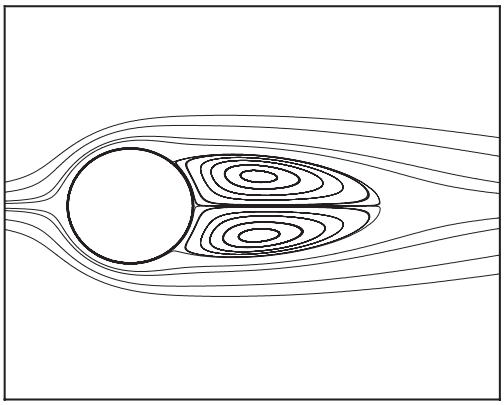

Figure 13. Streamlines around a circular cylinder for $Re = 30$.

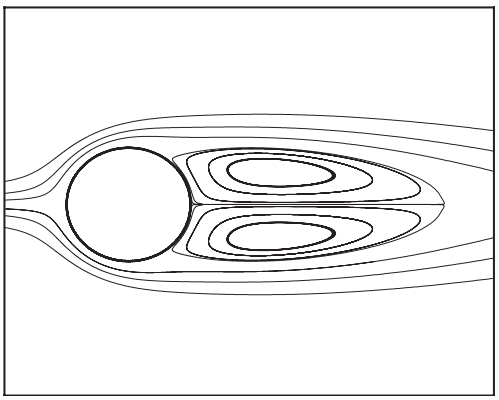

Figure 14. Streamlines around a circular cylinder for $Re = 40$.

6.3. Viscous flows past fixed bodies

In this subsection, the viscous incompressible and compressible flows around fixed bodies are simulated to demonstrate the accuracy of the Navier–Stokes solutions obtained by the local DFD method.

The first two cases consist of incompressible flows around a circular cylinder with Reynolds numbers equal to 30 and 40. The number of the computational mesh points for these cases is 24 641. The Mach number of free stream is taken to be 0.1 in the computation. Figures 13 and 14 display the streamlines around the cylinder at $Re = 30, 40$, respectively. The computed length of recirculation zone is 1.60 for Reynolds number of 30, and 2.06 for Reynolds number of 40. These values agree well with numerical results and experiment data reported in Reference [23].

The third and forth cases consist of compressible viscous flows past a NACA0012 airfoil and the calculations are performed on the same mesh as in the simulation of the inviscid flows. For these

DOI: 10.1002/fld

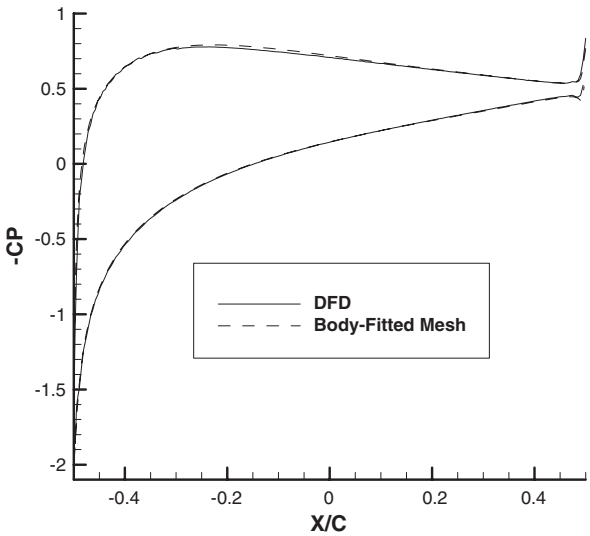

Figure 15. Pressure coefficient on the surface of NACA 0012 airfoil for $M_\infty = 0.8$, $\alpha = 10°$, $Re = 73$. DFD: $C_L = 0.578$, $C_D = 0.222$. Body-fitted mesh: $C_L = 0.587$, $C_D = 0.222$. Results of Reference [14]: $C_L = 0.589$, $C_D = 0.219$.

two cases, the same calculations were also performed on the structured C-type mesh (70×320) and the non-slip condition was imposed on the wall.

In the third case, the Mach number of free stream is 0.8, the angle of attack is 10° and the Reynolds number is 73. The pressure coefficient distribution on the surface is shown in Figure 15, which is compared with the results obtained by the body-fitted mesh solver. The lift and drag coefficients computed by the local DFD method are very close to the results obtained on the body-fitted mesh and those reported in Reference [14]. They also agree well with the results reported in Reference [24]. The skin friction distribution is shown in Figure 16. Also displayed in this figure are the results obtained on the body-fitted mesh and Muller's results given in Reference [25]. Obviously, the agreement is very good. This shows that the shear stress can be predicted well by the local DFD scheme.

In the fourth case, the Mach number of free stream is 0.8, the angle of attack is 10° and the Reynolds number is 500. The surface pressure coefficient distribution is presented in Figure 17 and compared with the results obtained by the body-fitted mesh solver. The lift and drag coefficients computed by the DFD method are close to the results obtained on the body-fitted mesh and those reported in References [14, 24].

6.4. Inviscid transonic flow around oscillating NACA0012 airfoil

The performance of the local DFD method for unsteady flows around moving bodies is investigated with the simulation of the inviscid transonic flow past an oscillating NACA0012 airfoil. The free-stream Mach number is taken as 0.8, the range of pitching angle is between $-0.5°$ and 4.5°, and the oscillation period is $T = 20$ (normalized by L/a_∞ where L is the airfoil length and a_∞ is the sound speed of free stream), which results in a circular frequency of $\omega = \pi/10$.

DOI: 10.1002/fld

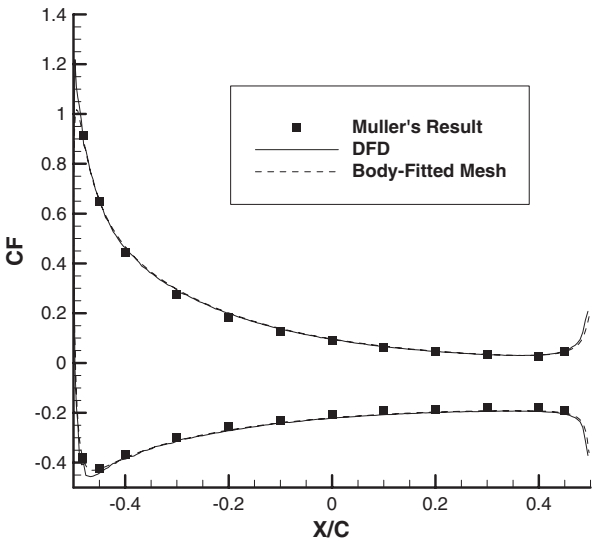

Figure 16. Skin friction coefficient on the surface of NACA0012 airfoil for $M_\infty = 0.8$, $\alpha = 10°$, $Re = 73$.

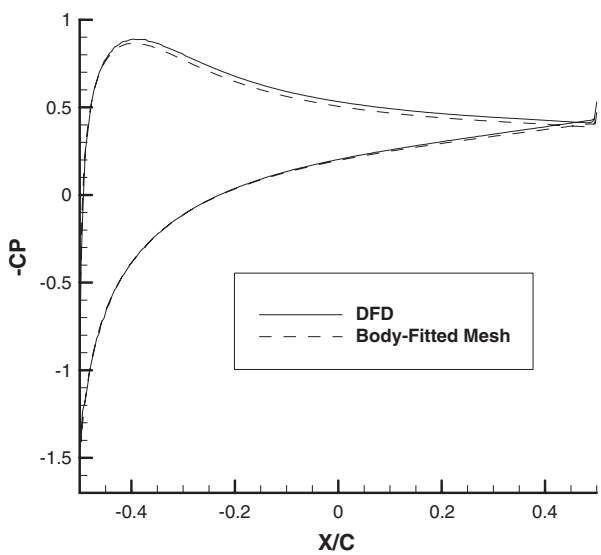

Figure 17. Pressure coefficient on the surface of NACA0012 airfoil for $M_\infty = 0.8$, $\alpha = 10°$, $Re = 500$. DFD: $C_L = 0.464$, $C_D = 0.153$. Body-fitted mesh: $C_L = 0.450$, $C_D = 0.149$. Results of Reference [14]: $C_L = 0.447$, $C_D = 0.147$.

DOI: 10.1002/fld

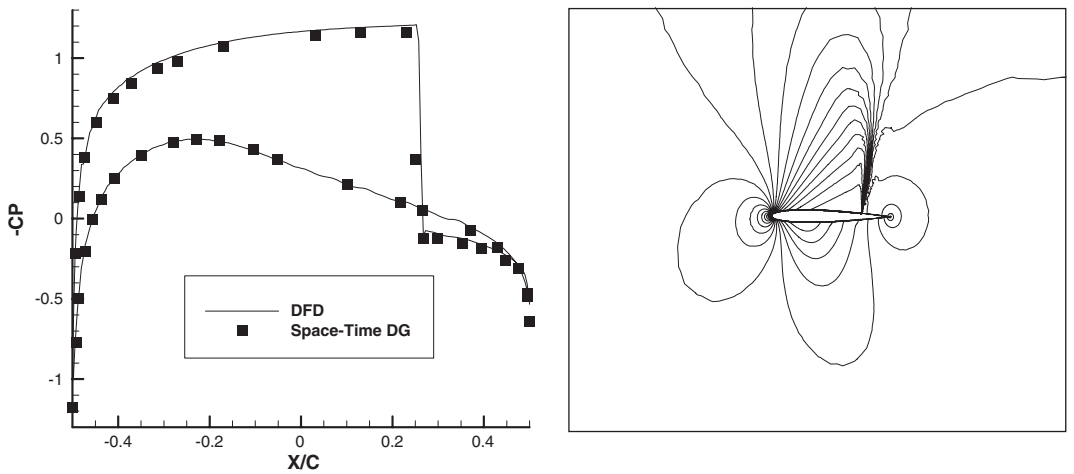

Figure 18. Pressure coefficient on the surface of oscillating NACA0012 airfoil and contours of pressure for $T = 32.0$ and $\alpha = 0.5°$ (pitching downward), $M_\infty = 0.8$, $\omega = \pi/10$.

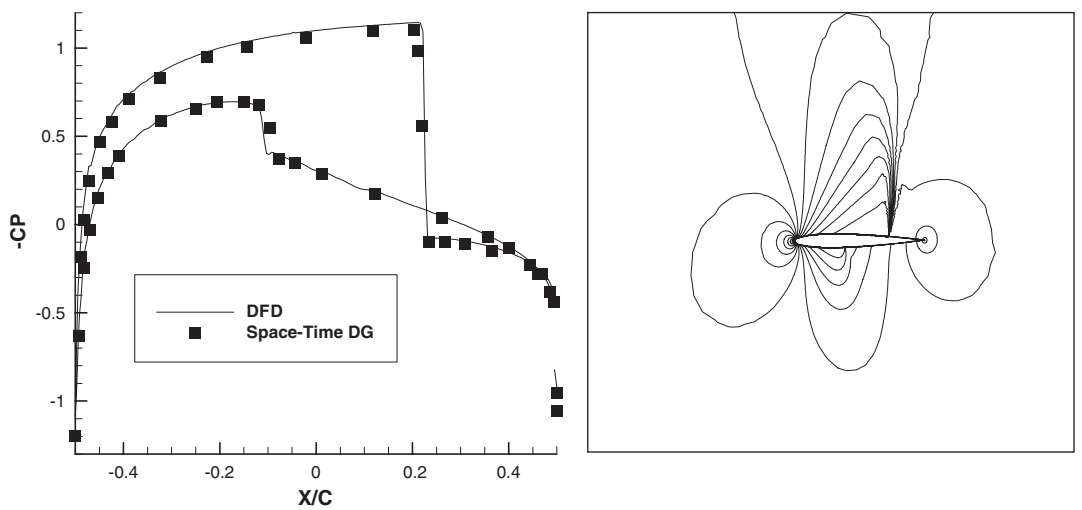

Figure 19. Pressure coefficient on the surface of oscillating NACA0012 airfoil and contours of pressure for $T = 35.0$ and $\alpha = -0.5°$, $M_\infty = 0.8$, $\omega = \pi/10$.

In this test case of the flow around a moving body, the mesh is fixed, and the number of computational points is about 41 400 at each time step. The DFD results are compared with the DG (discontinuous Galerkin) finite element results in Reference [3] which are obtained on a moving mesh with 32 768 quadrangles.

The pressure coefficient C_P on the airfoil surface and contours of pressure at several time steps are shown in Figures 18–22. In Figure 23, the hysteresis curves of the lift and drag coefficients

DOI: 10.1002/fld

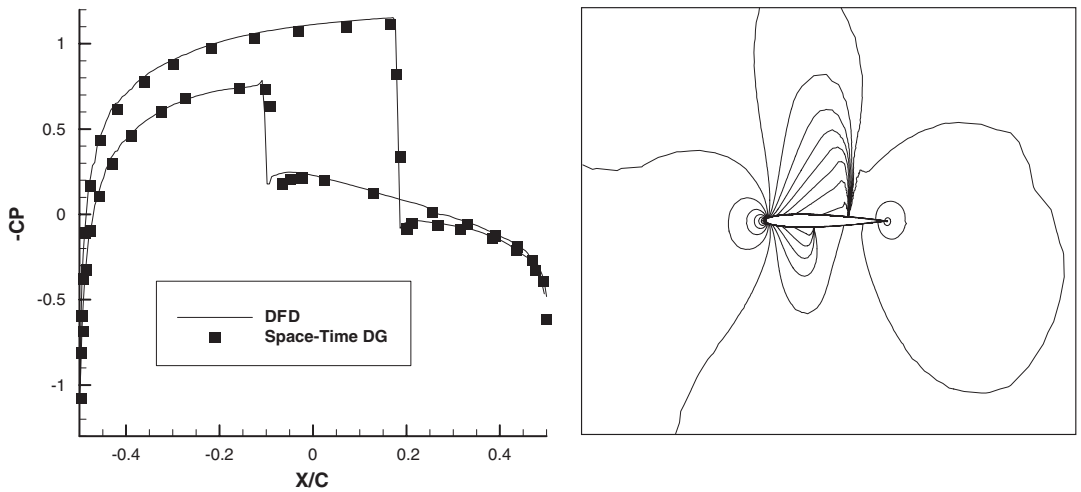

Figure 20. Pressure coefficient on the surface of oscillating NACA0012 airfoil and contours of pressure for $T = 37.5$ and $\alpha = 0.23°$ (pitching upward), $M_\infty = 0.8$, $\omega = \pi/10$.

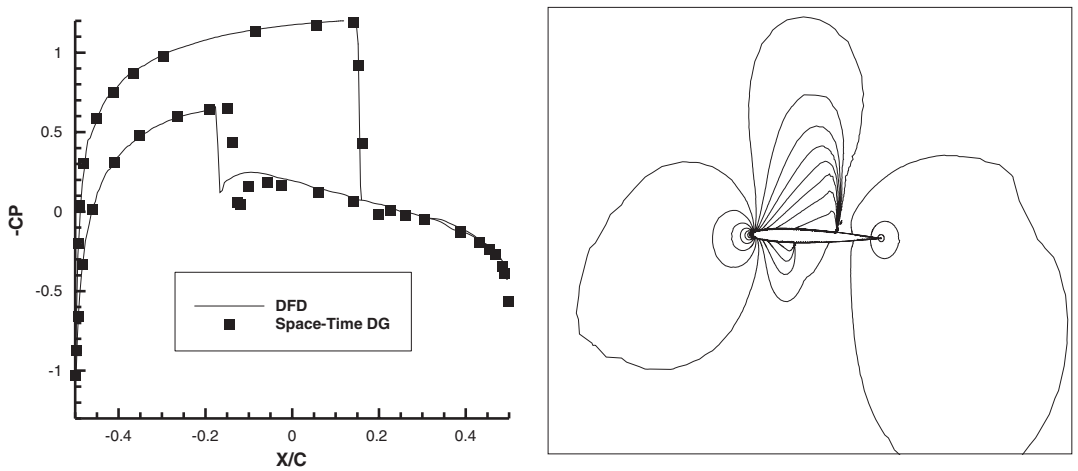

Figure 21. Pressure coefficient on the surface of oscillating NACA0012 airfoil and contours of pressure for $T = 40.0$ and $\alpha = 2.0°$ (pitching upward), $M_\infty = 0.8$, $\omega = \pi/10$.

C_L and C_D are shown. Comparisons between the DFD results and the DG finite element results, except for contours of pressure, are also shown in Figures 18–23. The results computed by the two methods agree each other. Due to differences of the schemes (diffusion terms in particular) and meshes used, the small deviations that appear in Figures 18–23 are acceptable.

DOI: 10.1002/fld

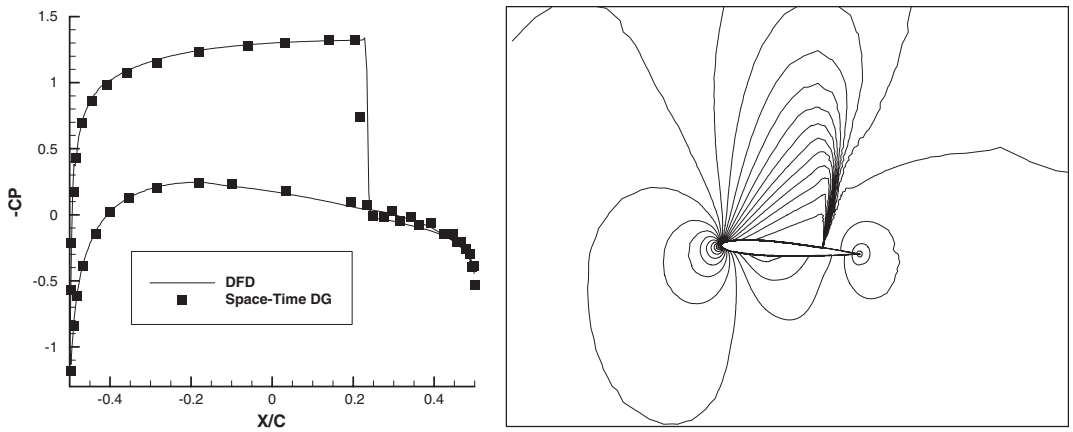

Figure 22. Pressure coefficient on the surface of oscillating NACA0012 airfoil and contours of pressure for $T = 47.0$ and $\alpha = 4.0°$ (pitching downward), $M_\infty = 0.8$, $\omega = \pi/10$.

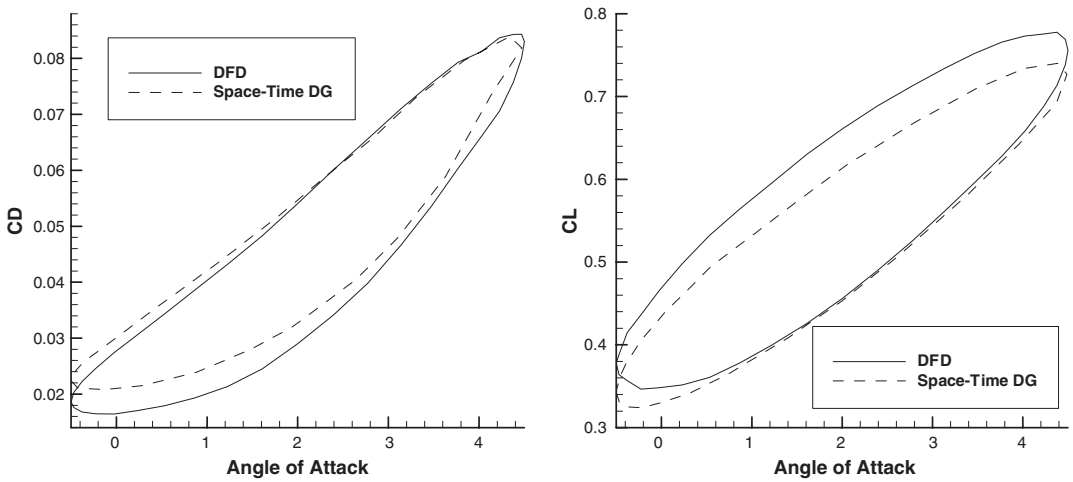

Figure 23. Lift and drag coefficients on oscillating NACA0012 airfoil, $M_\infty = 0.8$, $\omega = \pi/10$.

7. CONCLUSIONS

In this work, the local DFD method was presented to solve the compressible Navier–Stokes/Euler equations in the conservative form. Being different from classical numerical methods, the discrete form of the PDEs can involve some points outside the solution domain. All mesh points are classified into three groups: (1) the points inside the solution domain; (2) the points outside the solution domain on which the computations at some interior points near the wall boundary depend; and (3) the points outside the solution domain which are never used. The values of flow variables at the exterior dependent points are updated at each time step, to satisfy the wall boundary condition, by the approximate form of the solution along the normal direction to the wall boundary.

DOI: 10.1002/fld

In the local DFD method, the boundary can be superimposed upon computational meshes. The mesh can stay fixed while the body is moving, and there is no need to generate a new mesh or deform the previous mesh at each time step. In principle, the mesh generation for complex solution domains becomes quite easy.

A number of numerical experiments have been performed for the flows over fixed and moving bodies, which showed that the local DFD results compare very well with those obtained on the body-fitted meshes and the reference data available in the literature.

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. van der Vegt JJW, van der Ven H. Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. *Journal of Computational Physics* 2002; **182**:546-585.
4. Gaitonde AL. A dual-time method for the solution of the 2-D unsteady Navier–Stokes equations on structured moving meshes. *AIAA Paper 95-1877-CP*, 1995.
5. Glowinski R, Pan TW, Periaux J. Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies. *Computer Methods in Applied Mechanics and Engineering* 1998; **151**:181–194.
6. Peskin CY, McQueen DM. A three-dimensional computational method for flood flows in the heart: I. Immersed elastic fibers in a viscous incompressible fluid. *Journal of Computational Physics* 1989; **81**:372–405.
7. Almgren AS, Bell J, Colella P, Marthaler T. A Cartesian grid projection method for the incompressible Euler equations in complex geometries. *SIAM Journal on Scientific Computing* 1997; **18**:1289–1309.
8. McCorquodale P, Colella P, Johansen H. A Cartesian grid embedded boundary method for the heat equation on irregular domains. *Journal of Computational Physics* 2001; **173**:620–635.
9. 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.
10. Arienti M, Hung P, Morano E, Shepherd JE. A level set approach to Eulerian–Lagrangian coupling. *Journal of Compuational Physics* 2003; **185**:213–251.
11. Fedkiw RP, Aslam T, Merriman B, Osher S. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *Journal of Compuational Physics* 1999; **152**:457–492.
12. Gibou F, Fediw RP, Cheng LT, Kang M. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. *Journal of Computational Physics* 2002; **176**:205–227.
13. Morton KW, Mayers DF. *Numerical Solution of Partial Differential Equations* (2nd edn). Cambridge University Press: Cambridge, MA, 2005.
14. Mavriplis DJ, Jameson A. Multigrid solution of the Navier–Stokes equations on triangular meshes. *AIAA Journal* 1990; **28**(8):1415–1425.
15. Steger J, Warming RF. Flux vector splitting for the inviscid gas dynamic with applications to finite-difference methods. *Journal of Computational Physics* 1983; **40**:263–293.
16. Jameson A, Baker TJ, Weatherill NP. Calculations of transonic flows over a complete aircraft. *AIAA Paper 86-0103*, 1986.
17. Jameson A. Time dependent calculations using multigrid with applications to unsteady flows past airfoils and wings. *AIAA Paper 91-1596*, 1991.
18. Ding H, Shu C, Tang DB. Error estimates of local multiquadric-based differential quadrature (LMQDQ) method through numerical experiments. *International Journal for Numerical Methods in Engineering* 2005; **63**:1513–1529.
19. Lyche T, Morken K. Knot removal for parametric B-spline curves and surfaces. *Computer Aided Geometric Design* 1987; **4**:217–230.
20. Lock RC. Test cases for numerical methods in two-dimensional transonic flows. *AGARD Report No*. 575, 1970.
21. Kroll N, Jain RK. Solution of two-dimensional Euler equations-experience with a finite volume Code. *DFVLR-FB 87-41*, 1987.

DOI: 10.1002/fld

22. Yoshihara H, Sacher P. Test cases for steady inviscid flow fields. *AGARD AR 211*, 1985.
23. Ramakrishnan SV, Bihari B, Szema KY, Shankar V, Ota D. Development and application of an unstructured grid based CFD solver. In *Computational Fluid Dynamics Review 1998*, Hafez M, Oshima K (eds). World Scientific Publishing: Singapore, 1998; 701–709.
24. Grasso F, Jameson A, Martinelli L. A multistage multigrid method for the compressible Navier–Stokes equations. In *Proceedings of the GAMM—Workshop on Numerical simulation of Compressible Navier–Stokes Flows*, Bristeau MO, Glowinski R, Periaux J, Viviand H (eds). INRIA, Sophia-Antipolis. Notes on Numerical Fluid Mechanics, vol. 18. Vieweg-Verlag: Braunschweig, 1986; 123–138.
25. Muller B, Berglind T, Rizzi A. Implicit central difference simulation of compressible Navier–Stokes flow over a NACA0012 airfoil. In *Proceedings of the GAMM—Workshop on Numerical simulation of Compressible Navier–Stokes Flows*, Bristeau MO, Glowinski R, Periaux J, Viviand H (eds). INRIA, Sophia-Antipolis. Notes on Numerical Fluid Mechanics, vol. 18. Vieweg-Verlag: Braunschweig, 1986; 183–200.

DOI: 10.1002/fld