INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **29**: 975–996 (1999)

ON THE IMPLICIT TIME INTEGRATION OF SEMI-DISCRETE VISCOUS FLUXES ON UNSTRUCTURED DYNAMIC MESHES

BRUNO KOOBUS AND CHARBEL FARHAT*

Department of Aerospace Engineering Sciences and Center for Aerospace Structures, University of Colorado, Campus Box 429, Boulder, CO 80309-0429, USA

SUMMARY

Numerical simulations of viscous flow problems with complex moving and/or deforming boundaries commonly require the solution of the corresponding fluid equations of motion on unstructured dynamic meshes. In this paper, a systematic investigation of the importance of the choice of the mesh configuration for evaluating the viscous fluxes is performed when the semi-discrete Navier–Stokes equations are time-integrated using the popular second-order implicit backward difference algorithm. The findings are illustrated with the simulation of a laminar viscous flow problem around an oscillating airfoil. Copyright © 1999 John Wiley & Sons, Ltd.

KEY WORDS: dynamic mesh; Navier–Stokes equations; unsteady viscous flow

1. INTRODUCTION

The numerical solution of unsteady viscous flow problems with complex moving and deforming boundary conditions arises in many scientific and engineering applications, including, to name only a few, blood flow simulations, airfoil oscillations in a separated flow, parachute dynamics, fighter tail buffeting, gate sliding and a large class of free-surface problems. When in such applications some of the fluid domain boundaries undergo a motion with a large amplitude, it becomes necessary to solve the flow equations on a moving and possibly deforming grid. Such a grid is often referred to in the computational aerodynamics literature as a dynamic mesh.

Several approaches have been proposed in the past for solving computational fluid dynamics problems on moving and deforming meshes, among which the two closely related arbitrary Lagrangian–Eulerian (ALE) [1,2] and dynamic mesh [3] methods, the co-rotational approach [4,5], and space–time formulations [6,7] are noted. All of these and other solution methods have to compute the grid displacement and velocity fields, and address the issue of deciding where to evaluate the fluxes when advancing the flow solution from time step t^n to time step t^{n+1}: on the mesh configuration at t^n, on that at t^{n+1}, or on 'phantom' mesh configurations between t^n and t^{n+1}? These issues can be partially addressed by noting that whichever method is selected for solving the fluid equations on dynamic meshes, it is desirable that this method

* Correspondence to: Department of Aerospace Engineering Sciences, University of Colorado, Campus Box 429, Boulder, CO 80309-0429, USA.

CCC 0271–2091/99/080975–22$17.50

Received 19 August 1996
Revised 15 May 1998

preserves the trivial solution of a uniform flow field (in the absence of other boundary conditions, a uniform flow field is a solution of the Navier–Stokes equations). In [8,9], the authors have shown that, except for space–time finite element discretization schemes, this property is verified only when the numerical method chosen for solving the flow problem with moving boundary conditions, and the algorithm constructed for updating the position of the moving grid, satisfy a discrete geometric conservation law (DGCL) that is similar in its principle to the GCL condition that was first pointed out in [10] for structured grids and finite difference schemes. However, because any DGCL condition affects only the convective fluxes, many authors who have discussed flow computations on moving grids have addressed the correct time integration of the convective fluxes only [8,9,11–14]. Little if any has been written about the time-accurate prediction of the diffusive fluxes on unstructured dynamic meshes. Perhaps that, for simplicity, the moving viscous fluxes are often computed on the readily available mesh configuration at time t^n. Nevertheless, a mathematical analysis and justification would certainly improve the understanding and increase the popularity of such a computational approach. The main objective of this paper is to fill this gap in the literature, when finite element and/or finite volume ALE or dynamic meshes formulations are chosen for solving the flow problem with moving and/or deforming boundary conditions. For this purpose, the remainder of this paper is organized as follows.

In Section 2, the ALE formulation of the Navier–Stokes equations for flow problems with moving grids is summarized. The convective fluxes are semi-discretized by an unstructured finite volume method, and the diffusive ones by a piecewise linear finite element approximation. In Section 3, the second-order time-accurate implicit backward difference scheme for unsteady flow computations on fixed grids is considered, and a family of extensions for unsteady flow computations on unstructured dynamics meshes are derived. These extensions lead to a family of numerical algorithms that differ by the choice of the mesh configurations on which the diffusive fluxes are computed while the flow solution is advanced from time step t^n to time step t^{n+1}. In particular, it is shown that using for that purpose the readily available mesh configuration at time step t^n corresponds to a first-order time-accurate strategy for evaluating the viscous fluxes on moving grids. In Section 4, the family of algorithms is applied to the simulation of a two-dimensional transonic laminar viscous flow around an oscillating airfoil. Based on the numerical results obtained for this problem, conclusions are formulated in Section 5.

2. FORMULATION AND SEMI-DISCRETIZATION OF THE ALE NAVIER–STOKES EQUATIONS

Let $\Omega(t) \subset \mathscr{R}^n$ $(n = 2, 3)$ be the flow domain of interest, and $\Gamma(t)$ be its moving and/or deforming boundary (Figure 1). A mapping function between $\Omega(t)$, where time is denoted by t and a grid point's co-ordinates by x, and a reference configuration $\Omega(0)$, where time is denoted by τ and a grid point's co-ordinates by ξ, is introduced as follows:

$$x = x(\xi, \tau); \quad t = \tau. \tag{1}$$

The ALE non-dimensional conservative form of the Navier–Stokes equations describing viscous flows on dynamic meshes can be written as [2,8,9]

$$\begin{aligned}&\left.\frac{\partial J\mathcal{W}}{\partial t}\right|_{\xi} + J\nabla_x\cdot\mathcal{F}^c(\mathcal{W},\dot{x}) = J\nabla_x\cdot\mathcal{R}(\mathcal{W}),\\ &\mathcal{F}^c(\mathcal{W},\dot{x}) = \mathcal{F}(\mathcal{W}) - \dot{x}\mathcal{W}\end{aligned} \tag{2}$$

where a dot superscript designates a time derivative, $J=\det(\mathrm{d}x/\mathrm{d}\xi)$, $\dot{x}=\partial x/\partial\tau|_{\xi}$, $\mathcal{W}$ is the fluid state vector, $\mathcal{F}^c$ denotes the ALE convective fluxes and $\mathcal{R}$ the diffusive fluxes.

For two-dimensional flows, $\mathcal{W}$, $\mathcal{F}$ and $\mathcal{R}$ are given by

$$\mathcal{W}=\begin{pmatrix}\rho\\ \rho v_1\\ \rho v_2\\ E\end{pmatrix},$$

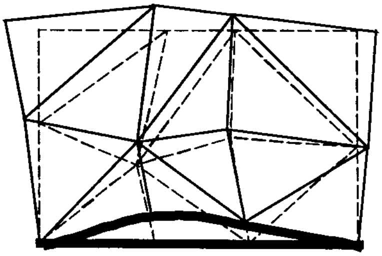

Figure 1. A moving and deforming boundary necessitating a dynamic mesh.

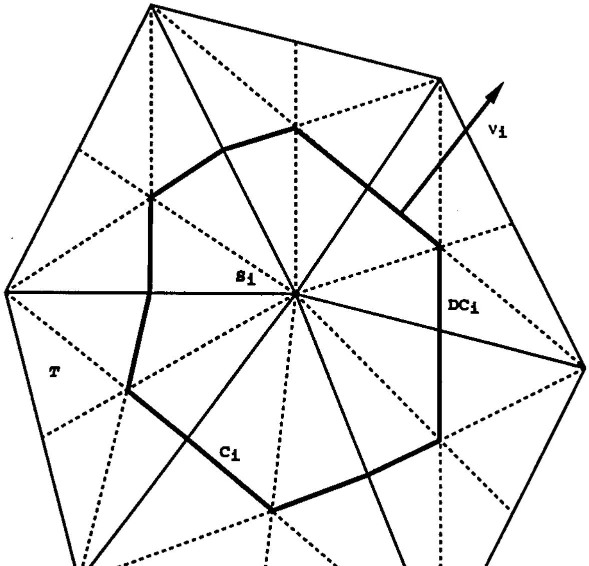

Figure 2. A control volume in a two-dimensional unstructured mesh.

$$\mathscr{F}(\mathscr{W}) = \begin{pmatrix} \mathscr{F}_1(\mathscr{W}) \\ \mathscr{F}_2(\mathscr{W}) \end{pmatrix}, \qquad \mathscr{R}(\mathscr{W}) = \begin{pmatrix} \mathscr{R}_1(\mathscr{W}) \\ \mathscr{R}_2(\mathscr{W}) \end{pmatrix},$$

$$(\mathscr{F}_1(\mathscr{W}), \mathscr{F}_2(\mathscr{W})) = \begin{pmatrix} \rho \vec{v}^t \\ \rho v_1 \vec{v}^t + p \vec{e}_1^t \\ \rho v_2 \vec{v}^t + p \vec{e}_2^t \\ (E+p)\vec{v}^t \end{pmatrix},$$

$$(\mathscr{R}_1(\mathscr{W}), \mathscr{R}_2(\mathscr{W})) = \frac{\mu}{Re} \begin{pmatrix} \vec{0}^t \\ (\sigma \cdot \vec{e}_1)^t \\ (\sigma \cdot \vec{e}_2)^t \\ (\sigma \cdot \vec{v})^t + \dfrac{\gamma}{Pr} \nabla^t e \end{pmatrix},$$

where

$$E = \rho e + \frac{1}{2} \rho \, \|\vec{v}\|^2,$$

$$p = (\gamma - 1)\rho e,$$

$$\sigma = \nabla \vec{v} + \nabla^t \vec{v} - \frac{2}{3} \nabla \cdot \vec{v} I_{\mathrm{d}},$$

$$\vec{v} = (v_1, v_2)^t,$$

$$\gamma = \frac{c_p}{c_v},$$

$$\vec{0} = (0, 0)^t,$$

$$\vec{e}_1 = (1, 0)^t \quad \text{and} \quad \vec{e}_2 = (0, 1)^t$$

and ρ is the fluid density, p its pressure, e its specific internal energy, $\vec{v} = (v_1, v_2)$ its velocity vector, μ its laminar viscosity, γ its specific heat ratio, Re the Reynolds number, Pr the Prandtl number, σ the stress tensor, I_{d} the identity matrix, and a t superscript designates the transpose of a vector.

Equation (2) is semi-discretized on a triangulation (two-dimensional problems) or a tetrahedral mesh (three-dimensional problems) from which one derives a dual mesh defined by control volumes or cells (Figures 2 and 3).

First, Equation (2) is integrated over a reference cell $C_i(0)$ of the ξ space

$$\int_{C_i(0)} \left. \frac{\partial J \mathscr{W}}{\partial t} \right|_{\xi} \mathrm{d}\Omega_{\xi} + \int_{C_i(0)} \nabla_x \cdot \mathscr{F}^c(\mathscr{W}, \dot{x}) J \, \mathrm{d}\Omega_{\xi} = \int_{C_i(0)} \nabla_x \cdot \mathscr{R}(\mathscr{W}) J \, \mathrm{d}\Omega_{\xi}.$$

Since the partial time derivative is evaluated at a constant ξ, it can be moved outside the integral sign to obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{C_i(0)} \mathscr{W} J \, \mathrm{d}\Omega_{\xi} + \int_{C_i(0)} \nabla_x \cdot \mathscr{F}^c(\mathscr{W}, \dot{x}) J \, \mathrm{d}\Omega_{\xi} = \int_{C_i(0)} \nabla_x \cdot \mathscr{R}(\mathscr{W}) J \, \mathrm{d}\Omega_{\xi}. \tag{3}$$

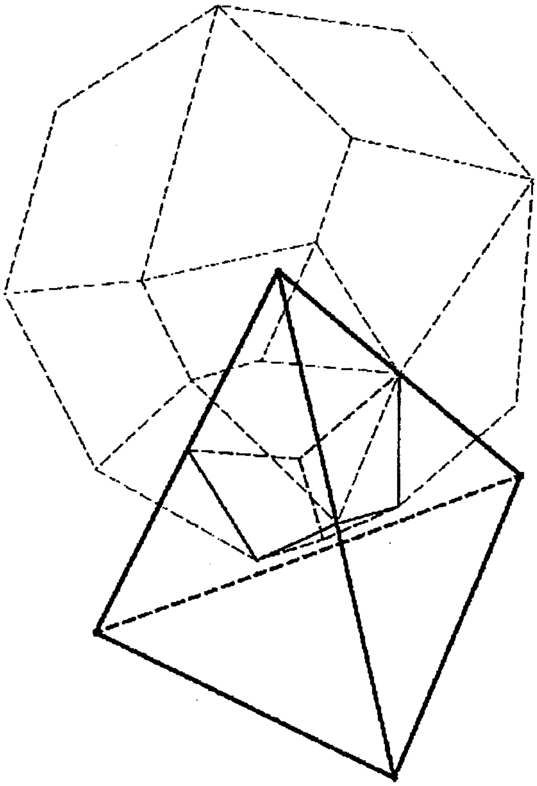

Figure 3. A control volume in a three-dimensional unstructured mesh.

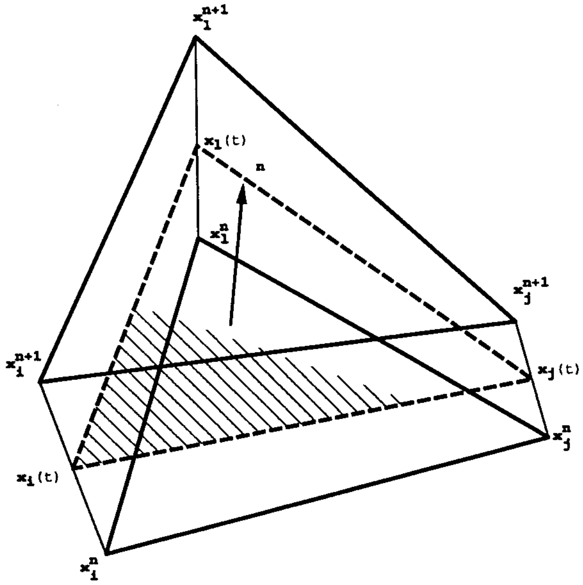

Figure 4. Evolution in time of a cell facet in a three-dimensional mesh.

Next, we switch from the ξ reference space to the x space at time t and transform Equation (3) into

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{C_i(t)} \mathscr{W}\,\mathrm{d}\Omega_x + \int_{C_i(t)} \nabla_x\cdot\mathscr{F}^c(\mathscr{W},\dot{x})\,\mathrm{d}\Omega_\xi = \int_{C_i(t)} \nabla_x\cdot\mathscr{R}(\mathscr{W})\,\mathrm{d}\Omega_x. \tag{4}$$

Finally, the convective and diffusive fluxes are integrated by parts, which leads to

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{C_i(t)} \mathscr{W}\,\mathrm{d}\Omega_x + \int_{\partial C_i(t)} \mathscr{F}^c(\mathscr{W},\dot{x})\vec{n}\,\mathrm{d}\sigma = \int_{\partial C_i(t)} \mathscr{R}(\mathscr{W})\cdot\vec{n}\,\mathrm{d}\sigma, \tag{5}$$

where $\vec{n}$ denotes the normal to the cell boundary $\partial C_i(t)$.

Throughout this paper, the ALE convective fluxes are resolved by a suitable Riemann solver [13,15–17], and approximate the diffusive terms by piecewise linear finite elements. The resulting semi-discrete version of Equation (5) is

$$\frac{\mathrm{d}}{\mathrm{d}t}(A_iW_i) + F_i(W, X, \dot{X}) = R_i(W, X), \tag{6}$$

where $A_i = \int_{C_i(t)} \mathrm{d}\Omega_x$, W_i denotes the average value of $\mathscr{W}$ over the cell $C_i(t)$, F_i and R_i denote respectively, the semi-discrete ALE convective and diffusive fluxes, W is the vector formed by the collection of W_i, and X is the vector of time-dependent grid point positions. Various expressions of the flux approximation $F_i(W, X, \dot{X})$ can be found in [13,15–17]. On the other hand, the piecewise linear finite element approximation of the diffusive fluxes leads to

$$R_i(W, X) = \sum_{T, i\in T} \mathscr{R}(T)\cdot\vec{v}_{i,T}, \tag{7}$$

where T is a triangle in two-dimensional problems and a tetrahedron in three-dimensional ones,

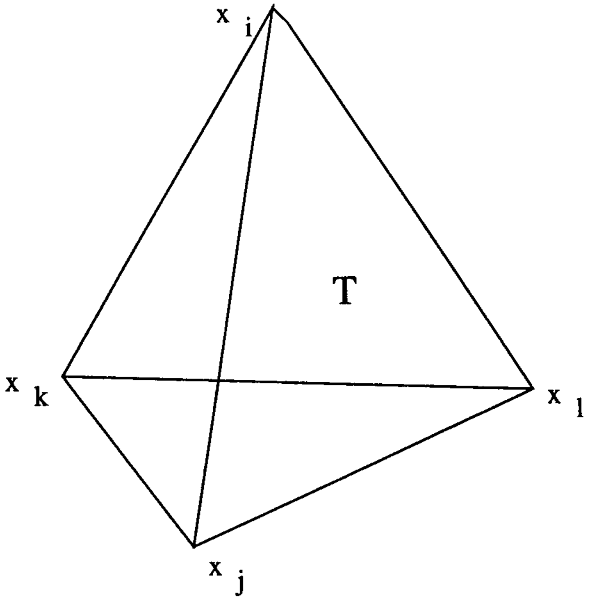

Figure 5. Description of a tetrahedron T.

$$\vec{\nu}_{i,T} = \int_{\partial C_i(x) \cap T} \vec{n} \, \mathrm{d}\sigma \tag{8}$$

and $\mathscr{R}(T) = \mathscr{R}(W)|_T$ is the constant value of $\mathscr{R}(W)$ over T obtained by computing the mean value of $\vec{v}$ over T as $\vec{v}(T) = \frac{1}{3} \Sigma_{k,k \in T} \vec{v}_k$ in two-dimensional problems, and $\vec{v}(T) = \frac{1}{4} \Sigma_{k,k \in T} \vec{v}_k$ in three-dimensional ones. This paper does not discuss the specific problem of computing or updating the grid point position and velocity fields, as this topic has already been discussed in several papers, including [2,3,6]. However, it is noted that all developments and conclusions presented in this paper are independent of the computational method employed for solving the fluid mesh motion problem.

3. IMPLICIT TIME-INTEGRATION OF THE SEMI-DISCRETE ALE NAVIER–STOKES EQUATIONS

Let t^n and $\Delta t^n = t^{n+1} - t^n$ denote the nth time station and the $(n+1)$th time step respectively. Integrating Equation (6) between t^n and t^{n+1} leads to

$$\int_{t^n}^{t^{n+1}} \frac{\mathrm{d}}{\mathrm{d}t} (A_i W_i) \, \mathrm{d}t + \int_{t^n}^{t^{n+1}} F_i(W, X, \dot{X}) \, \mathrm{d}t = \int_{t^n}^{t^{n+1}} R_i(W,X) \, \mathrm{d}t. \tag{9}$$

The proper evaluation of the integrals $\int_{t^n}^{t^{n+1}} F_i(W, X, \dot{X}) \, \mathrm{d}t$ and $\int_{t^n}^{t^{n+1}} R_i(W, X) \, \mathrm{d}t$ raises the question of where to evaluate the convective and diffusive fluxes: on the mesh configuration at (t^n, X^n), on that at (t^{n+1}, X^{n+1}), in between these two configurations, outside these two configurations, or on a combination of all of these configurations? For small time steps, it may not matter on which mesh configuration the fluxes are computed, if the differences between these configurations are insignificant. However, for the rather large time steps that are often employed with implicit time integration schemes, the method of evaluation of the integrals $\int_{t^n}^{t^{n+1}} F_i(X, X, \dot{X}) \, \mathrm{d}t$ and $\int_{t^n}^{t^{n+1}} R_i(W, X) \, \mathrm{d}t$ can have a dramatic effect on the accuracy of the flow solution.

For inviscid flows, the proper evaluation of the convective fluxes on unstructured dynamic meshes has been addressed in [8,9,13] for first-order time integration algorithms, and more recently in [14] for second-order time-accurate schemes. Here, the proper evaluation on moving grids of the diffusive fluxes is examined. To keep this paper as short as possible, the authors consider for this purpose only a second-order implicit time integration algorithm. However, they first summarize the conclusions presented in [14] for the systematic investigation of the importance of the choice of the mesh configurations for evaluating the semi-discrete convective fluxes on moving grids. These conclusions motivate and justify the approach adopted in this paper for computing the semi-discrete diffusive fluxes on dynamic meshes.

3.1. *The generalized second-order backward difference implicit scheme*

A second-order time-accurate implicit algorithm that is popular in CFD is the second-order backward difference scheme (2nd BDF). On fixed grids, this algorithm can be written as

$$\alpha_{n+1} A_i W_i^{n+1} + \alpha_n A_i W_i^n + \alpha_{n-1} A_i W_i^{n-1} + \Delta t^n (F_i(W^{n+1}) - R_i(W^{n+1})) = 0, \tag{10}$$

where the coefficients α_{n-1}, α_n and α_{n+1} are time-dependent if a variable time step Δt^n is employed, and are given by

$$\alpha_{n+1} = \frac{1+2\tau}{1+\tau}, \qquad \alpha_n = -1 - \tau, \qquad \alpha_{n-1} = \frac{\tau^2}{1+\tau}, \qquad \tau = \frac{\Delta t^n}{\Delta t^{n-1}}.$$

Here, the authors propose to generalize the 2nd BDF scheme to CFD on dynamic meshes as follows

$$\alpha_{n+1}A_i(X^{n+1})W_i^{n+1} + \alpha_n A_i(X^n)W_i^n + \alpha_{n-1}A_i(X^{n-1})W_i^{n-1}$$
$$+ \Delta t^n \Psi_i(W^{n+1}, X^{n-l}, \ldots, X^n, \ldots, X^{n+m}, \dot{X}^{n-j}, \ldots, \dot{X}^n, \ldots, \dot{X}^{n+k}) = 0, \qquad (11)$$

where j, k, l and m are positive integers, $X^{n+k} = X(t^{n+k})$, and the numerical flux function Ψ_i is given by

$$\Psi_i = \sum_s \{w_s^c F_i(W^{n+1}, X^{n_s^c}, \dot{X}^{n_s^c}) - w_s^d R_i(W^{n+1}, X^{n_s^d})\}, \qquad (12)$$

where w_s^c and w_s^d are real coefficients satisfying $\Sigma_s\, w_s^c = 1$ and $\Sigma_s\, w_s^d = 1$, and $X^{n_s^c}$ and $X^{n_s^d}$ denote different linear combinations of the mesh configurations $\{X^{n-l}, \ldots, X^n, \ldots, X^{n+m}\}$. This generalization is characterized by the following two properties:

1. If one sets $w_1^c = 1$, $w_s^c = 0$ for $s \neq 1$, and $w_1^d = 1$, $w_s^d = 0$ for $s \neq 1$, one recovers the basic 2nd BDF scheme for CFD computations on fixed grids. Such a property is desirable for the extension to moving grids of any algorithm originally designed for CFD computations on fixed grids. In particular, this is the reason why the numerical flux function is not constructed as $\Psi_i = \Sigma_s\, \{w_s^c F_i(W^{n_s^c}, X^{n_s^c}, \dot{X}^{n_s^c}) - w_s^d R_i(W^{n_s^d}, X^{n_s^d})\}$. Indeed, in the latter case one does not recover the basic 2nd BDF scheme when the mesh is steady, and therefore, one does not take advantage of the well-established properties of this algorithm.
2. For inviscid flows ($R_i = 0$), the proposed generalized time integrator (Equations (11) and (12)) equipped with suitable coefficients w_c^s and mesh configurations $(X^{n_s^c}, \dot{X}^{n_s^c})$, and with $w_s^d = 0$, was shown in [14] to achieve second-order time accuracy for flow problems on moving grids. Hence, the present objective is to extend the generalized 2nd BDF scheme to viscous flow problems on moving grids by augmenting the numerical flux function Ψ_i with suitable coefficients w_s^d and mesh configurations $X^{n_s^d}$. By 'suitable', it is meant coefficients and mesh configurations that allow the proposed time integrator [(11) and (12)] to achieve in practice the best possible time accuracy when applied to the solution of unsteady viscous flow problems on moving grids.

Note that the numerical flux function (12) can be interpreted as the evaluation of the integrals $\int_{t^n}^{t^{n+1}} F_i(W^{n+1}, X(t), \dot{X}(t), \dot{X}(t))\,\mathrm{d}t$ and $\int_{t^n}^{t^{n+1}} R_i(W^{n+1}, X(t))\,\mathrm{d}t$ quadrature rule. *The superscript* $n+1$ *of the fluid state vector* W *is a consequence of the choice of the 2nd BDF implicit scheme as the underlying time integrator*. The real numbers w_s^c and w_s^d are the weighting coefficients of this quadrature rule. The quadrature points $X^{n_s^c}$, $X^{n_s^d}$ and $\dot{X}^{n_s^c}$ are constructed as linear combinations of the mesh configurations $\{X^{n-l}, \ldots, X^n, \ldots, X^{n+m}\}$ and their velocities $\{\dot{X}^{n-j}, \ldots, \dot{X}^n, \ldots, \dot{X}^{n+k}\}$. It follows that the proposed time integration algorithm [(11) and (12)] differs from the basic second-order backward difference scheme (10) in the formulation of the numerical flux function Ψ_i. For flow computations on dynamic meshes, Ψ_i involves a set of mesh configurations and weighting parameters. These additional unknowns should be determined so that for CFD computations on moving grids, the generalized algorithm [(11) and (12)] retains, as much as possible, the second-order time accuracy that characterizes the 2nd BDF algorithm on fixed grids.

3.2. Time discretization of the semi-discrete convective fluxes

It can be shown [18] that a sufficient condition for the time integrator [(11) and (12)] to be mathematically consistent—i.e. to be at least first-order time-accurate on moving grids—is to predict exactly the state of a uniform flow. This sufficient condition, which was formulated in [14] as a geometric conservation law (GCL), can be used to determine the convective coefficients w_s^c and the mesh configurations $(X^{n_s^c}, \dot{X}^{n_s^c})$. For example, it was shown in [14] that for two-dimensional inviscid problems, the time-integrator [(11) and (12)] equipped with the following two w_s^c coefficients and two mesh configurations $(X^{n_s^c}, \dot{X}^{n_s^c})$ satisfies the GCL and achieves in practice second-order time accuracy

$$\begin{cases} w_1^c = \alpha_{n+1}; & w_2^c = -\dfrac{\alpha_{n-1}}{\tau} \\ X^{n_1^c} = \dfrac{X^{n+1} + X^n}{2}; & X^{n_2^c} = \dfrac{X^n + X^{n-1}}{2} \\ \dot{X}^{n_1^c} = \dfrac{X^{n+1} - X^n}{\Delta t^n}; & \dot{X}^{n_2^c} = \dfrac{X^n - X^{n-1}}{\Delta t^{n-1}} \end{cases} . \tag{13}$$

It was also shown in [14] that for three-dimensional inviscid problems, the proposed time integrator [(11) and (12)] equipped with the following four w_s^c coefficients and four mesh configurations $(X^{n_s^c}, \dot{X}^{n_s^c})$ satisfies the GCL and achieves in practice second-order time accuracy

$$\begin{cases} w_1^c = \dfrac{\alpha_{n+1}}{2}; \quad w_2^c = \dfrac{\alpha_{n+1}}{2}; \quad w_3^c = -\dfrac{\alpha_{n-1}}{2\tau}; \quad w_4^c = -\dfrac{\alpha_{n-1}}{2\tau} \\ d_1 = \dfrac{1}{2}\left(1 - \dfrac{1}{\sqrt{3}}\right); \quad d_2 = \dfrac{1}{2}\left(1 + \dfrac{1}{\sqrt{3}}\right) \\ X^{n_1^c} = d_1 X^{n+1} + d_2 X^n; \quad X^{n_2^c} = d_2 X^{n+1} + d_1 X^n \\ X^{n_3^c} = d_1 X^n + d_2 X^{n-1}; \quad X^{n_4^c} = d_2 X^n + d_1 X^{n-1} \\ \dot{X}^{n_1^c} = \dot{X}^{n_2^c} = \dfrac{X^{n+1} - X^n}{\Delta t^n}; \quad \dot{X}^{n_3^c} = \dot{X}^{n_4^c} = \dfrac{X^n - X^{n-1}}{\Delta t^{n-1}} \end{cases} . \tag{14}$$

Hence, for inviscid flow problems, the Ψ_i function does not have the same expression in the two-dimensional and three-dimensional cases (see [8,9,14] for further details). For two-dimensional inviscid flow problems, the semi-discrete convective fluxes are time integrated on two mesh configurations: the first one is at the midpoint between X^{n-1} and X^n, and the second one is at the midpoint between X^n and X^{n+1}. For three-dimensional inviscid flow problems, the semi-discrete convective fluxes are time-integrated on four different mesh configurations, none of which is intuitive.

Finally, it is pointed out that, using the inviscid aeroelastic analysis of the AGARD 445.6 wing as an example, the authors have shown in [14] that (a) for a specified time accuracy, the time integrator [(11) and (12)] equipped with the coefficients and mesh configurations given in Equation (14) can use a time step that is ten times larger than the maximum time step that can be afforded by the same time integrator when the convective fluxes are simply computed on the mesh configuration at time t^n, and (b) it is six times faster CPU-wise when equipped with the coefficients and mesh configurations specified in Equation (14).

3.3. Time discretization of the semi-discrete diffusive fluxes

While it has been successfully exploited for determining the convective coefficients w_s^c and the corresponding mesh configurations $X^{n_s^c}$, the principle of conservation of the state of a uniform flow W^* cannot be used as a guideline for determining the viscous coefficients w_s^d and the corresponding mesh configurations $X^{n_s^d}$. Indeed, setting $W^{n+1} = W^*$ in either Equations (9) or (12) annihilates the contributions of the viscous fluxes, and therefore does not shed any light on where to evaluate these fluxes on dynamic meshes. Here, it is shown that this specific issue can be addressed by performing a Taylor series expansion of $\int_{t^n}^{t^{n+1}} R_i(W^{n+1}, X)\,\mathrm{d}t$ and identifying its leading terms with $\Delta t^n \, \Sigma_s \, w_s^d R_i(W^{n+1}, X^{n_s^d})$, which completes the construction of the time integrator [(11) and (12)]. It turns out that the results are identical for both the two- and three-dimensional cases; for this reason only the three-dimensional case is discussed.

For three-dimensional problems, the semi-discrete diffusive fluxes (7) can be written as

$$R_i(W, X) = \sum_{j \in \{i, V(i)\}} \; \sum_{T \in \mathrm{Supp}(\phi_j)} \frac{1}{\mathrm{Vol}(T)} \, G_j(W, \vec{v}_{j,T}, \vec{v}_{i,T}), \tag{15}$$

where $V(i)$ is the set of the neighbors of vertex i, ϕ_j is the Galerkin linear shape function associated with vertex j, $\mathrm{Supp}(\phi_j) = \{T/\phi_j \neq 0 \text{ on } T\}$, $\mathrm{Vol}(T)$ denotes the volume of tetrahedron T, G_j is a five-component vector given by

$$G_j(W, \vec{v}_{j,T}, \vec{v}_{i,T}) = -\frac{\mu}{Re} \begin{pmatrix} G_{1j}(W, \vec{v}_{j,T}, \vec{v}_{i,T}) \\ G_{2j}(W, \vec{v}_{j,T}, \vec{v}_{i,T}) \\ G_{3j}(W, \vec{v}_{j,T}, \vec{v}_{i,T}) \end{pmatrix} \cdot \vec{v}_{i,T}$$

and G_{1j}, G_{2j} and G_{3j} are also five-component vectors that can be written as

$$\begin{pmatrix} G^t_{1j}(W, \vec{v}_{j,T}, \vec{v}_{i,T}) \\ G^t_{2j}(W, \vec{v}_{j,T}, \vec{v}_{i,T}) \\ G^t_{3j}(W, \vec{v}_{j,T}, \vec{v}_{k,T}) \end{pmatrix} = \Bigg(\vec{0}, \vec{r}(W_j, \vec{v}_{j,T}, \vec{e}_1),$$

$$\vec{r}(W_j, \vec{v}_{j,T}, \vec{e}_2), \vec{r}(W_j, \vec{v}_{j,T}, \vec{e}_e), \vec{r}(W_j, \vec{v}_{j,T}, \vec{v}(T)) + \frac{\gamma}{Pr} e_j \vec{v}_{j,T} \Bigg).$$

In the above expression, $\vec{0} = (0, 0, 0)^t$, $\vec{e}_1 = (1, 0, 0)^t$, $\vec{e}_2 = (0, 1, 0)^t$, $\vec{e}_3 = (0, 0, 1)^t$, and $\vec{r}$ is a three-component vector defined by

$$\vec{r}(W_j, \vec{v}_{j,T}, \vec{b}) = \left[M(\vec{v}_j) \otimes M^t(\vec{v}_{j,T}) + M^t(\vec{v}_j) \otimes M(\vec{v}_{j,T}) - \frac{2}{3} \vec{v}_j \cdot \vec{v}_{j,T} I_{\mathrm{d}} \right] \cdot \vec{b},$$

where $\vec{b} \in \mathbb{R}^3$ and the following notation is used for $\vec{c} \in \mathbb{R}^3$ and $A, B \in \mathbb{R}^3 \times \mathbb{R}^3$

$$M(\vec{c}) = \begin{pmatrix} c_1 & c_1 & c_1 \\ c_2 & c_2 & c_2 \\ c_3 & c_3 & c_3 \end{pmatrix}; \qquad (A \otimes B)_{ij} = A_{ij} B_{ij}; \quad i, j = 1, \ldots, 3.$$

From Equation (15) it follows that the variation in time of each of the quantities $\vec{v}_{i,T}$, $\vec{v}_{j,T}$ and $\mathrm{Vol}(T)$ must be determined before $\int_{t^n}^{t^{n+1}} R_i(W^{n+1}, X)\,\mathrm{d}t$ can be evaluated. This requires describing first the evolution within the time interval $[t^n, t^{n+1}]$ of the position of a fluid grid point X_i. Among the many possible parametrizations of $X_i(t)$, the following linear evolution is selected

$$X_i(t) = \delta(t)X_i^{n+1} + (1-\delta(t))X_i^n$$
$$0 \le \delta(t) = \frac{t-t^n}{\Delta t} \le 1 \tag{16}$$

because of its simplicity, and because the most popular methods for updating a dynamic mesh are linear two-step algorithms that involve the mesh positions at time steps t^n and t^{n+1}. Furthermore, it is noted that choosing another parametrization of $X_i(t)$ than that described in Equation (16) does not affect the fundamental ideas described in this paper.

For the linear transformation described in Equation (16), the evolution in time of a segment of a cell boundary is graphically depicted in Figure 4, and the variations of $\vec{v}_{j,T}$, $\vec{v}_{i,T}$ and Vol(T) as functions of $\delta(t)$ and Δt are given by

$$\begin{aligned}\vec{v}_{i,T} &= \int_{\partial C_i(X)\cap T} \vec{n}\,\mathrm{d}\sigma\\ &= \frac{1}{6}(X_k - X_j) \wedge (X_l - X_j)\\ &= \frac{1}{6}[(\delta(t)\Delta t(\dot{X}_k - \dot{X}_j) + X_k^n - X_j^n) \wedge (\delta(t)\Delta t(\dot{X}_l - \dot{X}_j) + X_l^n - X_j^n)],\end{aligned}$$

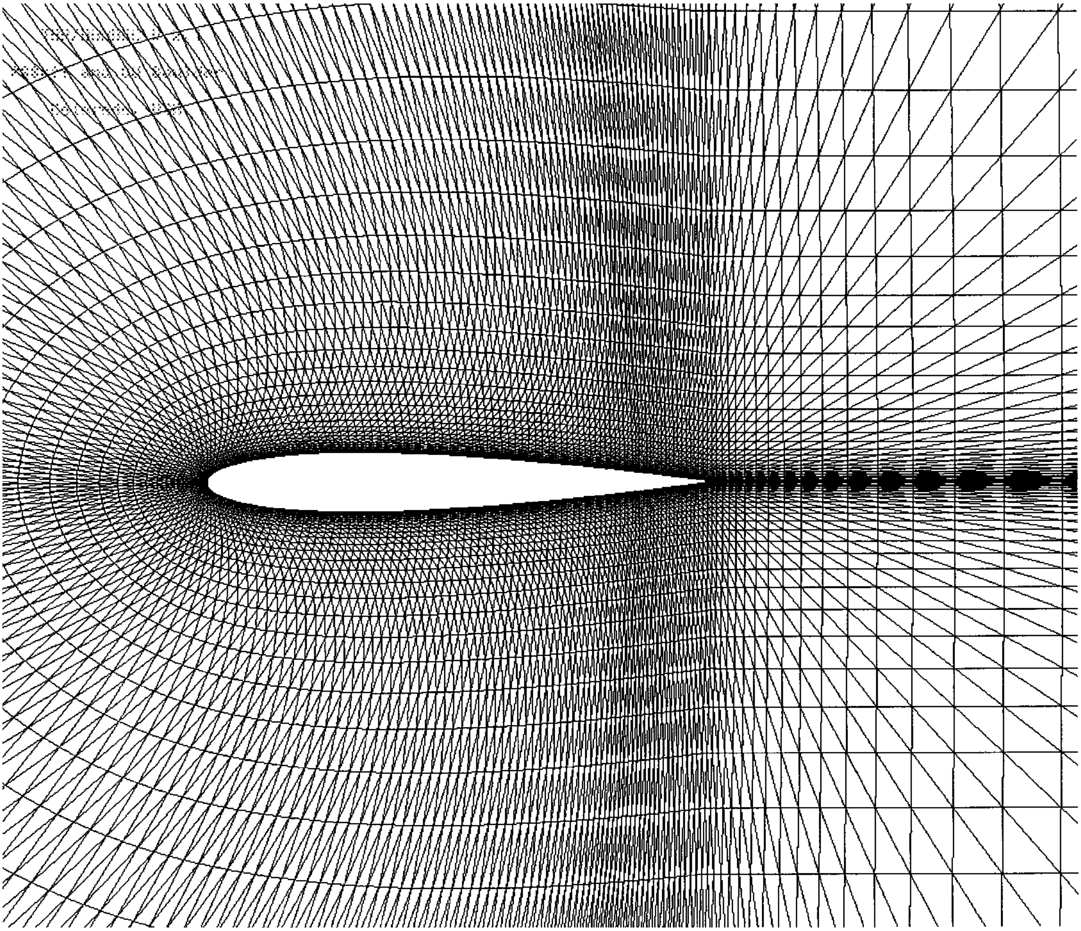

Figure 6. Partial view of the C-mesh around the NACA0015 airfoil.

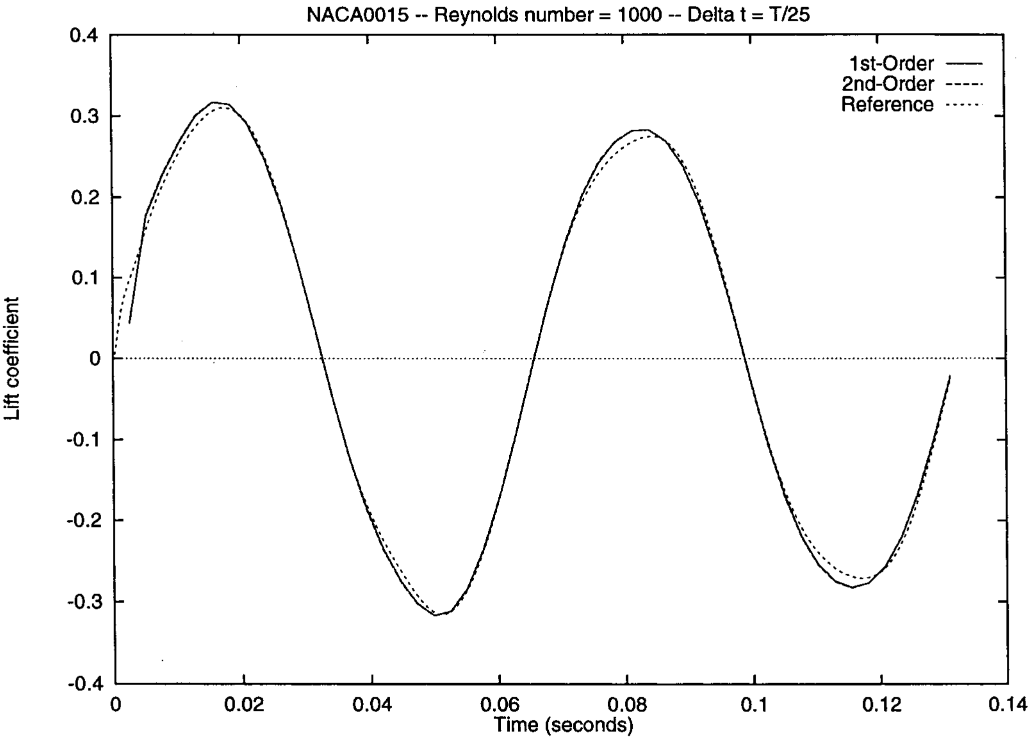

Figure 7. Evolution of the lift coefficient for $\Delta t = T/25$ and $Re = 10^3$.

$$\vec{v}_{j,T} = \frac{1}{6}[(\delta(t)\Delta t(\dot{X}_l - \dot{X}_i) + X_l^n - X_i^n) \wedge (\delta(t)\Delta t(\dot{X}_k - \dot{X}_i) + X_k^n - X_i^n)],$$

$$\begin{aligned}\mathrm{Vol}(T) &= \frac{1}{6}((X_k - X_j) \wedge (X_l - X_j))\cdot(X_j - X_i)\\ &= \frac{1}{6}[(\delta(t)\Delta t(\dot{X}_k - \dot{X}_j) + X_k^n - X_j^n) \wedge (\delta(t)\Delta t(\dot{X}_l - \dot{X}_j) + X_l^n - X_j^n)]\\ &\quad \cdot(\delta(t)\Delta t(\dot{X}_j - \dot{X}_i) + X_j^n - X_i^n),\end{aligned}$$

where

$$\dot{X} = \frac{X^{n+1} - X^n}{\Delta t} \tag{17}$$

and X_i, X_j, X_k and X_l are the four vertices defining tetrahedron T and positioned as shown in Figure 5.

From Equation (15) and the above expressions of $\vec{v}_{i,T}$, $\vec{v}_{j,T}$ and Vol(T), it follows that

$$\int_{t^n}^{t^{n+1}} R_i(W^{n+1}, X)\,\mathrm{d}t = \Delta t^n \int_0^1 \left(\sum_{k=0}^{4} a_k \Delta t^{n^k} \delta(t)^k / \sum_{k=0}^{3} b_k \Delta t^{n^k} \delta(t)^k \right) \mathrm{d}\delta(t), \tag{18}$$

where a_k is a function of (X^n, X^{n+1}, W^{n+1}), and b_k a function of (X^n, X^{n+1}).

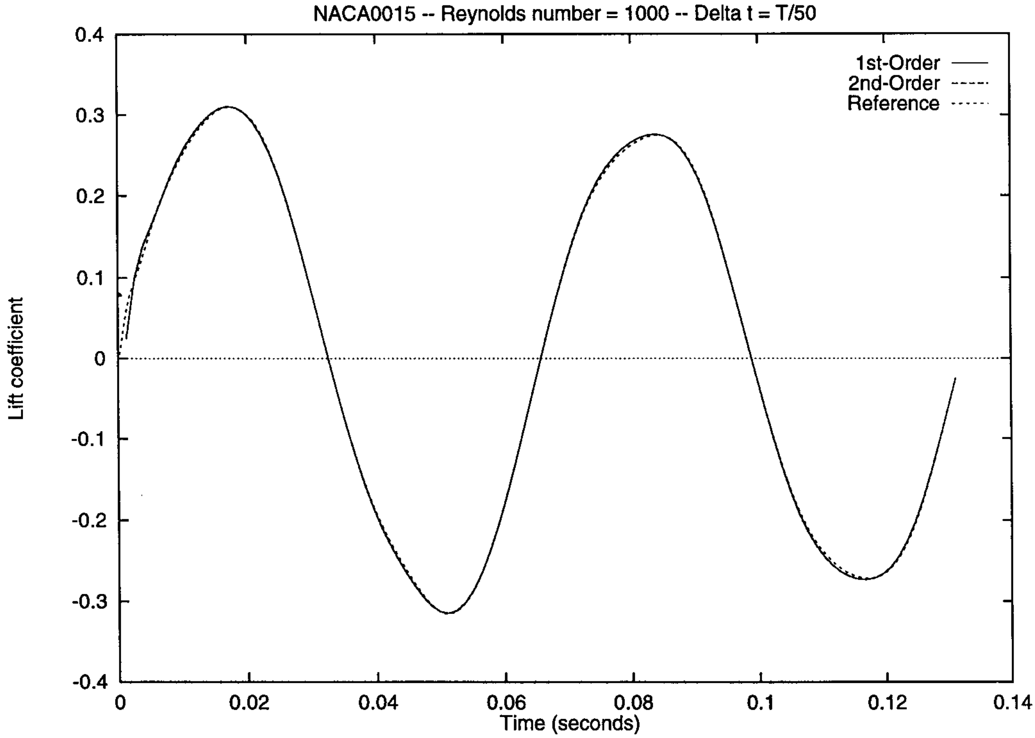

Figure 8. Evolution of the lift coefficient for $\Delta t = T/50$ and $Re = 10^3$.

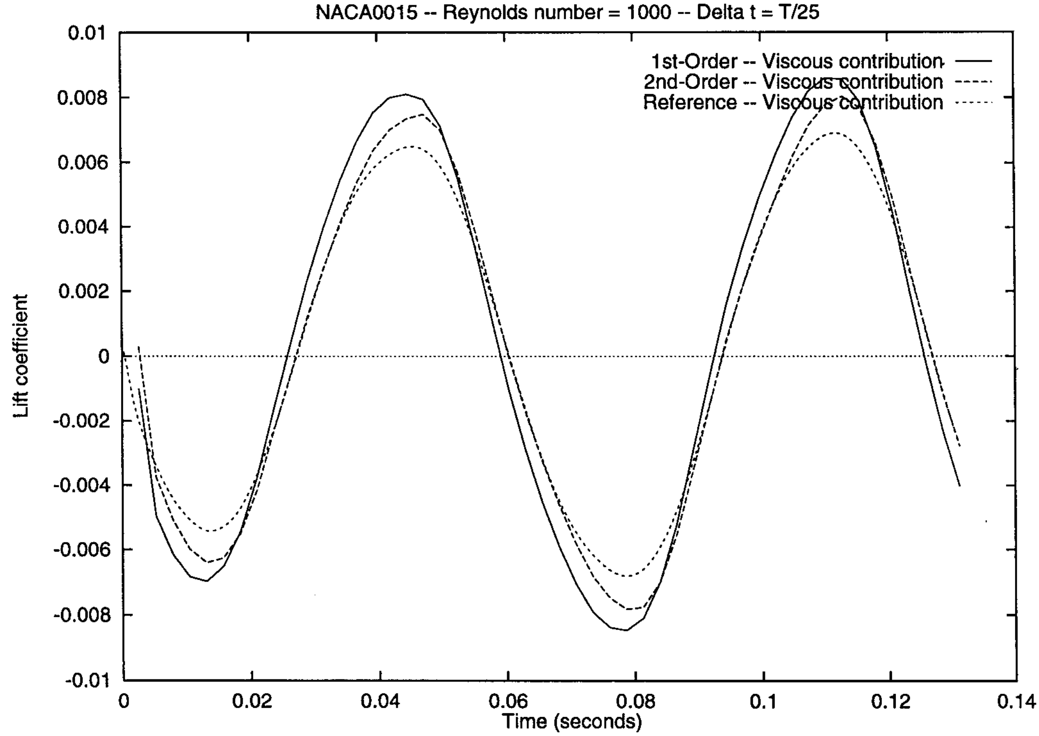

Figure 9. Evolution of the viscous contribution to the lift coefficient for $\Delta t = T/25$ and $Re = 10^3$.

Now, expanding the integrand of integral (18) as a Taylor series in Δt^n gives

$$\int_{t^n}^{t^{n+1}} R_i(W^{n+1}, X)\,\mathrm{d}t = \Delta t^n \int_0^1 (c_0 + c_1\Delta t^n \delta(t) + c_2 \Delta t^{n^2}\delta(t)^2 + \cdots)\,\mathrm{d}\delta(t), \tag{19}$$

where every coefficient c_k is yet another function of (X^n, X^{n+1}, W^{n+1}). Depending on the quadrature rule chosen for evaluating the above integral, the following time accuracy results can be established:

(a) If the integration of Equation (19) is performed at $\delta(t) = 0$, only constant polynomials in $\delta(t)$ are integrated exactly, and one obtains

$$\int_{t^n}^{t^{n+1}} R_i(W^{n+1}, X)\,\mathrm{d}t = \Delta t^n R_i(W^{n+1}, X^n) + O(\Delta t^{n^2}). \tag{20}$$

(b) If the midpoint rule is used for evaluating Equation (19), the constant and linear terms in $\delta(t)$ are integrated exactly, and therefore

$$\int_{t^n}^{t^{n+1}} R_i(W^{n+1}, X)\,\mathrm{d}t = \Delta t^n R_i(W^{n+1}, X^{n+1/2}) + O(\Delta t^{n^3}); \quad X^{n+1/2} = \frac{X^{n+1} + X^n}{2}. \tag{21}$$

(c) On the other hand, if the two-point integration rule at $\delta(t) = \frac{1}{2} \mp 1/(2\sqrt{3})$ is used, all constant, linear, and quadratic monomials in $\delta(t)$ are integrated exactly, and one obtains

$$\int_{t^n}^{t^{n+1}} R_i(W^{n+1}, X)\,\mathrm{d}t = \frac{\Delta t^n}{2}[R_i(W^{n+1}, X^{m1}) + R_i(W^{n+1}, X^{m2})] + O(\Delta t^{n^5}) \tag{22}$$

$$m1 = n + \frac{1}{2} - \frac{1}{2\sqrt{3}},$$

$$m2 = n + \frac{1}{2} + \frac{1}{2\sqrt{3}},$$

$$X^{n+\beta} = \beta X^{n+1} + (1-\beta)X^n.$$

Hence, if the objectives are to time integrate the semi-discrete Navier–Stokes equations (6) using the generalized 2nd BDF algorithm [(11) and (12)], and to retain, as much as possible on moving grids, the time accuracy of the 2nd BDF scheme on fixed grids, the theoretical analysis presented above suggests that as far as the semi-discrete viscous fluxes are concerned, the proposed time integrator [(11) and (12)] should be constructed using the following single coefficient w_1^d and single mesh configuration $X^{n_1^d}$

$$w_1^d = \frac{1}{2}; \qquad X^{n_1^d} = \frac{X^{n+1} + X^n}{2}, \tag{23}$$

i.e. the semi-discrete diffusive fluxes should be integrated on the midpoint configuration between X^n and X^{n+1}. Note that this result is not as intuitive as it may seem, because as shown in Equation (14), it does not apply for the semi-discrete convective fluxes.

The truncation error analysis performed herein can be repeated for a first-order time-accurate version of the generalized implicit algorithm [(11) and (12)] (e.g. see [8,9]), as well as for higher-order versions. Such an analysis leads to the following guidelines. When the unsteady Navier–Stokes equations are semi-discretized on a moving grid, and a given time integration scheme $\mathscr{S}$ is generalized to advance the flow solution on this moving grid, the integral $\int_{t^n}^{t^{n+1}} R_i(W^{n+1}, X)\,\mathrm{d}t$ should be computed using

(a) the configuration (t^n, X^n) if $\mathscr{S}$ is first-order time-accurate on fixed grids,
(b) the midpoint configuration if $\mathscr{S}$ is second-order time-accurate,
(c) the two-point quadrature rule (22) if $\mathscr{S}$ is third- or fourth-order time-accurate,

in order to ensure that the truncation error associated with the mesh motion is of the same order as the truncation error that characterizes $\mathscr{S}$ on fixed grids.

4. APPLICATIONS

So far it has been concluded that if a given flow solver is only first-order time-accurate on fixed grids, the moving diffusive fluxes that appear in its extension to dynamic meshes can be simply computed on the (t^n, X^n) mesh configuration. In that case, the overall solution of the viscous flow problem with moving boundary conditions can be expected to remain first-order time-accurate if the moving convective fluxes are properly discretized (i.e. if the time integrator [(11) and (12)] is equipped with the proper w_s^c coefficients and mesh configurations $(X^{n_s^c}, \dot{X}^{n_s^c})$ for evaluating the convective fluxes on moving grids). However, if the basic flow solver is second- or higher-order time-accurate on fixed grids, it becomes interesting to determine whether on dynamic meshes, the moving diffusive fluxes can still be evaluated on the (t^n, X^n) mesh configuration without reducing in practice the order of time accuracy of the flow solution, or whether these diffusive fluxes should be computed by the second- or higher-order computational strategies described in Section 3.3. In order to address this issue, the authors consider the simulation of the laminar viscous flow past a NACA0015 airfoil forced into the harmonic pitching motion

$$\alpha(t) = \alpha_0 + \alpha_{\max} \sin \omega t,$$

where $\alpha(t)$, α_0 and $\alpha_{\max}$ denote respectively, the instantaneous angle of attack, the initial angle of attack and the maximum perturbation around this initial angle of attack, and ω denotes the circular frequency of oscillation. Here, these parameters are set to

$$\alpha_0 = 0°, \qquad \alpha_{\max} = 10°, \qquad \omega = 95 \text{ rad s}^{-1}$$

and the free-stream Mach number is set to $M_\infty = 0.85$. Note that the frequency associated with $\omega = 95$ rad s^{-1} is $f = 15$ Hz, and that this frequency corresponds to the first torsional mode of a realistic aircraft wing.

An appropriate Reynolds number for this aerodynamic application is in the range of millions. Usually, for such a high Reynolds number, a numerical simulation calls for a turbulence model. However, in order not to influence the results of the present investigation by the specifics of a given turbulence model, the numerical simulations are restricted to a lower Reynolds number, $Re = 10^3$, and solve directly the governing Navier–Stokes equations on a moving grid.

The computational domain around the NACA0015 airfoil is descritized using a C-mesh with a unit chord $L_C = 1$ and 57010 vertices (see Figure 6). Within the boundary layer, the distance between two grid points is typically $d = 8.7 \times 10^{-4}$. Hence, the numerical Reynolds number associated with this C-mesh is

$$Re^{\text{num}} = \frac{M_\infty \times L_C}{d} = \frac{0.85 \times 1.0}{7.5 \times 10^{-4}} = 1.133 \times 10^3 > Re,$$

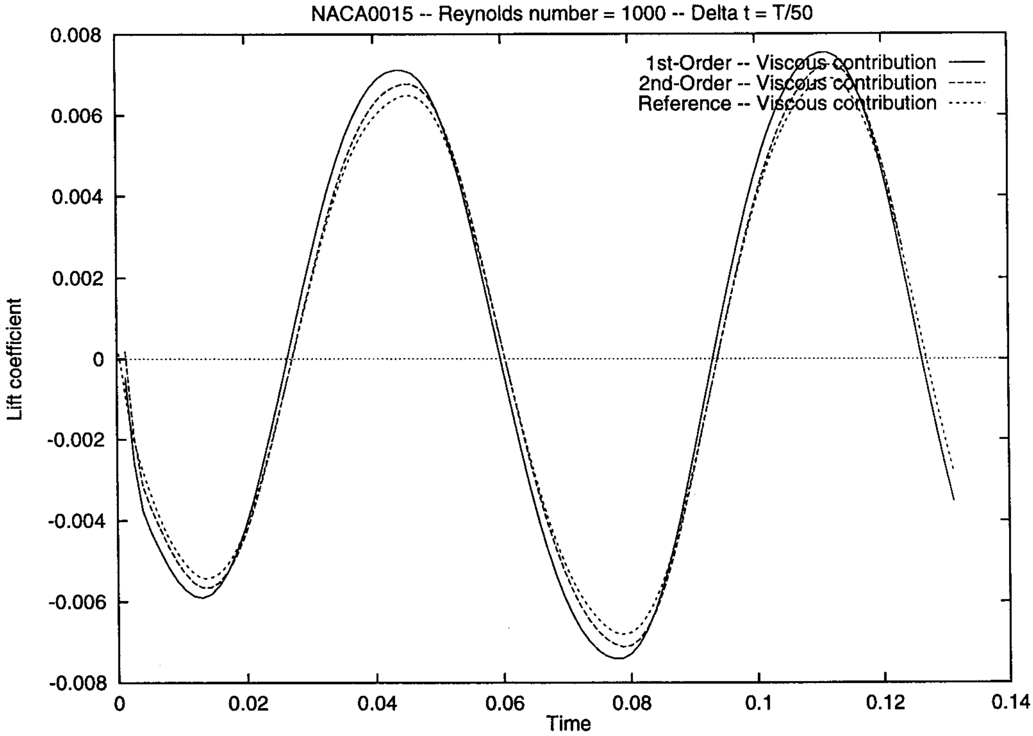

Figure 10. Evolution of the viscous contribution to the lift coefficient for $\Delta t = T/50$ and $Re = 10^3$.

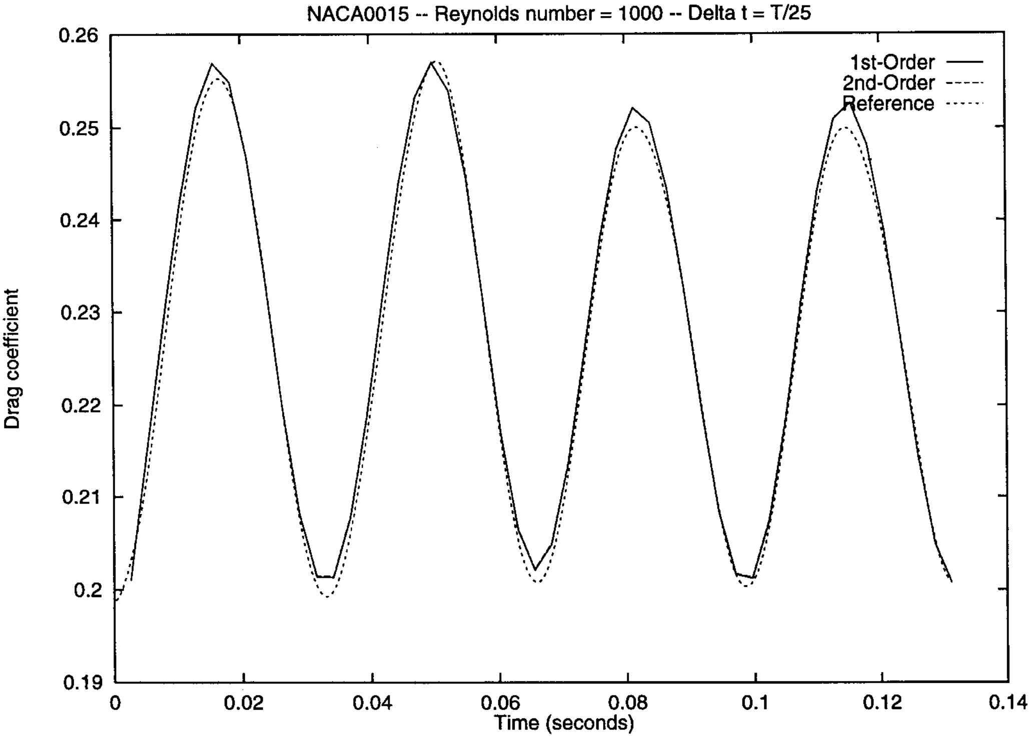

Figure 11. Evolution of the drag coefficient for $\Delta t = T/25$ and $Re = 10^3$.

which suggests that this C-mesh is capable of resolving both the large and small structures of the flow along with all scales in between.

For all numerical simulations discussed herein, a two-dimensional unstructured flow solver is employed that can be described as follows. It incorporates an ALE formulation to allow the grid points to displace in a Lagrangian fashion, or be held fixed in an Eulerian manner, or be moved in some specified way to give a continuous and automatic rezoning capability, depending on the needs of the physical problem to be solved. At each time step, the position of the fluid dynamic mesh is updated using the spring analogy method introduced in [3] and refined in [19]. The flow solver here combines the Roe upwinding scheme [20] for the computation of the convective fluxes with a Galerkin centered scheme for the approximation of the viscous terms. Second-order spatial accuracy is achieved through the use of a piecewise linear interpolation method [13,15–17] that follows the principle of the monotonic upwind scheme for conservative laws (MUSCL) procedure [21]. Time integration is carried out by the generalized 2nd BDF implicit time integrator [(11) and (12)], where the w_s^c coefficients and the mesh configurations $(X^{n_s^c}, \dot{X}^{n_s^c})$ are set as in Equation (13). The authors remind the reader that for inviscid flows, the time integrator [(11) and (12)] equipped with the parameters given in (13) achieves second-order time accuracy on moving grids [14]. For the purpose of the present investigation, the generalized 2nd BDF time integrator is also equipped with two different choices for the viscous coefficients w_s^d and mesh configurations $(X^{n_s^d}, \dot{X}^{n_s^d})$

1. $w_1^d = 1$, $X^{n_1^d} = X^n$, which corresponds to evaluating the moving diffusive fluxes on the (t^n, X^n) mesh configuration.
2. $w_1^d = 1$, $X^{n_1^d} = X^{n+1/2} = (X^{n+1} + X^n)/2$, which corresponds to evaluating the moving diffusive fluxes on the midpoint mesh configuration.

Two series of numerical simulations are performed corresponding to the two different strategies specified above for computing the moving diffusive fluxes. In each case, twice the computations are performed: first with a time step fixed to $\Delta t = T/25$, then with a twice smaller time step $\Delta t = T/50$. Here, T denotes the period of oscillations of the airfoil ($T = 1/f = 1/15 = 0.0667$ s). Note that both of these time steps are typical of implicit computations. The authors also compute an additional solution using a much smaller time step $\Delta t = T/250$ and the second-order scheme for the evaluation of the moving diffusive fluxes. This additional solution is labeled the ‘reference’ solution because of its small computational time step.

The lift and drag time histories predicted by the numerical simulations are reported in Figures 7–14. The focus on lift and drag is motivated by the interest in computational aeroelasticity. These figures show that

1. the numerical solutions obtained with $\Delta t = T/50$ are ‘converged’ in time.
2. when the generalized 2nd BDF time integrator is equipped with the two coefficients w_s^c and two mesh configurations $(X^{n_s^c}, \dot{X}^{n_s^c})$ given in Equation (13) for evaluating the moving convective fluxes, the same accuracy is achieved for the lift and drag coefficients whether the moving diffusive fluxes are computed on the (t^n, X^n) mesh configuration, or on the midpoint mesh configuration $(t^{n+1/2}, X^{n+1/2})$.
3. on the other hand, if one focuses exclusively on the contribution of the viscous terms to the lift coefficient, one can observe in Figures 9 and 10 that computing the moving viscous fluxes on the midpoint configuration reduces by 50% the amplitude error obtained when these fluxes are computed on the current mesh configuration, and does not generate any phase error.

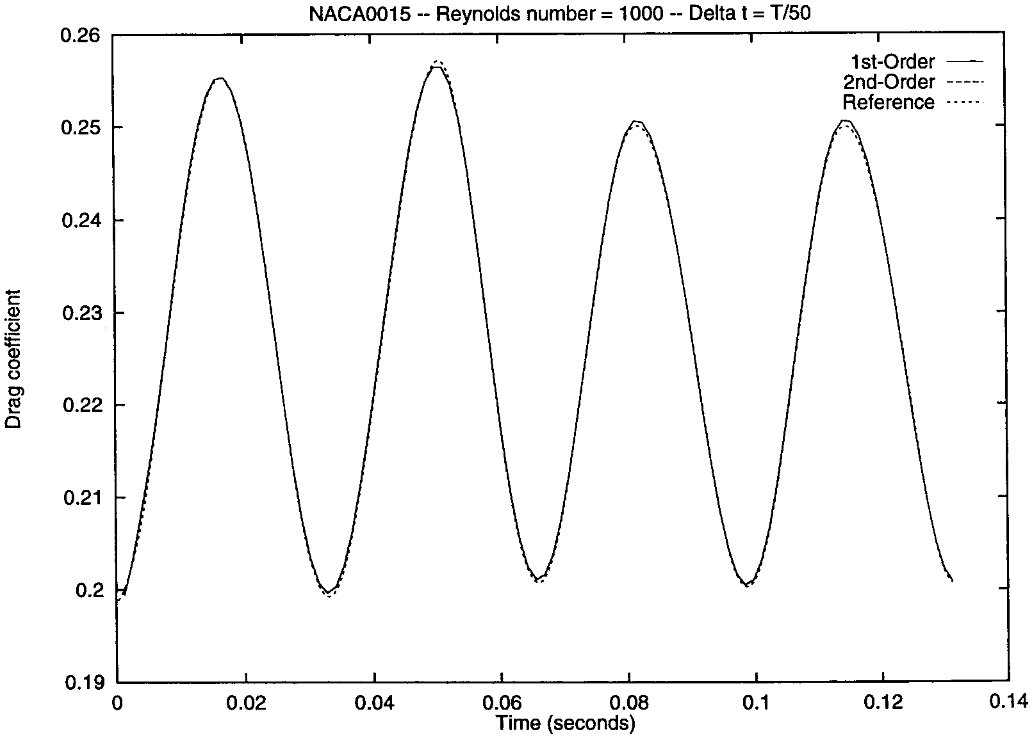

Figure 12. Evolution of the drag coefficient for $\Delta t = T/50$ and $Re = 10^3$.

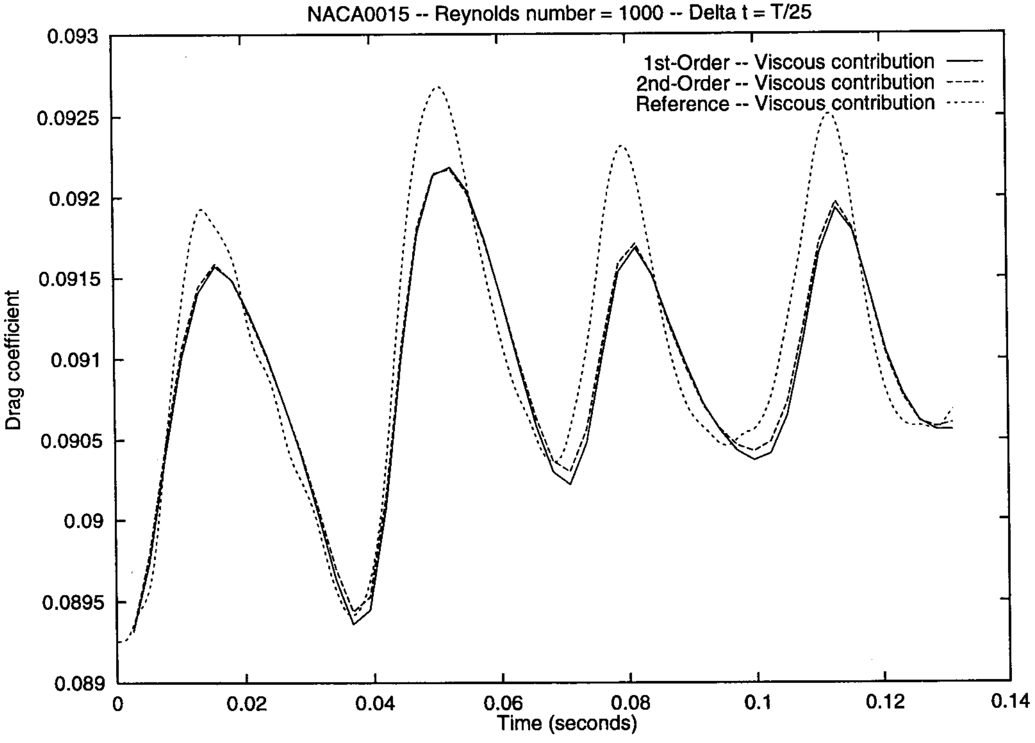

Figure 13. Evolution of the viscous contribution to the drag coefficient for $\Delta t = T/25$ and $Re = 10^3$.

4. from the above two observations, it follows that the second-order computational strategy (21) for evaluating the unsteady viscous fluxes on moving grids delivers an intrinsically better accuracy than its first-order counterpart, where the moving viscous fluxes are evaluated on the current mesh configuration. However, for the application presented herein, the contributions of the viscous terms to the lift and drag coefficients are not sufficiently important—their contribution to the lift coefficient is 40 times smaller than that of the pressure—to impact the accuracy of the total lift and drag solutions.

In order to support the last of the above three conclusions, the authors consider another flow problem around a NACA0012 airfoil forced into the same harmonic pitching motion as in the previous example, but where the Reynolds number is $Re = 5$. Clearly, such a problem is more of an academic nature than the previous one. However, it does illustrate a situation where the viscous terms provide more important contributions to lift and drag than in the previous example. The computational domain around this airfoil is descretized using an unstructured mesh with a unit chord $L_C = 1$ and 3114 vertices only. The distance between two grid points in the boundary layer is typically $d = 2.6 \times 10^{-2}$. The numerical Reynolds number associated with this mesh is

$$Re^{\text{num}} = \frac{M_\infty \times L_C}{d} = \frac{0.85 \times 1.0}{2.6 \times 10^{-2}} = 32.692 > Re,$$

which implies that this mesh is suitable for laminar viscous flow computations at a Reynolds number $Re = 5$. Two simulations are performed using first $\Delta t = T/25$, and then $\Delta t = T/250$,

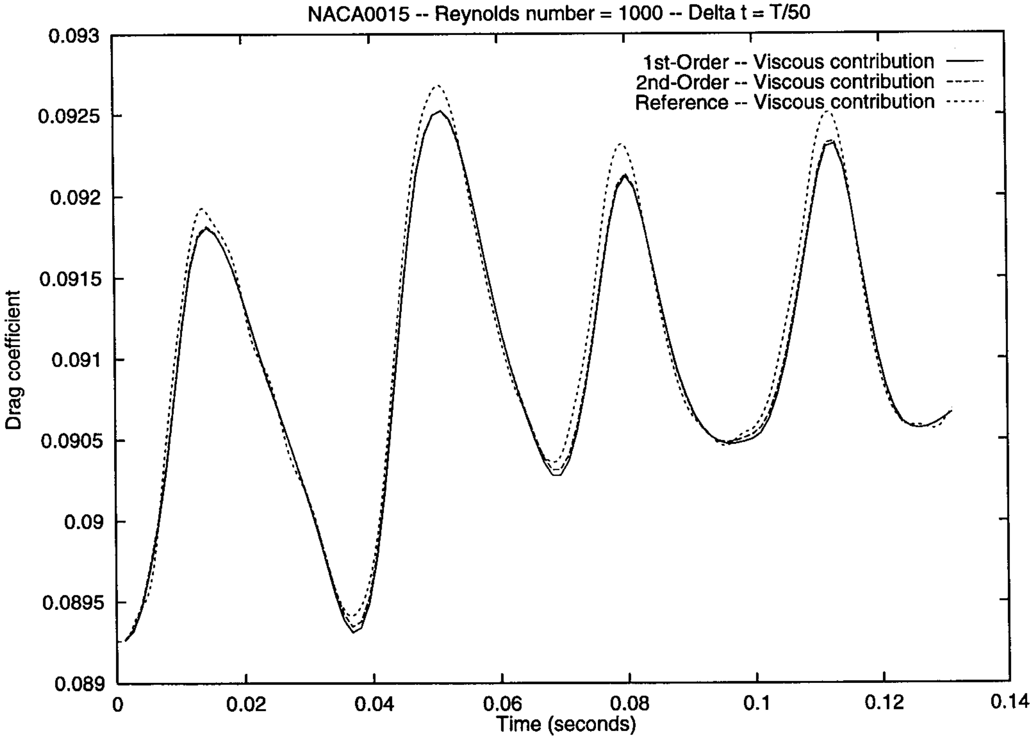

Figure 14. Evolution of the viscous contribution to the drag coefficient for $\Delta t = T/50$ and $Re = 10^3$.

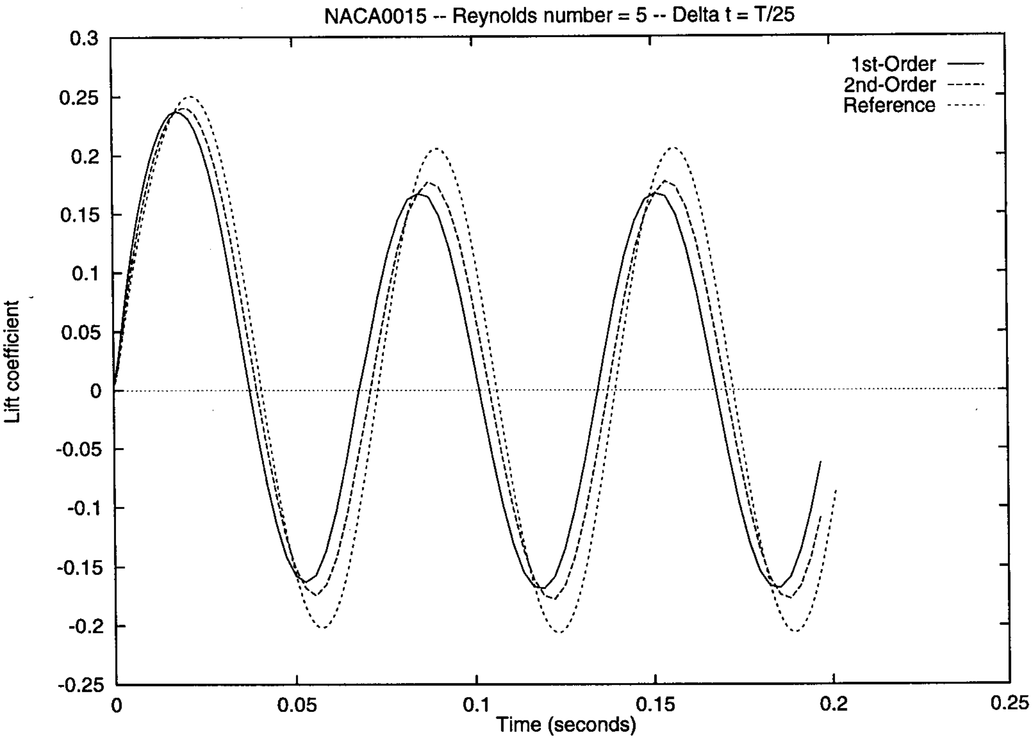

Figure 15. Evolution of the lift coefficient for $\Delta t = T/25$ and $Re = 5$.

and the predicted lift and drag time histories are reported in Figures 15 and 16. For the same reason as before, the authors refer to the flow solution produced by the second simulation as the 'reference' solution.

For this low Reynolds number application, Figure 15 shows that, for the fixed time step $\Delta t = T/25$, the second-order strategy (21) for computing the moving diffusive fluxes reduces the amplitude error by 33% and the phase error obtained for the total lift by 50% when these fluxes are evaluated by the first-order computational strategy. This observation highlights the importance of the choice of the mesh configuration for time integrating the semi-discrete viscous fluxes, when these fluxes contribute significantly to the sought after flow solution.

5. CONCLUSIONS

Many computational fluid dynamics unsteady applications require the discretization of the Navier–Stokes equations on unstructured dynamic meshes. Advancing the flow solution between time t^n and time t^{n+1} on these grids raises the question of where to evaluate the convective and diffusive fluxes: on the mesh configuration at (t^n, X^n), on that at (t^{n+1}, X^{n+1}), in between these two configurations, outside these two configurations, or on combinations of these and other configurations? For each numerical scheme designed for the solution of unsteady flow problems on fixed grids, a discrete geometric conservation law (DGCL) can be derived to ensure that the extension of this scheme to moving grids preserves the state of a uniform flow. Enforcing the DGCL can be used as a guideline for answering the restriction of

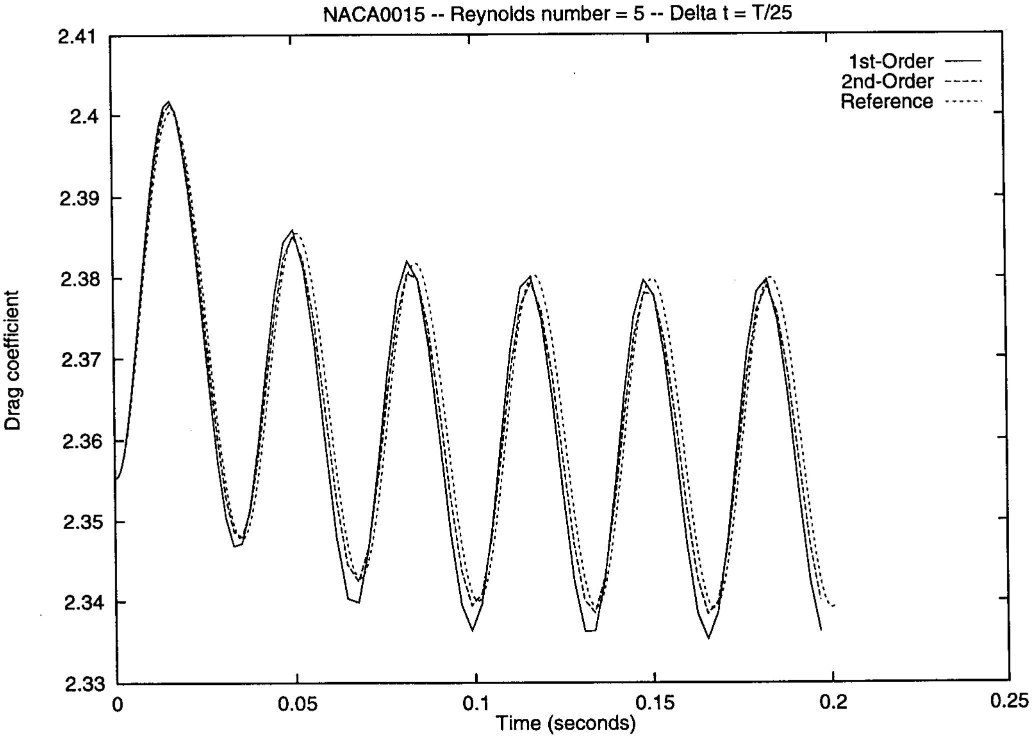

Figure 16. Evolution of the drag coefficient for $\Delta t = T/25$ and $Re = 5$.

the above question to the convective fluxes. For example, the DGCL associated with a three-dimensional flow solver based on the finite volume method and the second-order implicit backward difference formula is satisfied when the integration of the convective fluxes between t^n and t^{n+1} is evaluated by a four-point quadrature rule recalled in this paper. This four-point rule involves four different mesh configurations and was shown to preserve the second-order accuracy of the implicit backward difference formula on dynamic meshes. The DGCL cannot be used as a guideline for evaluating the diffusive fluxes on dynamic meshes because these fluxes vanish for a uniform flow. On the other hand, this paper has shown that a local truncation error analysis can be used as a guideline for addressing the issue of where to evaluate the diffusive fluxes when time integrating between t^n and t^{n+1} the semi-discrete Navier–Stokes equations. More specifically, it has been shown that the popular strategy consisting of computing these fluxes on the current mesh configuration (t^n, X^n) is a first-order computational strategy, while computing these fluxes on the midpoint configuration $(t^{n+1/2}, X^{n+1/2})$ is a second-order one. In that sense, the extension to dynamic meshes of a three-dimensional flow solver based on the finite volume method and the second-order implicit backward difference formula calls for the evaluation of the moving convective fluxes by a four-point quadrature rule involving four distinct mesh configurations, and the evaluation of the moving diffusive fluxes by a one-point rule involving the midpoint configuration. Sample aerodynamic applications seem to confirm that evaluating the moving diffusive fluxes on the midpoint configuration is an intrinsically more time-accurate procedure than evaluating these fluxes on the current mesh configuration. However, practice also shows that when the viscous effects are not sufficiently important, computing the moving diffusive fluxes on the current configuration does not reduce the overall time accuracy of the sought after flow solution, as long as the moving convective fluxes are properly evaluated, i.e. by a procedure that obeys the governing DGCL.

ACKNOWLEDGMENTS

The authors acknowledge partial support by the National Science Foundation under Grant ASC-9217394, and partial support by the Air Force Office of Scientific Research under Grant F49620-97-1-0059. They also thank the two anonymous reviewers for their constructive and helpful criticisms.

REFERENCES

1. J. Donea, 'An arbitrary Lagrangian–Eulerian finite element method for transient fluid–structure interactions', *Comput. Methods Appl. Mech. Eng.*, **33**, 689–723 (1982).
2. C. Farhat, M. Lesoinne and N. Maman, 'Mixed explicit/implicit time integration of coupled aeroelastic problems: three-field formulation, geometric conservation and distributed solution', *Int. J. Numer. Methods Fluids*, **21**, 807–835 (1995).
3. J.T. Batina, 'Unsteady Euler airfoil solutions using unstructured dynamic meshes', *AIAA Paper 89-0115*, *AIAA 27th Aerospace Sciences Meeting*, Reno, NA, January 9–12, 1989.
4. O.A. Kandil and H.A. Chuang, 'Unsteady vortex-dominated flows around manoeuvring wings over a wide range of Mach numbers', *AIAA Paper No. 88-0317*, *AIAA 26th Aerospace Sciences Meeting*, Reno, NA, January 11–14, 1988.
5. C. Farhat and T.Y. Lin, 'Transient aeroelastic computations using multiple moving frames of reference', *AIAA Paper No. 90-3053*, *AIAA 8th Applied Aerodynamics Conference*, Portland, OR, August 20–22, 1990.
6. T. Tezduyar, M. Behr and J. Liou, 'A new strategy for finite element computations involving moving boundaries and interfaces—the deforming spatial domain/space–time procedure: I. The concept and the preliminary numerical tests', *Comput. Methods Appl. Mech. Eng.*, **94**, 339–351 (1992).
7. A. Masud, 'A space–time finite element method for fluid structure interaction', *Ph.D. Thesis*, Stanford University, 1993.
8. M. Lesoinne and C. Farhat, 'Geometric conservation laws for aeroelastic computations using unstructured dynamic meshes', *AIAA Paper 95-1709*, *12th AIAA Computational Fluid Dynamics Conference*, San Diego, CA, June 19–22, 1995.
9. M. Lesoinne and C. Farhat, 'Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations', *Comput. Methods Appl. Mech. Eng.*, **134**, 71–90 (1996).
10. P.D. Thomas and C.K. Lombard, 'Geometric conservation law and its application to flow computations on moving grids', *AIAA J.*, **17**, 1030–1037 (1979).
11. W.K. Anderson, J.L. Thomas and C.L. Rumsey, 'Extension and application of flux–vector splitting to unsteady calculations on dynamic meshes', *AIAA Paper 87-1152-CP*, 1987.
12. H. Zhang, M. Reggio, J.Y. Trépanier and R. Camarero, 'Discrete form of the GCL for moving meshes and its implementation in CFD schemes', *Comput. Fluids*, **22**, 9–23 (1992).
13. B.N. Konga and H. Guillard, 'Godunov type method on non-structured meshes for three-dimensional moving boundary problems', *Comp. Methods Appl. Mech. Eng.*, **113**, 183–204 (1994).
14. B. Koobus and C. Farhat, 'Second-order time-accurate and geometrically conservative implicit schemes for flow computations on unstructured dynamic meshes', *Comput. Methods Appl. Mech. Eng.* (1999) in press. Also published as *AIAA Paper 98-0113*, *56th Aerospace Sciences Meeting and Exhibit*, Reno, NA, January 12–15, 1998.
15. L. Fezoui and B. Stoufflet, 'A class of implicit upwind schemes for Euler simulations with unstructured meshes', *J. Comput. Phys.*, **84**, 174–206 (1989).
16. C. Farhat, L. Fezoui and S. Lantéri, 'Two-dimensional viscous flow computations on the connection machine: unstructured meshes, upwind schemes, and massively parallel computations', *Comput. Methods Appl. Mech. Eng.*, **102**, 61–88 (1993).
17. C. Farhat and S. Lantéri, 'Simulation of compressible viscous flows on a variety of MPPs: computational algorithms for unstructured dynamic meshes and performance results', *Comput. Methods Appl. Mech. Eng.*, **119**, 35–60 (1994).
18. H. Guillard and C. Farhat, 'On the significance of the geometric conservation law for flow computations on moving meshes', *AIAA Paper 99-0793*, *37th Aerospace Sciences Meeting*, Reno, NV, January 11–14, 1999.
19. C. Farhat, C. Degand, B. Koobus and M. Lesoinne, 'Torsional springs for two-dimensional dynamic unstructured fluid meshes', *Comput. Methods Appl. Mech. Eng.*, **163**, 231–245 (1998). Also published as *AIAA Paper 98-2070*, *S9th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Long Beach, CA, April 20–23, 1998.
20. P.L. Roe, 'Approximate Riemann solvers, parameter vectors and difference schemes', *J. Comput. Phys.*, **43**, 357–371 (1981).
21. B. Van Leer, 'Towards the ultimate conservative difference scheme. V: A second-order sequel to Goudonov's method', *J. Comput. Phys.*, **32**, 361–370 (1979).