INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2003; **43**:253–279 (DOI: 10.1002/fld.606)

An implicit edge-based ALE method for the incompressible Navier–Stokes equations

Richard W. Smith[1,*,†] and Jeffrey A. Wright[2]

[1]*Code R11, Coastal Systems Station, Naval Surface Warfare Center, Panama City, FL 32407, U.S.A.*
[2]*Streamline Numerics Inc., Gainesville, FL 32605, U.S.A.*

SUMMARY

A new finite volume method for the incompressible Navier–Stokes equations, expressed in arbitrary Lagrangian–Eulerian (ALE) form, is presented. The method uses a staggered storage arrangement for the pressure and velocity variables and adopts an edge-based data structure and assembly procedure which is valid for arbitrary *n*-sided polygonal meshes. Edge formulas are presented for assembling the ALE form of the momentum and pressure equations. An implicit multi-stage time integrator is constructed that is geometrically conservative to the precision of the arithmetic used in the computation. The method is shown to be second-order-accurate in time and space for general time-dependent polygonal meshes. The method is first evaluated using several well-known unsteady incompressible Navier–Stokes problems before being applied to a periodically forced aeroelastic problem and a transient free surface problem. Published in 2003 by John Wiley & Sons, Ltd.

KEY WORDS: unstructured ALE methods; unsteady incompressible Navier–Stokes equations

1. INTRODUCTION

Over the last decade or so, one class of problems in computational fluid dynamics that has undergone substantial development is the class where the fluid domain boundary is either explicitly time-dependent or is unknown *a priori* and determined, in a coupled fashion, as part of an unsteady flow solution. Free surface, fluid-structure interaction, and forced boundary motion flows are typical of problems in this class. A natural way to formulate moving boundary problems is the so-called arbitrary Lagrangian–Eulerian (ALE) form of the fundamental conservation laws where the domain boundary and interior control surfaces are allowed to move arbitrarily in time and which recovers the Eulerian and Lagrangian forms as special limiting cases of the general ALE form [1–3].

*Correspondence to: R. W. Smith, Coastal Systems Station, Code R11, Naval Surface Warfare Center, Panama City, FL 32407-7001, U.S.A.
†E-mail: SmithRW@ncsc.navy.mil

Contract/grant sponsor: Office of Naval Research

Received 2 October 2002
 Revised 27 May 2003

In addition to the usual considerations associated with implementing an incompressible flow algorithm (e.g. the particulars of the pressure-velocity coupling procedure, the suppression of a checkerboard pressure field, etc.) the presence of a time-dependent mesh introduces a number of additional considerations that need to be addressed. Arguably, the most basic of these is a purely geometric requirement that, in order for the discrete form to be conservative in time, the volume increments swept out by a moving control surface must be computed in a way that is consistent with the time integration of the conservation laws governing the flow. This requirement for a consistent prescription for the time-dependent geometric quantities in a finite volume method has been formalized by many researchers [4–6] and is commonly referred to as the geometric conservation law (GCL).

Here, a recently developed edge-based finite volume method [7] for the incompressible Navier–Stokes equations is revised and extended to admit a time-dependent mesh. An implicit multi-stage time integrator is constructed that is GCL compliant to the precision of the arithmetic used in the computation and is shown to be second-order-accurate in time for an arbitrarily deforming mesh. As in Reference [7], the use of a staggered grid system eliminates the need for *ad hoc* modifications to suppress a checkerboard pressure field as well as the need to prescribe a boundary condition for pressure. Other aspects of the basic method on fixed grids, as well as a brief discussion of other related edge-based finite volume methods can be found in Reference [7]. Here, the method is first evaluated using several well-known unsteady incompressible Navier–Stokes problems before being applied to a periodically forced aeroelastic problem and a transient free surface problem.

2. BASIC CONSERVATION LAWS

The equations governing incompressible fluid dynamics stated in weak co-ordinate-free form for a time-dependent finite volume $\Omega(t)$ with boundary $\Gamma(t)$ are given by

$$\frac{\partial}{\partial t}\int_{\Omega(t)} \rho\mathbf{v}\,\mathrm{d}\Omega + \oint_{\Gamma(t)} [\rho\mathbf{v}(\mathbf{v}-\mathbf{s})\cdot\mathbf{n} - \mathbf{n}\cdot\mathbf{T}]\,\mathrm{d}\Gamma + \int_{\Omega(t)} \mathbf{B}\,\mathrm{d}\Omega = 0 \tag{1}$$

and

$$\oint_{\Gamma(t)} [\mathbf{v}\cdot\mathbf{n}]\,\mathrm{d}\Gamma = 0 \tag{2}$$

Adopting the incompressible form of the Navier–Poisson law as the constitutive relation for the fluid gives

$$\mathbf{T} = -p\mathbf{I} + \boldsymbol{\tau} \tag{3}$$

where $\boldsymbol{\tau}$ is the viscous stress tensor given by

$$\boldsymbol{\tau} = \mu(\nabla\mathbf{v} + \mathbf{v}\nabla) \tag{4}$$

In Equations (3) and (4), $\mathbf{T}$ and $\mathbf{I}$ are the stress and identity tensors, respectively, and $\mathbf{v}\nabla$ is the transpose of $\nabla\mathbf{v}$. In Equation (1), $\mathbf{v}$ is the flow velocity, $\mathbf{n}$ and $\mathbf{s}$ are the outward unit normal and control surface velocity vectors, respectively, and $\mathbf{B}$ is the vector of volume sources. The variables ϱ, μ, and p denote the fluid density and viscosity, and flow pressure, respectively. In

the case of turbulent flow, all flow variables take on their usual Reynolds-averaged definitions with the fluid viscosity interpreted as the flow eddy-viscosity. Appropriate initial and boundary conditions for the velocity field complete the specification of the incompressible flow problem.

In Equation (1) $\mathbf{s}=\mathbf{v}$ corresponds to the Lagrangian view of conservation, whereas $\mathbf{s}=\mathbf{0}$ corresponds to the Eulerian view. Because of the generality, or in other words, arbitrariness of the description offered by Equation (1) it is often referred to as the arbitrary Lagrangian–Eulerian (ALE) form of the Navier–Stokes equations.

3. NUMERICAL METHOD

3.1. Data structures and assembly

Equation (1) is discretized in space using an edge-based finite volume method that results in a strictly conservative form that is valid for arbitrary n-sided time-dependent polygonal control volumes. Figure 1 illustrates a typical mixed-element mesh comprised of quadrilateral and triangular control volumes. In addition to the main mesh, a dual mesh is constructed by connecting the median point of each main edge sharing a common vertex with the centroids of the adjoining main control volumes as shown in the figure. This system of main and median dual control volumes forms a staggered grid arrangement where the velocity components are stored at the vertices of the main mesh and pressure is stored at the centroids of the main control volumes as indicated.

At this point in the history of the development of numerical methods for incompressible flows, the use of a staggered mesh may appear, at least initially, to be somewhat quaint, and indeed, some algorithmic complexity is associated with a staggered grid arrangement

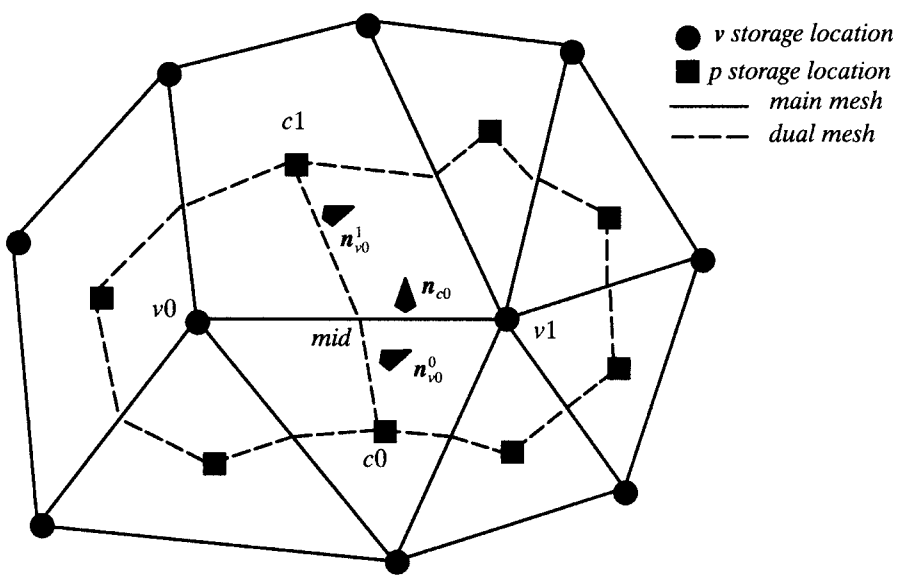

Figure 1. Connectivity convention and storage locations for edge-based discretization.

that can be avoided by simply collocating all flow variables. However, the use of staggered control volumes for evaluating the momentum and pressure equation fluxes eliminates the need for a pressure boundary condition for incompressible Navier–Stokes problems with Dirichlet boundary conditions. Furthermore, the use of staggered control volumes eliminates the need to modify the numerical fluxes in order to suppress a checkerboard pressure field, which is admitted by collocated finite volume and finite difference schemes. It has been our experience that the benefits of a staggered mesh easily make up for any additional algorithmic complexity or computational overhead incurred by the approach.

The spatial residuals of mass and momentum are evaluated by projecting the dependent variables in Equations (1) and (2) onto a Cartesian basis and performing a counter-clockwise integration around the contour surrounding the main and dual control volumes, respectively. The former integration is the basis for the assembly of a purely elliptic equation that is independent of time, which, in conjunction with the momentum equations, governs the evolution of the pressure field.

The integration of the mass and momentum residuals is facilitated by forming a list of main edges comprising the tessellation with endpoints denoted $v0$ and $v1$ for each edge in the list. Each main edge in the tessellation shares exactly two main control volumes with centroids denoted by $c0$ and $c1$ which are arranged by construction such that $c0$ lies to the right of the directed line from $v0$ and $v1$ as shown in Figure 1. This connectivity convention allows the instantaneous outward normal vector for the main control volume edges to be given by

$$\left.\begin{aligned} \mathbf{n}_{c0} &= -(y_{v1} - y_{v0})\mathbf{i} + (x_{v1} - x_{v0})\mathbf{j} \\ \mathbf{n}_{c1} &= -\mathbf{n}_{c0} \end{aligned}\right\} \text{main edge} \tag{5a}$$

As shown in the figure, each median dual control volume edge has two facets with scaled normal vectors $\mathbf{n}^0_{v0}$ and $\mathbf{n}^1_{v0}$ oriented outward from dual control volume $v0$, and $\mathbf{n}^0_{v1}$ and $\mathbf{n}^1_{v1}$ oriented outward from dual control volume $v1$. The normal vector for each dual edge facet is given by

$$\left.\begin{aligned} \mathbf{n}^0_{v0} &= (y_{\text{mid}} - y_{c0})\mathbf{i} - (x_{\text{mid}} - x_{c0})\mathbf{j} \\ \mathbf{n}^0_{v1} &= -\mathbf{n}^0_{v0} \end{aligned}\right\} \text{dual edge facet 0} \tag{5b}$$

$$\left.\begin{aligned} \mathbf{n}^1_{v0} &= (y_{c1} - y_{\text{mid}})\mathbf{i} - (x_{c1} - x_{\text{mid}})\mathbf{j} \\ \mathbf{n}^1_{v1} &= -\mathbf{n}^1_{v0} \end{aligned}\right\} \text{dual edge facet 1} \tag{5c}$$

For notational brevity, the superscripts denoting the two dual edge facets in Equations (5b) and (5c) will be deleted in all future references to the dual edge normal vectors with the understanding that in all expressions involving a dual edge normal both facets are included.

Since a staggered variable arrangement is used in the present method, three distinct data structures are used to facilitate the computation of the control surface fluxes and assembly of the discrete equations for the velocity and pressure fields. The first of these is an edge data structure which stores the node and adjoining element numbers, the normal vectors, and

fluxes associated with each main and dual edge. The second of these is a node data structure storing the fluid velocity vector, velocity gradient tensor, pseudo velocity vector, dual control volume area, source term, as well as node and neighbor node coefficients associated with the momentum equations. Finally, an element data structure is defined to store the node and neighbour node coefficients as well as the source term for the pressure equation.

An edge-based procedure is used to assemble the discrete form of the conservation laws for general, *n*-sided polygonal meshes. The first step in assembling the discrete linearized equations is a sweep over edges where edge formulas are used to compute the individual flux components associated with each main and dual edge. The edge flux data, expressed in the form of node and neighbour node coefficients and source term, are then assigned to the two nodes of the edge. Finally, a sweep over edges is performed which, using edge connectivity information, sums the edge nodal data into global node and global element data. After the global equations for velocity and pressure have been assembled, they are solved in a segregated fashion using Jacobi iteration. A detailed prescription of the overall solution strategy for unsteady flows with moving boundaries is given in Section 3.6.

3.2. Time integration of the momentum equations

Integrating the momentum equations over the dual control volumes leads to the semi-discrete form

$$\frac{\mathrm{d}\mathbf{W}}{\mathrm{d}t} + \mathbf{R} = 0 \tag{6a}$$

where $\mathbf{W} = \rho\Omega\mathbf{v}$ is the momentum vector and the spatial momentum residuals are given by

$$\mathbf{R} = \sum [\rho\mathbf{v}(\mathbf{v} - \mathbf{s}) \cdot \mathbf{n} + p\mathbf{n} - \mathbf{n} \cdot \tau]_e + \mathbf{B}\Omega \tag{6b}$$

and the summation is taken over all edge facets forming a closed dual control volume. A single-step diagonally implicit Runge–Kutta (DIRK) method is used to advance the velocity field according to the two-stage formula [8]

$$\mathbf{W}^{(1)} = \mathbf{W}^n - \Delta t(\beta_{10}\mathbf{R}^n + \beta_{11}\mathbf{R}^{(1)}) \tag{7a}$$

$$\mathbf{W}^{n+1} = \mathbf{W}^n - \Delta t(\beta_{20}\mathbf{R}^n + \beta_{21}\mathbf{R}^{(1)} + \beta_{22}\mathbf{R}^{n+1}) \tag{7b}$$

where the following set of weights yield a formally second-order-accurate method when Equation (6a) is a set of linear equations.

$$\beta_{10} = 0.0,\ \beta_{11} = 0.2551,\ \beta_{20} = -0.0545,\ \beta_{21} = 0.7545,\ \beta_{22} = 0.3$$

This two-stage method is self-starting, can accommodate grid motion on a stage by stage basis and requires storage of the spatial residuals at only two time levels. Furthermore, the first-order implicit Euler method can be easily recovered as a special case of the two-stage method by executing only the first stage with β_{11} set to unity.

Since the velocity field is advanced in time implicitly, a system of algebraic equations is formed for the momentum equations at each outer iteration of each time step. Furthermore,

since Jacobi iteration is used to solve the system of equations, the spatial momentum residuals in Equation (6b) are cast in the form:

$$\mathbf{R} = A\mathbf{v} - \mathbf{b} \tag{8}$$

where A is the diagonal coefficient of $\mathbf{v}$ and $\mathbf{b}$ contains source terms as well as off-diagonal neighbour contributions to the spatial residual.

Using the notation of Equation (8), the first stage of the DIRK method which advances the velocity from t^n to $t^{(1)}$ is governed by the non-linear system of algebraic equations

$$\left[\left(\frac{\rho\Omega}{\Delta t\beta_{11}} + A\right)\mathbf{v} - \mathbf{b}\right]^{(1)} = \left[\frac{\rho\mathbf{v}\Omega}{\Delta t\beta_{11}}\right]^{n} \tag{9a}$$

Similarly, the second stage of the Runge–Kutta method which advances the velocity from $t^{(1)}$ to t^{n+1} is governed by the non-linear system

$$\left[\left(\frac{\rho\Omega}{\Delta t\beta_{22}} + A\right)\mathbf{v} - \mathbf{b}\right]^{n+1} = \left[\frac{\rho\mathbf{v}\Omega}{\Delta t\beta_{22}} - \frac{\beta_{20}}{\beta_{22}}(A\mathbf{v} - \mathbf{b})\right]^{n} - \left[\frac{\beta_{21}}{\beta_{22}}(A\mathbf{v} - \mathbf{b})\right]^{(1)} \tag{9b}$$

Equation (9a) and (9b) are linearized by forming the coefficients using the most recent values of $\mathbf{v}$ and p computed in the outer loop of the iterative solution procedure. The flow pressure, which appears in Equation (6) through (9), is not obtained by a time marching procedure such as Equation (6) or (7). Rather, a pressure equation is solved in a segregated fashion along with the solution for the velocity field within the iterative procedure. The details of the pressure equation are developed in Section 3.5 and the overall solution procedure for unsteady incompressible flow with moving boundaries is described in Section 3.6.

3.3. Geometric conservation

Under certain assumptions regarding the flow field, Equation (1) reduces to a purely geometric statement relating the time-dependent control volume $\Omega(t)$, the control surface $\Gamma(t)$, the control surface velocity $\mathbf{s}(t)$ and the unit normal $\mathbf{n}(t)$. In the context of finite volume methods the consistent treatment of these time-dependent geometric quantities is often referred to as the geometric conservation law (GCL) which may be regarded as an identity that must be satisfied, either explicitly or implicitly, if the conservative property is to be maintained [4–6]. Here, the geometric identity is satisfied implicitly at each stage of time integration by adopting the 'time-averaged normals form' of the GCL [5].

In the present two-stage ALE method, mesh movement is, by construction, limited to the first stage of time integration so that mesh co-ordinates need only be stored at two discrete time levels. More precisely, during the first stage, $x^{(1)}$ is a state variable and is updated using the same implicit procedure as for the pressure and velocity variables. During the second stage, $x^{(n+1)}$ is held constant and equal to $x^{(1)}$. However, even with mesh movement limited to the first stage of time integration, the two-stage DIRK scheme requires a consistent GCL compliant prescription for the control surface velocity and unit normal vectors for both stages of time integration since the vector of conserved variables appears differenced across the full time step in the second stage formula, Equation (9b).

When the GCL compliant control surface velocity and normal vectors are adopted in the assembly of the convective flux for both stages of the DIRK time integrator, the resulting

method is geometrically conservative to the precision of the arithmetic used in the calculation and is second-order-accurate in time on an arbitrary time-dependent mesh.

Although the compressible form of the mass conservation equation is typically used to derive the GCL, the momentum equations, Equation (1), can be used with equal facility to derive the basic identity under appropriate restrictions on the flow variables. Assuming uniform velocity, pressure and density, and the absence of a momentum source term, Equation (1), in semi-discrete form becomes

$$\frac{\mathrm{d}\Omega}{\mathrm{d}t} - \sum(\mathbf{s}\cdot\mathbf{n})_e = 0 \tag{10}$$

where the summation is taken over all edge facets forming a closed dual control volume. Integrating Equation (10) over the first stage of the DIRK time integrator gives

$$\Omega^{(1)} - \Omega^n - \int_t^{t+\beta_{11}\Delta t} \sum(\mathbf{s}\cdot\mathbf{n})_e \,\mathrm{d}t = 0 \tag{11}$$

Equation (11) is a statement of geometric conservation in discrete form and is an identity which must be satisfied if the conservative property in time is to be respected. Here, the identity is satisfied exactly by defining the scaled normals $\mathbf{n}(t)$ as a weighted average of the t^n to $t^{(1)}$ time level areas.

For polygonal control volumes, the integrand of Equation (11) is a linear function of time and therefore the integral can be computed exactly by sampling the integrand once at the midpoint of the time interval Δt. Evaluating Equation (11) at the midpoint of the first stage gives without approximation

$$\Omega^{(1)} - \Omega^n - \beta_{11}\Delta t \sum(\mathbf{s}\cdot\mathbf{n})_e^{(1/2)} = 0 \tag{12}$$

Following Zhang *et al.* [5], it can be easily verified that the identity, Equation (12), is satisfied exactly when the following time-averaged definitions are adopted for the dual edge velocity, $\mathbf{s}^{(1/2)}$, and dual edge normal vector, $\mathbf{n}^{(1/2)}$, when assembling the spatial residual $\mathbf{R}^{(1)}$ for that stage.

$$\mathbf{s}^{(1/2)} = \frac{1}{2}\left(\frac{\mathbf{x}^{(1)} - \mathbf{x}^n)_{n0}}{\beta_{11}\Delta t} + \frac{\mathbf{x}^{(1)} - \mathbf{x}^n)_{n1}}{\beta_{11}\Delta t}\right) \tag{13a}$$

$$\mathbf{n}^{(1/2)} = \frac{1}{2}(\mathbf{n}^{(1)} + \mathbf{n}^n) \tag{13b}$$

Similarly, the discrete form of the GCL will be satisfied identically during the second stage if the definitions given in Equation (13) are used in the evaluation of the convective fluxes when assembling the spatial residual $\mathbf{R}^{(1)}$ for that stage but with β_{11} replaced by β_{21}.

Since the GCL prescribes the consistent evaluation of volume increments associated with grid motion the time-averaged normals given in Equation (13b) should only be used to evaluate the convective flux at a control surface. The pressure and viscous fluxes should be evaluated using the current mesh configuration, i.e. either $x^{(1)}$ or $x^{(n+1)}$, rather than the time-averaged normal vector given in Equation (9a). To that end, the time-averaged edge normal and edge velocity vectors, Equation (13), are notated explicitly in the edge flux formulas that

are presented in the following sections. Again, as a notational convenience, all other geometric quantities appearing without a superscript are taken to be at the current mesh configuration. Finally, it may be noted that in the absence of grid motion the mid-interval definitions given in Equation (13) recover their appropriate fixed grid meaning.

3.4. Flux evaluation for the momentum equations

Recalling Equations (8) and (9), the assembly procedure for the linearized momentum equations reduces to evaluating A and $\mathbf{b}$ for each velocity component at each vertex in the tessellation. During the assembly procedure each edge makes contributions to A and $\mathbf{b}$ with valid expressions for the momentum conservation available only after all dual edge facets in the tessellation have contributed and all dual control volumes have been closed. Per-edge contributions to the diagonal coefficient and source term, denoted by A_e and $\mathbf{b}_e$, respectively, are given below for the convective, viscous and pressure fluxes forming the steady-state momentum balance.

3.4.1. Convective flux. The momentum flux at a dual edge

$$[\rho \mathbf{v}(\mathbf{v}-\mathbf{s})\cdot\mathbf{n}]_e = [\mathbf{v}f]_e \tag{14}$$

is evaluated using the trapezoidal rule in conjunction with a linear reconstruction [7, 9, 10] of the velocity based on the upwind value of $\mathbf{v}$ and it's gradient $\mathbf{v}\nabla$. Sampling the edge mass flux, f_e, at the edge midpoint yields

$$f_e = \rho\left(\frac{1}{2}(\mathbf{v}_{c0}+\mathbf{v}_{c1}) - \mathbf{s}^{(1/2)}\right)\cdot\mathbf{n}_{v0}^{1/2} \tag{15}$$

where the cell-centred velocities, $\mathbf{v}_{c0}$ and $\mathbf{v}_{c1}$ are taken to be the average of the main mesh vertex values surrounding $c0$ and $c1$.

Linearly reconstructing the edge velocity, $\mathbf{v}_e$, at the edge midpoint using upwind data yields

$$\text{if } f_e>0 \ \mathbf{v}_e = \mathbf{v}_{v0} + (\mathbf{v}\nabla)_{v0}\cdot\mathbf{r}_{v0} \quad \text{else } \mathbf{v}_e = \mathbf{v}_{v1} + (\mathbf{v}\nabla)_{v1}\cdot\mathbf{r}_{v1} \tag{16}$$

where $\mathbf{r}$ is the vector from the main mesh vertex to the mid-edge sample point [7]. Recalling that $\mathbf{n}_{v1} = -\mathbf{n}_{v0}$, the following per-edge contributions to the convective flux are obtained.

Assembling at $v0$:

$$\text{if } f_e>0 \begin{cases} A_e = f_e \quad \text{else} \\ \mathbf{b}_e = -f_e((\mathbf{v}\nabla)_{v0}\cdot r_{v0}) \end{cases} \quad \begin{cases} A_e = 0 \\ \mathbf{b}_e = -f_e(\mathbf{v}_{v1} + (\mathbf{v}\nabla)_{v1}\cdot r_{v1}) \end{cases} \tag{17a}$$

Assembling at $v1$:

$$\text{if } f_e>0 \begin{cases} A_e = 0 \quad \text{else} \\ \mathbf{b}_e = f_e(\mathbf{v}_{v0} + (\mathbf{v}\nabla)_{v0}\cdot r_{v0}) \end{cases} \quad \begin{cases} A_e = -f_e \\ \mathbf{b}_e = f_e((\mathbf{v}\nabla)_{v1}\cdot \mathbf{r}_{v1}) \end{cases} \tag{17b}$$

The velocity gradient tensor, $\mathbf{v}\nabla$, appearing in Equations (16) and (17) is computed using Green's Theorem over the median dual control volume using the edge-based procedure described in Reference [7].

3.4.2. Viscous flux. The viscous stress vector at a dual edge

$$-[\mathbf{n}\cdot\boldsymbol{\tau}]_e \tag{18}$$

can be written for constant viscosity, divergence-free conditions as

$$-[\mathbf{n}\cdot\boldsymbol{\tau}]_e = -\mu[\mathbf{v}\nabla\cdot\mathbf{n}]_e \tag{19}$$

where the Navier–Poisson law has been used to relate $\boldsymbol{\tau}$ to $\mathbf{v}\nabla$. Again, Green's Theorem is used to evaluate the components of $\mathbf{v}\nabla$ over the edge area Ω_e, shown in Figure 2. After some manipulation and again recalling that $\mathbf{n}_{v1} = -\mathbf{n}_{v0}$ the following per-edge contributions to the viscous flux are obtained.
Assembling at $v0$:

$$\begin{aligned} A_e &= \frac{\mu}{2\Omega_e}(\mathbf{n}_{v0}\cdot\mathbf{n}_{v0}) \\ \mathbf{b}_e &= \frac{\mu}{2\Omega_e}(\mathbf{n}_{v0}\cdot\mathbf{n}_{v0})\mathbf{v}_{v1} - \frac{\mu}{2\Omega_e}(\mathbf{v}_{c0}-\mathbf{v}_{c1})(\mathbf{n}_{c0}\cdot\mathbf{n}_{v0}) \end{aligned} \tag{20a}$$

Assembling at $v1$:

$$\begin{aligned} A_e &= \frac{\mu}{2\Omega_e}(\mathbf{n}_{v0}\cdot\mathbf{n}_{v0}) \\ \mathbf{b}_e &= \frac{\mu}{2\Omega_e}(\mathbf{n}_{v0}\cdot\mathbf{n}_{v0})\mathbf{v}_{v0} + \frac{\mu}{2\Omega_e}(\mathbf{v}_{c0}-\mathbf{v}_{c1})(\mathbf{n}_{c0}\cdot\mathbf{n}_{v0}) \end{aligned} \tag{20b}$$

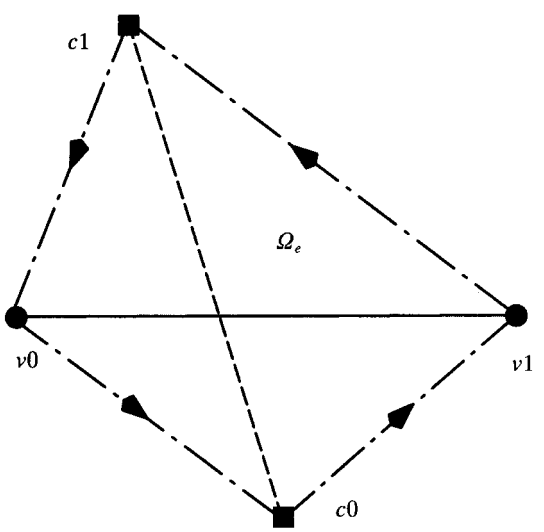

Figure 2. Integration path used for evaluating $\mathbf{v}\nabla$ for viscous fluxes and ∇p for pressure equation.

3.4.3. Pressure flux. The pressure flux at a dual edge

$$[p\mathbf{n}]_e \tag{21}$$

is evaluated using the trapezoidal rule. Again, recalling that $\mathbf{n}_{v1} = -\mathbf{n}_{v0}$ the following per-edge contributions to the pressure flux are obtained.
Assembling at $v0$:

$$\mathbf{b}_e = -\frac{1}{2}(p_{co} + p_{c1})\mathbf{n}_{v0} \tag{22a}$$

Assembling at $v1$:

$$\mathbf{b}_e = \frac{1}{2}(p_{co} + p_{c1})\mathbf{n}_{v0} \tag{22b}$$

Equations such as (17), (20), and (22), which express the per-edge contributions to the flux balance in a finite volume method are often referred to as edge formulas.

3.5. Flux evaluation for the pressure equation

For incompressible flow, conservation of mass is a purely kinematic constraint requiring the velocity field to be divergence-free. Since this constraint is instantaneous, the discrete form of the pressure equation should also be instantaneous and not contain artifacts of the flow history. To clarify this point consider the momentum equations in co-ordinate-free divergence form:

$$\frac{\partial}{\partial t}(\rho\mathbf{v}) + \nabla\cdot(\rho\mathbf{v}\mathbf{v}) = -\nabla p + \mu\nabla^2\mathbf{v} + \mathbf{B} \tag{23}$$

and continuity

$$\nabla\cdot\mathbf{v} = 0 \tag{24}$$

The convective term $\rho\mathbf{v}\mathbf{v}$ is a tensor product and can be rewritten as

$$\nabla\cdot(\rho\mathbf{v}\mathbf{v}) = \rho(\mathbf{v}\cdot\nabla)\mathbf{v} + \mathbf{v}(\nabla\cdot\rho\mathbf{v}) \tag{25}$$

For a uniform density flow the second term on the right in Equation (25) is zero by continuity. Consequently, the momentum equations can be rewritten as

$$\frac{\partial}{\partial t}(\rho\mathbf{v}) + (\rho\mathbf{v}\cdot\nabla)\mathbf{v} = -\nabla p + \mu\nabla^2\mathbf{v} + \mathbf{B} \tag{26}$$

Taking the divergence of Equation (26) gives

$$\nabla\cdot\left(\frac{\partial}{\partial t}(\rho\mathbf{v}) + (\rho\mathbf{v}\cdot\nabla)\mathbf{v}\right) = \nabla\cdot(-\nabla p + \mu\nabla^2\mathbf{v} + \mathbf{B}) \tag{27}$$

Switching the order of differentiation in the unsteady and viscous terms and invoking continuity once more reveals the pressure equation in co-ordinate-free form to be

$$\nabla^2 p = -\nabla\cdot(\rho\mathbf{v}\cdot\nabla)\mathbf{v} + \nabla\cdot\mathbf{B} \tag{28}$$

The pressure equation, Equation (28), when coupled with the momentum equations, governs the time evolution of the pressure field. As can be seen from Equation (28), the pressure equation is purely elliptic, of Poisson type, contains no flow history and is inviscid.

We now derive the finite volume form of Equation (28) from the finite volume form of the steady-state momentum equation and continuity. In accordance with Equation (28), the final discrete form of the pressure equation will contain no artifacts of the flow history. However, for reasons that will become clear in the following development, the viscous term is retained in the finite volume form of Equation (28).

In finite volume form, Equation (2), expressing mass conservation, is given by

$$\sum(\mathbf{v}\cdot\mathbf{n})_e = 0 \tag{29}$$

where the summation is taken over all edges forming a closed main control volume. In order to facilitate the construction of the discrete form of the pressure equation, it is convenient to rewrite the discrete steady-state form of the momentum equations, Equation (8), as

$$\mathbf{v} = \hat{\mathbf{v}} - \frac{1}{A}\left(\frac{1}{2}(p_{c0} + p_{c1})\mathbf{n}\right) \tag{30a}$$

where the nodal pseudo-velocity vector, $\hat{\mathbf{v}}$, is given by

$$\hat{\mathbf{v}} = \frac{\bar{\mathbf{b}}}{A} \tag{30b}$$

and where $\bar{\mathbf{b}}$ is the flux source term excluding the pressure flux, but including all other flux contributions as well as any volume sources. It is emphasized that A and $\bar{\mathbf{b}}$ appearing in Equation (30) are assembled from the steady state form of the momentum equations and therefore are entirely independent of flow history. This is consistent with Equation (28). However, the viscous flux contribution is retained in the assembly of A and $\bar{\mathbf{b}}$ as they appear in Equation (30). Although this is inconsistent with Equation (28), experience has shown that by retaining the viscous contribution in the momentum equation the resulting pressure equation coefficients, to be presented in Equation (35), are guaranteed to remain positive even under pathological conditions such as zero fluid velocity. If the viscous contribution is not retained in the construction of the pressure equation coefficients, this guarantee of positivity is lost.

The inconsistency between Equation (28), which is inviscid, and Equation (30), which contains viscous contribution, is proportional to the degree that mass is conserved over a main control volume. If continuity is satisfied to machine precision the inconsistency vanishes. Although, the present method does not conserve mass to machine precision, numerical experiments have indicated that the mass error is higher-order and has no significant impact on the accuracy or stability of the method [7].

Recalling that mass conservation, Equation (29), is enforced as a sum over edges for a main control volume, a main edge velocity, $\mathbf{v}_e$, is proposed to facilitate coupling the pressure and velocity fields. Appealing to the definition of the nodal pseudo-velocity defined in Equation (30), $\mathbf{v}_e$ is taken to be of the form:

$$\mathbf{v}_e = \hat{\mathbf{v}}_e - \left(\frac{1}{A}\right)_e \int_{\Omega_e} (\nabla p)_e \, \mathrm{d}\Omega_e \tag{31}$$

where $\hat{\mathbf{v}}_e$ is the main edge pseudo-velocity vector. Taking ∇p to be a constant over Ω_e allows the integral in Equation (31) to be evaluated using Green's Theorem over the edge area,

Ω_e, shown in Figure 2, with the contour integral evaluated by the trapezoidal rule. After some manipulation, the integral in Equation (31) can be written succinctly in terms of the instantaneous main and dual control volume normal vectors, $\mathbf{n}_{c0}$ and $\mathbf{n}_{v0}$, respectively. The resulting relation is

$$\int_{\Omega_e} (\nabla p)\, \mathrm{d}\Omega_e = \frac{1}{2}((p_{c1} - p_{c0})\mathbf{n}_{c0} + (p_{v1} - p_{v0})\mathbf{n}_{v0}) \tag{32}$$

Substituting Equation (32) into Equation (31) yields the following instantaneous pressure-velocity coupling on a per-edge basis.

$$\mathbf{v}_e = \hat{\mathbf{v}}_e - \left(\frac{1}{2A}\right)_e ((p_{c1} - p_{c0})\mathbf{n}_{c0} + (p_{v1} - p_{v0})\mathbf{n}_{v0}) \tag{33}$$

Substituting this edge velocity into the mass conservation equation, Equation (29), gives the finite volume form of the pressure equation

$$\sum \frac{1}{2A_e}((p_{c1} - p_{c0})\mathbf{n}_{c0} \cdot \mathbf{n}_{c0} + (p_{v1} - p_{v0})\mathbf{n}_{c0} \cdot \mathbf{n}_{v0}) - (\hat{\mathbf{v}}_e \cdot \mathbf{n}_{c0}) = 0 \tag{34}$$

where, again, the summation is taken over all edges forming a closed main control volume.

Recalling that all discrete equations are ultimately written in the form of Equation (8) and solved by Jacobi iteration, the assembly procedure for the pressure equation reduces to evaluating A and b at each cell centre in the domain. During the assembly procedure each main edge makes contributions to A and b with valid expressions for the pressure equation available only after all main edges in the tessellation have contributed and all main control volumes have been closed. Per-edge contributions to the diagonal coefficient and source term, denoted by A_e and b_e, respectively, are given below for the pressure equation fluxes.
Assembling at $c0$:

$$\begin{aligned} A_e &= \left(\frac{\mathbf{n}_{c0} \cdot \mathbf{n}_{c0}}{2A_e}\right) \\ b_e &= \left(\frac{\mathbf{n}_{c0} \cdot \mathbf{n}_{c0}}{2A_e}\right) p_{c1} - \left(\frac{\mathbf{n}_{v0} \cdot \mathbf{n}_{c0}}{2A_e}\right)(p_{v0} - p_{v1}) - (\hat{\mathbf{v}}_e \cdot \mathbf{n}_{c0}) \end{aligned} \tag{35a}$$

Assembling at $c1$:

$$\begin{aligned} A_e &= \left(\frac{\mathbf{n}_{c0} \cdot \mathbf{n}_{c0}}{2A_e}\right) \\ b_e &= \left(\frac{\mathbf{n}_{c0} \cdot \mathbf{n}_{c0}}{2A_e}\right) p_{c0} + \left(\frac{\mathbf{n}_{v0} \cdot \mathbf{n}_{c0}}{2A_e}\right)(p_{v0} - p_{v1}) - (\hat{\mathbf{v}}_e \cdot \mathbf{n}_{c0}) \end{aligned} \tag{35b}$$

The edge values appearing in Equation (35) are interpolated from main mesh vertex data as

$$\frac{A_e}{\Omega_e} = \frac{1}{2}\left(\frac{A_{n0}}{\Omega_{n0}} + \frac{A_{n1}}{\Omega_{n1}}\right) \tag{36a}$$

$$\hat{\mathbf{v}}_e = \frac{1}{2}(\hat{\mathbf{v}}_{n0} + \hat{\mathbf{v}}_{n1}) \tag{36b}$$

where the area weighted interpolation appearing in Equation (36a) properly accounts for the different control volume areas involved in the interpolation.

For an orthogonal quadrilateral mesh the second term appearing in the expression for b_e is identically zero. For general mesh topologies, the magnitude of the product of the main and dual edge normal vectors, $\mathbf{n}_{c0} \cdot \mathbf{n}_{v0}$, serves as an indicator of the orthogonality of a triangular, hybrid and other general polygonal mesh. Here, the vertex centred pressure values, p_{v0} and p_{v1} are taken as the average of the cell centred values for all main control volumes sharing a common vertex.

As noted previously, the use of a staggered storage arrangement for the pressure and velocity variables eliminates the need for a prescription of boundary and initial conditions for pressure. Accordingly, the decomposition implied by Equation (30) is not performed for an edge on the domain boundary since no pressure-velocity coupling is sought across boundary edges. Consequently, when the pressure equation is assembled for control volumes adjacent to the domain boundary, the physical velocity flux leaving the domain through a boundary edge, $(\mathbf{v} \cdot \mathbf{n})_e$, appears naturally in place of the pseudo-velocity flux $(\hat{\mathbf{v}} \cdot \mathbf{n})_e$ appearing in Equation (35). Therefore, the global mass flux

$$\oint_{\Gamma(t)} \mathbf{v} \cdot \mathbf{n} \, \mathrm{d}\Gamma \tag{37}$$

naturally appears in the discrete equation for pressure. This is consistent with the basic conservation laws, where the global mass flux must be zero in order to satisfy the solvability constraint associated with the incompressible Navier–Stokes equations [7].

3.6. Overall algorithm

The overall solution procedure for an unsteady moving grid problem can now be summarized as follows.

Initialize flow velocity and grid coordinate state at t^n, *i.e.*, $(\mathbf{v}, \mathbf{x})^n$.
Compute temporal source terms for first and second stages of time integration.
Solve first stage of time integrator.
  Move grid to $\mathbf{x}^{(1)}$.†
  Compute spatial momentum residuals.
  Compute $\hat{\mathbf{v}}^{(1)}$.
  Compute $p^{(1)}$ *using Jacobi point-iteration.*
  Compute $\mathbf{v}^{(1)}$ *using Jacobi point-iteration.*
  Return to † *until convergence criteria is met.*
State is now $(\mathbf{v}, p, \mathbf{x})^{(1)}$; *begin next stage.*
Add temporal source term at $t^{(1)}$.
Solve second stage of time integrator.
  Copy grid at $\mathbf{x}^{(1)}$ *to* $\mathbf{x}^{n+1}$.
  Compute spatial momentum residuals‡.
  Compute $\hat{\mathbf{v}}^{n+1}$.
  Compute p^{n+1} using Jacobi point-iteration.
  Compute $\mathbf{v}^{n+1}$ using Jacobi point-iteration.
  Return to ‡ *until convergence criteria is met.*
State is now $(\mathbf{v}, p, \mathbf{x})^{n+1}$; *go to next time-step.*

The following special cases of the general ALE algorithm are noted:

(i) For problems using a fixed grid the grid movement step is skipped.
(ii) For first-order time integration execute only the first stage with β_{11} set to unity.
(iii) For steady-state problems execute only the first stage for one time-step with Δt set to infinity.

Since the momentum and pressure equations are solved iteratively in a sequential fashion, underrelaxation is incorporated into the algorithm in order to retard changes in the solution (principally, in the velocity field) from iteration to iteration in the outer solution loop. Relaxation is implemented according to

$$\frac{A}{\alpha}\mathbf{v} = b + (1-a)\frac{A}{\alpha}\mathbf{v}^* \quad \text{and} \quad \frac{A}{\alpha}p = b + (1-\alpha)\frac{A}{\alpha}p^* \tag{38}$$

where $\mathbf{v}^*$ and p^* are the stored values of velocity and pressure from the previous outer iteration and α is the relaxation coefficient. Typical values of α for the momentum equations range from 0.2 for complex turbulent flows to 0.8 for simple laminar flows. The value of α for the pressure equation is typically 0.95 for all flows.

4. NUMERICAL RESULTS

For steady-state problems on fixed orthogonal grids the method presented here degenerates exactly to the steady formulation in Reference [7] and results for those problems will not be repeated here. For unsteady problems even with a fixed grid the present method is somewhat different from that of [7] and the performance of the current method for these problems is evaluated using Taylor flow and the impulsively started flow over a circular cylinder. Next, an example demonstrating the GCL compliance of the DIRK time integrator is given followed by an order of accuracy demonstration on a time-dependent mesh. Finally, the method is applied to a periodically forced aeroelastic problem and a transient free surface problem.

4.1. Taylor decaying vortex flow on a fixed mesh

In order to verify the order of accuracy of the present method, decaying vortex flow is solved using the present method on a square domain of dimension $\pi/2$. The exact solution, owing to Taylor [11], in non-dimensional form is given by

$$u = -\cos x \sin y \, \mathrm{e}^{-2t/Re} \tag{39a}$$

$$v = \sin x \cos y \, \mathrm{e}^{-2t/Re} \tag{39b}$$

$$p = -\frac{1}{4}(\cos 2x + \cos 2y)\mathrm{e}^{-4t/Re} \tag{39c}$$

and the following L_1 measure of the error norm is adopted

$$e_1 = \frac{1}{N}\sum_{i=1}^{N} |\phi_{\text{computed}} - \phi_{\text{exact}}| \tag{40}$$

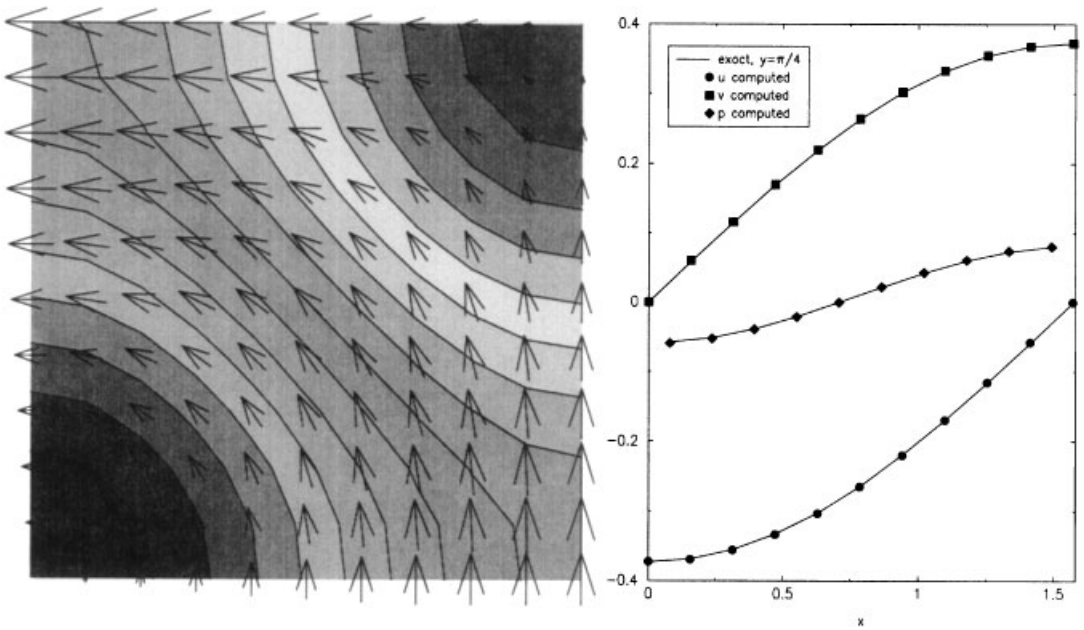

Figure 3. Computed velocity vectors, pressure contours and comparison with exact solution for decaying vortex flow, $Re = 100$, $t = 32.0$, on a fixed 10×10 uniform quadrilateral mesh using DIRK time integration, $\Delta t = 32.0$.

where ϕ is a generic dependent variable (i.e. u, v or p) and N is the number of nodes where the solution is computed. Figure 3 shows the computed solution for the decaying vortex problem on a fixed, 10×10 uniform quadrilateral mesh at $Re = 100$ after one time step. The solution shown in the figure corresponds to the time when the velocity field has decayed to half its initial value. The agreement with the exact solution is also shown in the figure and is quite good. Figure 4 shows the convergence of the method with mesh and time-step refinement for a family of fixed uniform quadrilateral meshes and a series of time steps at $Re = 100$. The method may be seen to be second-order-accurate in time and space for this problem on a fixed grid.

4.2. Impulsively started flow over a circular cylinder

The computed length of the recirculation region in the wake of an impulsively started circular cylinder at $Re = 40$ is shown in Figure 5. Computations were done using a fixed 200×60 quadrilateral O-grid with the outer domain boundary located twenty diameters away from the cylinder where free-stream conditions were imposed on the velocity field.

Time step independent results were obtained using both the first-order Euler implicit and the second-order DIRK method and are shown in the figure along with the experimental data of Coutanceau and Bouard [12]. Agreement with the experimental data is quite good using either the first- or second-order method once a time-step independent solution is obtained. Time-step independence is achieved with $\Delta t = 0.2$ and 0.02 for the second- and first-order

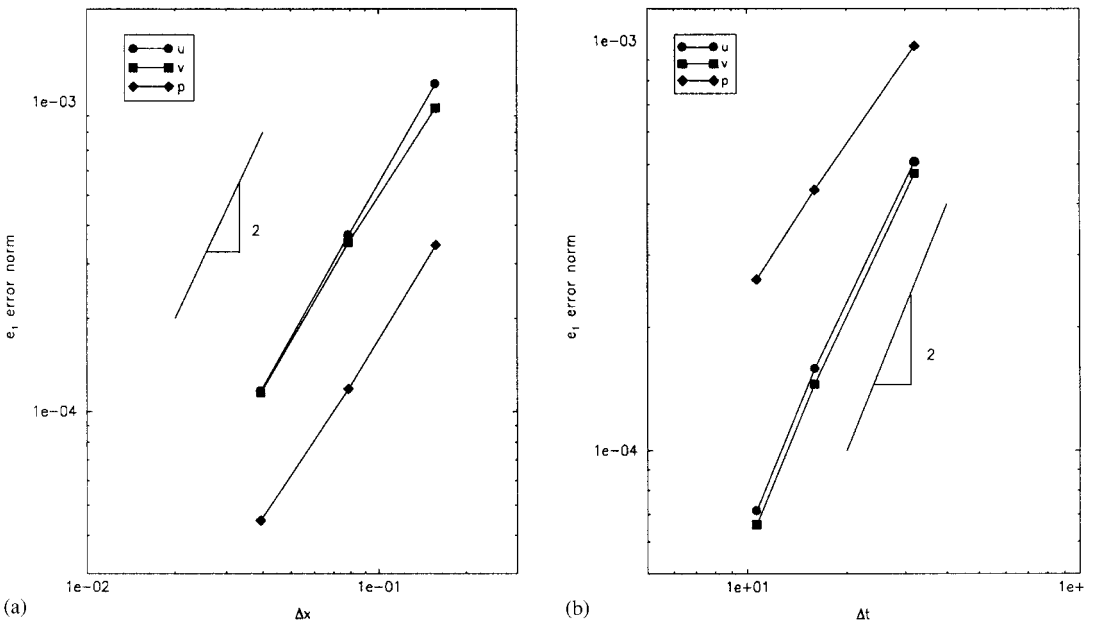

Figure 4. Error norms for decaying vortex flow at $Re = 100$ and $t = 32.0$: (a) For a family of fixed uniform quadrilateral meshes using DIRK time integration, $\Delta t = 0.5$; (b) For a series of time steps using DIRK time integration on a fixed 80×80 uniform quadrilateral mesh.

methods, respectively. Also shown in the figure is the number of outer iterations required per time step to reduce the residuals to machine zero using 32 bit arithmetic. Although the DIRK method requires two to three times the number of outer iterations compared to the Euler method, the DIRK time integrator was approximately three times faster in run time compared to the Euler method due to the greatly reduced number of time steps required to achieve time step independence.

4.3. GCL compliance

Figure 6 shows the initial condition and the computed solution for uniform flow after one time step using the DIRK time integrator with severe deformation of the grid. The initial condition is a uniform, diagonally inclined flow with the same velocity field imposed as a boundary condition for all subsequent times. The pressure gradient is shown in the contour plot at $t = t + \Delta t$ and is shown to be zero to the precision of the 32 bit arithmetic used in the calculation. Identical results were obtained using the Euler time integrator and consequently the algorithm is seen to be GCL compliant using either the first- or second-order time integration methods.

4.4. Taylor decaying vortex flow on a deforming mesh

Figure 7 shows the initial condition and the computed solution for decaying vortex flow after one time step using the DIRK time integrator with severe random deformation of the grid.

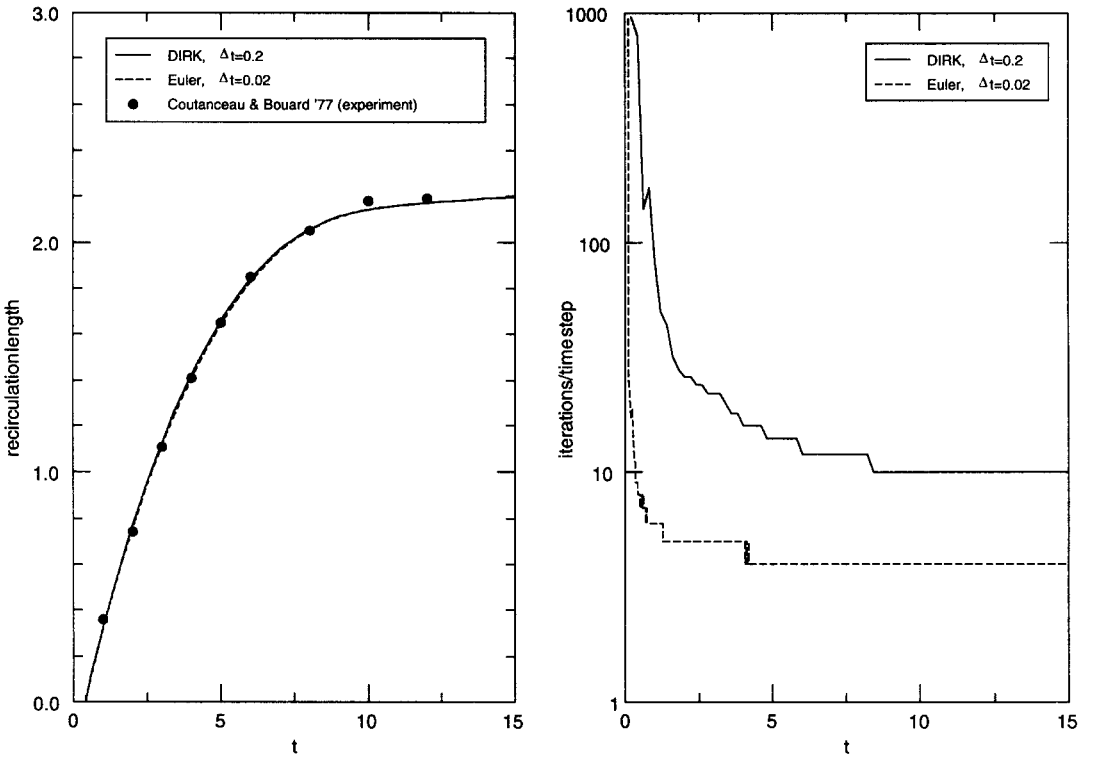

Figure 5. Wake recirculation length and convergence performance for an impulsively started circular cylinder, $Re = 40$, 200×60 fixed quadrilateral O-grid using DIRK and Euler time integration.

The temporal order-of-accuracy of the method on a moving grid is also shown in the figure and is compared with the order-of-accuracy for the same problem on a fixed grid taken from Figure 4. The method is shown to be second-order-accurate in time for the moving grid calculation with essentially the same absolute error as for the fixed grid case. The accuracy shown for this problem as well as previous demonstration problems suggests that the present ALE method can be applied with confidence to the more complex moving boundary problems that follow.

4.5. Impulsively started hydrofoil with free surface

A more complex problem using a surface-fitted time-dependent mesh is that of an impulsively started hydrofoil submerged below a free surface. The parameters of the simulation were chosen to coincide with the experiment of Duncan [13]. The hydrofoil is at an angle of attack of 5.0°, submerged 1.03 chord lengths below the free surface. The Froude number, based on chord length, is 0.567, and the Reynolds number was taken to be one million in accordance with other numerical studies of the same problem [14, 15]. In what follows, edge formulas are developed for assembling the kinematic free surface boundary condition and the computed steady-state wave elevation is compared with other simulation results before the method is

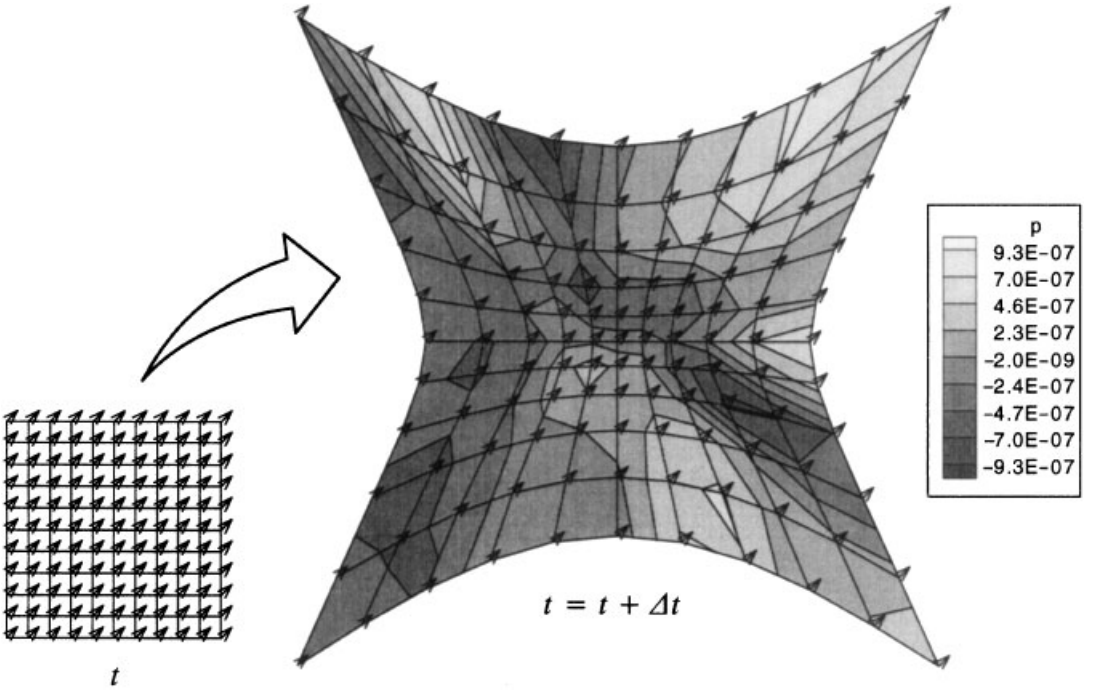

Figure 6. Uniform initial conditions and computed solution at $t = t + \Delta t$ showing isobars after one time step using DIRK time integration with a time-dependent mesh.

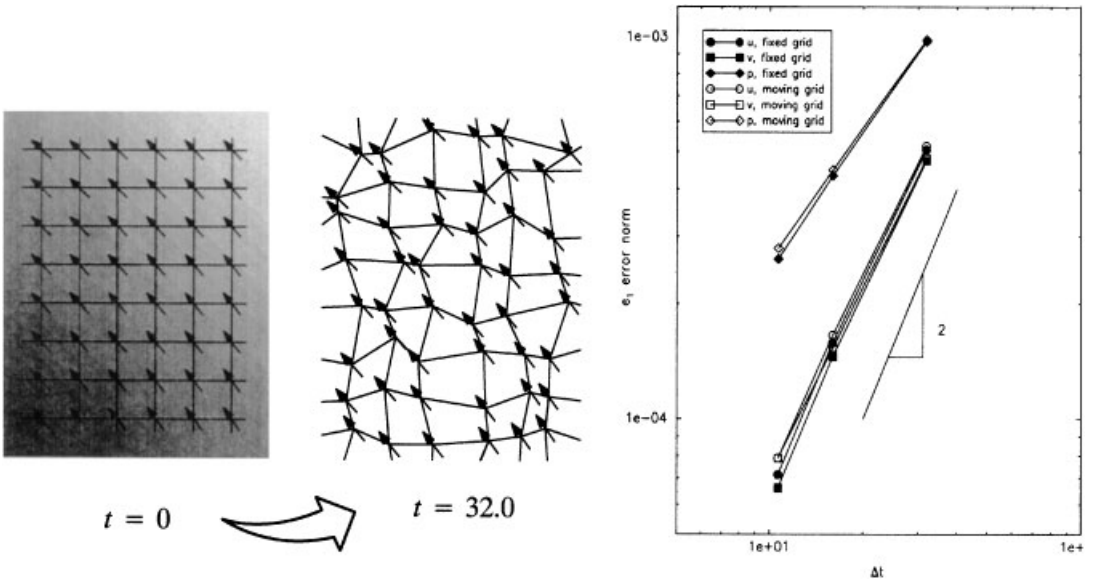

Figure 7. Error norms for decaying vortex flow at $Re = 100$ and $t = 32.0$ using DIRK time integration on a time-dependent 80×80 quadrilateral mesh. Typical portions of the mesh are shown at $t = 0.0$ and 32.0 after a random distortion of the mesh.

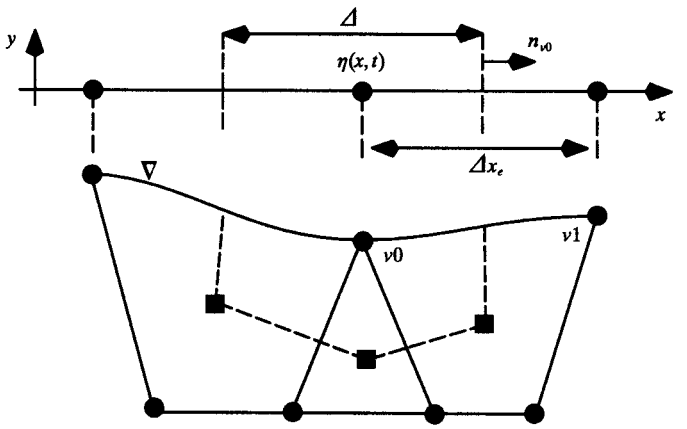

Figure 8. Control volumes for discretizing the dynamic and kinematic free surface boundary conditions.

applied to the unsteady, impulsively started problem. Details of the present free surface boundary condition implementation and numerical results can be found in Reference [16].

In the case of two-dimensional gravity-driven flow the vector of momentum sources becomes

$$\mathbf{B} = [0, \rho g]^{\mathrm{T}} \tag{41}$$

where gravity is assumed to act in the negative y direction of a fixed Cartesian co-ordinate system as shown in Figure 8. The gravitational body force term can be conveniently incorporated into the Navier–Stokes equations by adopting the piezometric pressure, $\tilde{p}$, given by

$$\tilde{p} = p + \rho g y \tag{42}$$

where y is measured with respect to an arbitrary datum. The stress tensor, Equation (3), can then be redefined as

$$\mathbf{T} = -\tilde{p} I + \tau \tag{43}$$

Without a loss of generality, the pressure at a free surface can be set to zero for incompressible flows and the isotropic component of the dynamic free surface boundary condition can be written in terms of the piezometric pressure as

$$\tilde{p} = \rho g \eta(x,t) \quad \text{on } y = \eta(x,t) \tag{44}$$

where $\eta(x,t)$ is measured with respect the undisturbed level of the free surface. For this condition to be satisfied exactly without linearization it must be applied on the instantaneous free surface defined by $\eta(x,t)$.

For two-dimension flow the kinematic free surface boundary condition is given by

$$\frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} - v = 0 \quad \text{on} \quad y = \eta(x,t) \tag{45}$$

where u and v are the Cartesian components of the fluid velocity vector [17]. Again, for this condition is to be satisfied exactly without linearization it must be applied on the instantaneous free surface defined by $\eta(x,t)$.

Integrating Equation (45) over the one-dimensional vertex-centred control volume shown in Figure 8 gives the following finite volume form:

$$\frac{\partial}{\partial t}\int_{\Delta} \eta \, \mathrm{d}x + \int_{\Delta} u\frac{\partial \eta}{\partial x}\,\mathrm{d}x - \int_{\Delta}(v - \eta D(x))\,\mathrm{d}x = 0 \tag{46}$$

where Δ is the length of the free surface node control volume. A damping term has been included in Equation (46) to annihilate outgoing waves as they approach the boundary of the computational domain. Again, details of the implementation can be found in Reference [16].

As shown in Figure 8, the one-dimensional grid used to discretize Equation (46) is simply the projection of the free surface boundary edges used by the two-dimensional flow solver onto a constant y plane. Since $\eta(x,t)$ is measured in the y direction, the evolution of the free surface elevation does not affect the x co-ordinate of the free surface nodes. Consequently, the finite volume form of the free surface elevation equation, Equation (46), does not involve grid movement and the time dependency of the control volume used to discretize the free surface elevation equation has been dropped.

Integrating Equation (46) over the one-dimensional free surface control volumes leads to the semi-discrete form

$$\frac{\mathrm{d}\eta}{\mathrm{d}t} + R = 0 \tag{47a}$$

where spatial residual of the free surface elevation equation is given by

$$R = \sum [u\eta n_{v0}]_e - (v - \eta D(x))\Delta \tag{47b}$$

Here, n_{v0} is the x component of the dual edge unit normal vector defined outward to $v0$ as shown in Figure 8, and the summation is over the two free surface edges sharing a common node. Following the notation and solution procedure adopted for the momentum equations, the spatial residual of the free surface elevation equation is cast in the form:

$$R = A\eta - b \tag{48}$$

where A is the diagonal coefficient of η and b contains true source terms, as well as off-diagonal neighbour contributions to the flux balance. Using an edge-based assembly procedure, A and b are assembled by a sweep over all free surface edges with a valid system of linearized equations available only after both adjoining edges have contributed to A and b and all free surface node control volumes have been closed.

Using upwind biased linear reconstruction to evaluate η at a free surface dual edge gives

$$\text{if } f_e > 0 \ \eta_e = \eta_{v0} + \left(\frac{\mathrm{d}\eta}{\mathrm{d}x}\right)_{v0}\left(\frac{\Delta x_e}{2}\right) \quad \text{else } \eta_e = \eta_{v1} - \left(\frac{\mathrm{d}\eta}{\mathrm{d}x}\right)_{v1}\left(\frac{\Delta x_e}{2}\right) \tag{49}$$

where the dual edge flux is given by

$$f_e = un_{v0} \tag{50}$$

with u and v taken to be piecewise constant over Δ. Following the notation used for the assembly of the momentum equations the per-edge contributions to A and b are given below for the convective contribution to Equation (48).
Assembling at $v0$:

$$\text{if } f_e > 0 \begin{cases} A_e = f_e \\ b_e = -f_e\left(\dfrac{\mathrm{d}\eta}{\mathrm{d}x}\right)_{v0}\left(\dfrac{\Delta x_e}{2}\right) \end{cases} \quad \text{else} \begin{cases} A_e = 0 \\ b_e = -f_e\left(\eta_{v1} - \left(\dfrac{\mathrm{d}\eta}{\mathrm{d}x}\right)_{v1}\left(\dfrac{\Delta x_e}{2}\right)\right) \end{cases} \tag{51a}$$

Assembling at $v1$:

$$\text{if } f_e > 0 \begin{cases} A_e = 0 \\ b_e = -f_e\left(\eta_{v0} + \left(\dfrac{\mathrm{d}\eta}{\mathrm{d}x}\right)_{v0}\left(\dfrac{\Delta x_e}{2}\right)\right) \end{cases} \quad \text{else} \begin{cases} A_e = -f_e \\ b_e = -f_e\left(\dfrac{\mathrm{d}\eta}{\mathrm{d}x}\right)_{v1}\left(\dfrac{\Delta x_e}{2}\right) \end{cases} \tag{51b}$$

The nodal derivative in Equation (51) and control volume length in Equation (47) may also be computed by a sweep over free surface edges with the per-edge contributions given by the following.

$$\text{Assembling at } v0 \text{ and } v1: \quad \frac{\mathrm{d}\eta}{\mathrm{d}x} = \frac{1}{2}\left(\frac{\eta_{v1} - \eta_{v0}}{\Delta x_e}\right) \text{ and } \Delta = \left(\frac{\Delta x_e}{2}\right) \tag{52}$$

Finally, adding the volume source to b by a sweep over free surface nodes completes the assembly of the spatial residual in Equation (47). The free surface elevation equation, Equation (47), is advanced in time using the same implicit two-stage integrator used for the momentum and turbulence closure equations and is simply imbedded within the sequential solution strategy previously described for the basic system of flow equations.

A typical numerical result for the steady-state problem is shown in Figure 9. In the figure, the hydrofoil, it's viscous wake, and the free surface elevation are clearly evident. The SST $k-\omega$ based turbulence model [18, 19] was used to prescribe the eddy-viscosity in the calculation and a simple algebraic grid mover is adopted to co-ordinate the movement of interior nodes [16]. A close up of a typical mixed-element mesh near the leading edge of the hydrofoil is shown in Figure 10. In Figure 11 the free surface elevation computed with the present method is compared with the numerical results of Lohner *et al.* [15] and Hino [14] and the experimental data of Duncan [13]. It can be seen that all three methods give results that are similar.

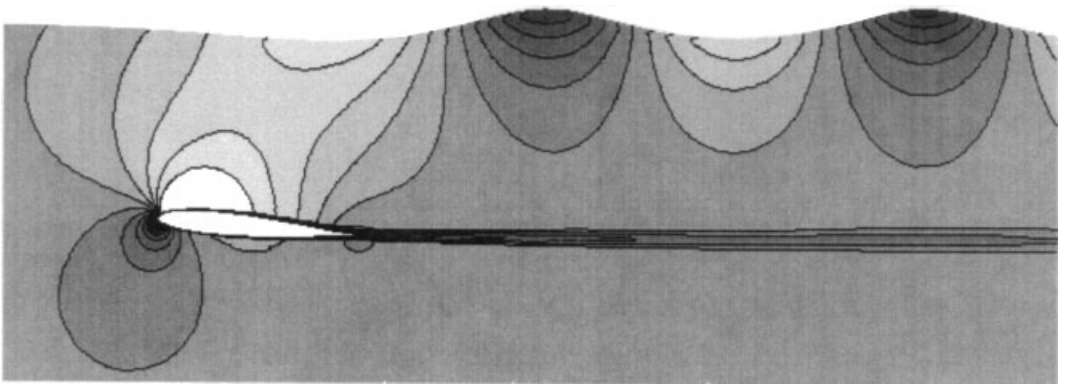

Figure 9. Computed steady-state free surface elevation and velocity contours for NACA0012 hydrofoil: $Fr = 0.567$ and $Re = 10^6$.

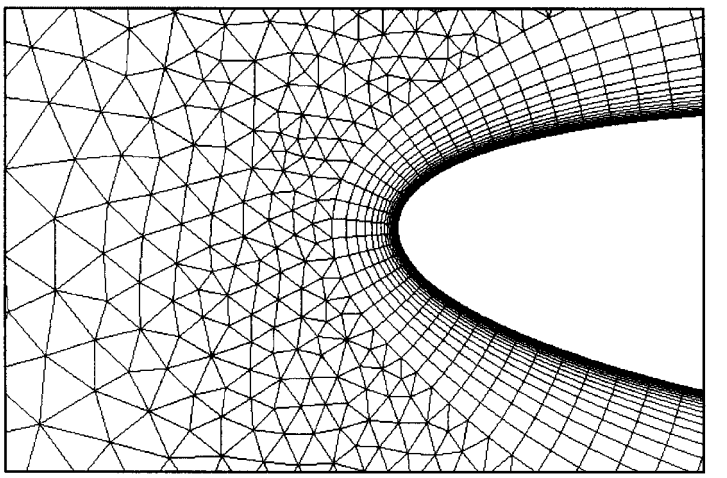

Figure 10. Mixed-element mesh for submerged NACA0012 hydrofoil: $Fr = 0.567$ and $Re = 10^6$.

The transient, impulsively started problem is quite different from the steady-state problem discussed above. In the case of steady flow, all transient perturbations have departed the domain, leaving a wave system that is stationary in the reference frame of the hydrofoil. In the impulsively started case, however, waves of all lengths and phase speeds are allowed to exist, some of which are long wavelength rapidly moving waves that propagate both upstream and downstream. The presence of a long wavelength surge can be seen moving upstream in Figure 12, as well as the shorter length waves characteristic of the steady-state wave system developing downstream. Details of the calculation and a mesh refinement study can be found in Reference [16] and will not be repeated here.

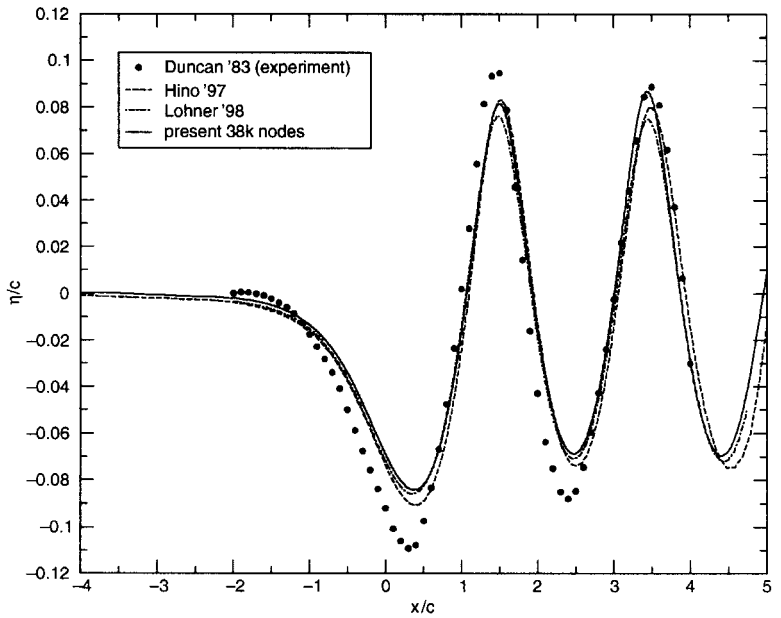

Figure 11. Comparison of computed steady state free surface elevations with experimental data for submerged NACA0012 hydrofoil: $Fr = 0.567$, $Re = 10^6$ and $s = 1.03$.

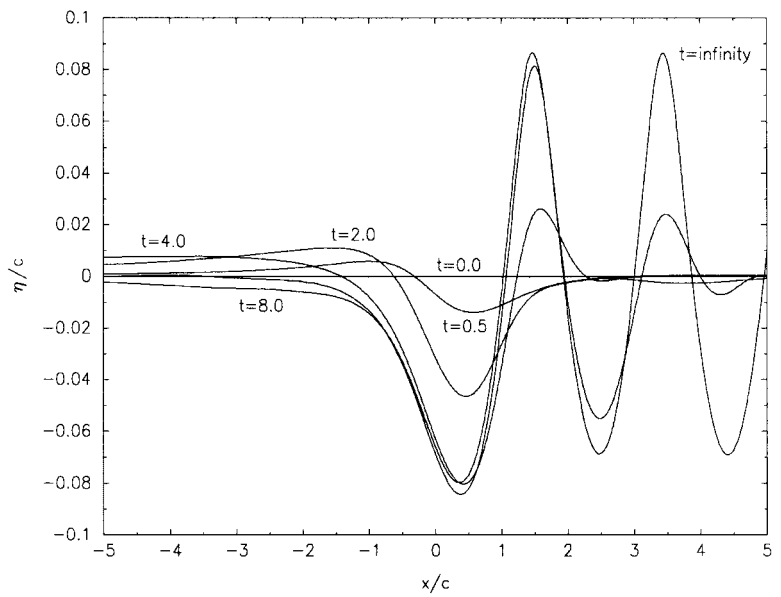

Figure 12. Computed free surface elevation time history for submerged NACA0012 hydrofoil: $Fr = 0.567$, $Re = 10^6$ and $s = 1.03$.

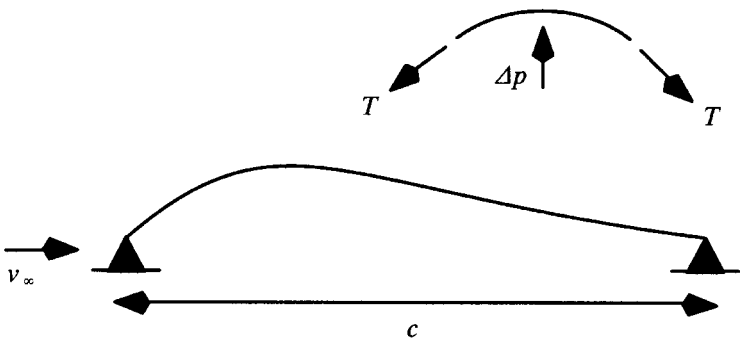

Figure 13. Membrane wing section and free body.

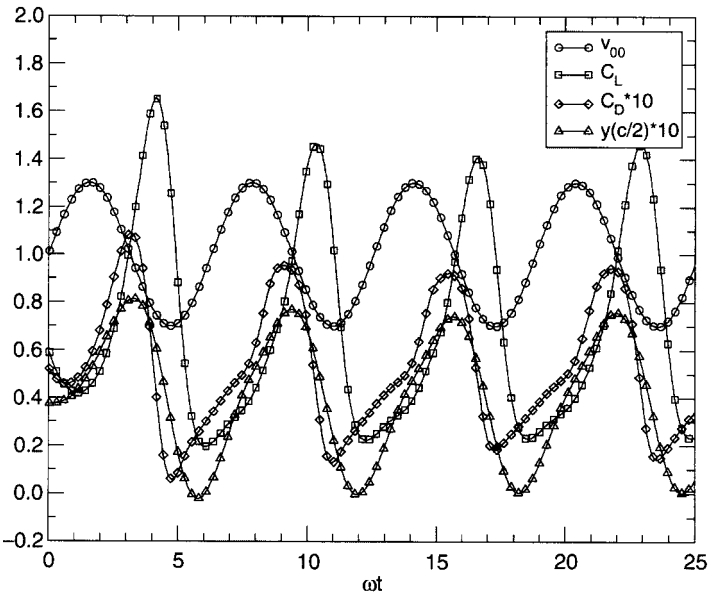

Figure 14. Time series of free stream velocity, lift, drag, and mid-chord co-ordinate for a constant tension membrane airfoil at $Re = 4000$, $St = 1.8$, $\beta = 0.3$ and $\Pi_2 = 2.0$.

4.6. Periodically forced elastic membrane airfoil

Another example problem which uses a body-fitted time-dependent mesh is the aeroelastic problem of a membrane airfoil forced by a harmonically varying free stream velocity. This problem has been referred to as the 'sail in a gust problem' in previous work [20, 21]. The membrane is assumed to be massless and the equilibrium conditions are stated in terms of spatial Cartesian co-ordinates. Figure 13 illustrates an extensible flexible membrane restrained at the leading and trailing edges subjected to fluid dynamic pressure. A discussion of the physical parameters governing the viscous aeroelastic membrane wing problem can be found in Reference [21].

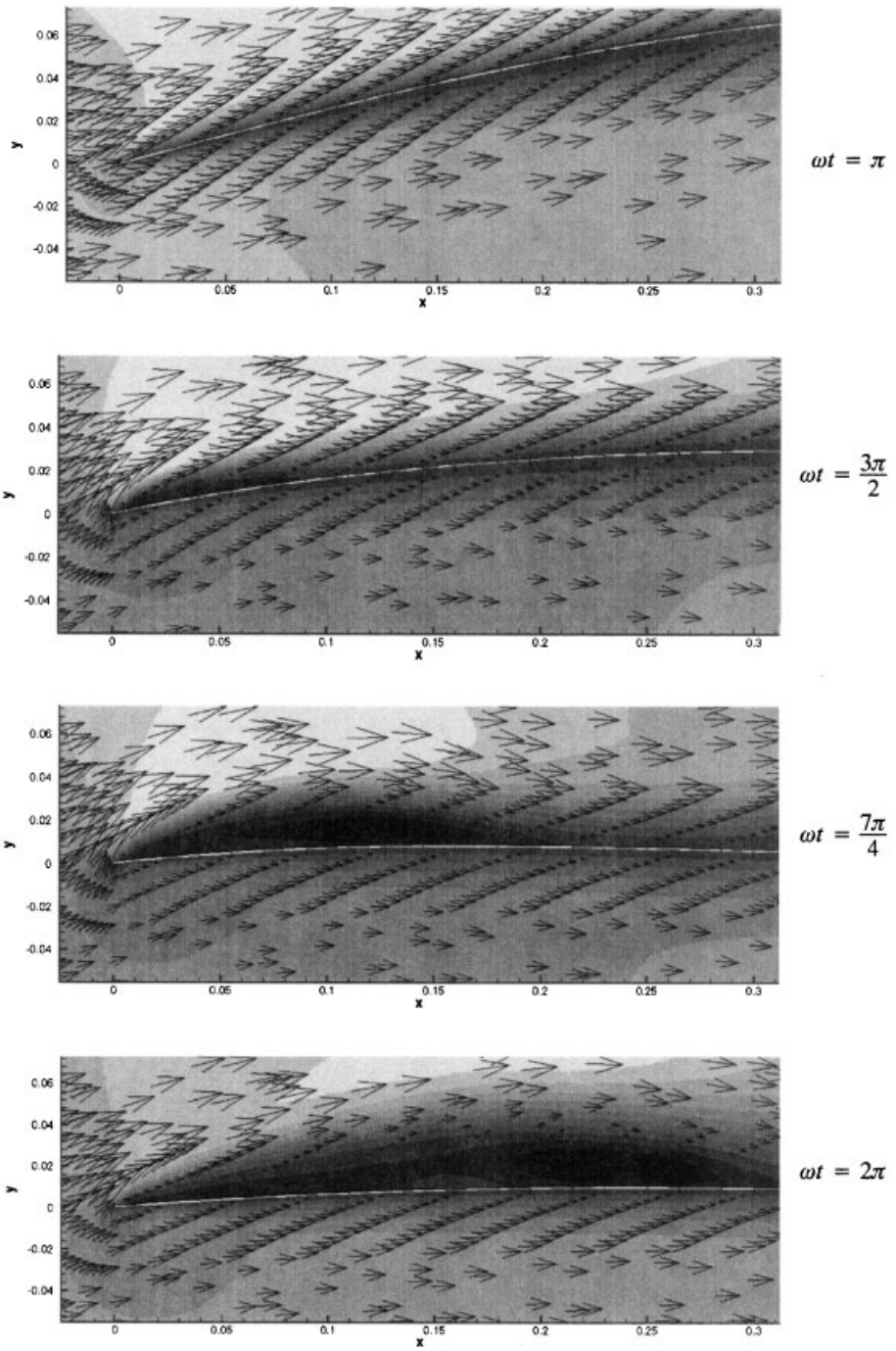

Figure 15. Velocity field and membrane configuration near the leading edge of a membrane airfoil at four instants during the harmonic free stream forcing cycle at $Re = 4000$, $St = 1.8$, $\beta = 0.3$ and $\Pi_2 = 2.0$.

Imposing equilibrium in the normal direction gives

$$\frac{d^2y}{dx^2}\left(1+\left(\frac{dy}{dx}\right)^2\right)^{-3/2} = -\left(\frac{\Delta p}{T}\right) \tag{53}$$

where T is the membrane tension, assumed constant here, and Δp is the net flow pressure difference acting on the membrane. A discrete form of Equation (53) can easily be derived and the equilibrium equation can be appended to the basic system of flow equations as was done for the free surface elevation equation.

Figure 14 shows the time series of free stream velocity, aerodynamic lift and drag, and the y co-ordinate at the midchord point for the end-constrained membrane airfoil at an angle of attack of 4.0°. Again, a simple algebraic grid mover is adopted to co-ordinate the movement of interior nodes [20]. Four complete cycles of free stream forcing are shown in the figure and after an initial transient period the system response can be seen to be nearly periodic at the free stream forcing frequency. The figure shows that the peak in the membrane deflection lags the peak in the freestream velocity by approximately 90°. This phase lag, as well as the large amplitude of the motion of the membrane is characteristic of a system that is being driven near the system natural frequency.

The variation in the membrane profile may also be seen in Figure 15 where the fluid velocity near the membrane leading edge is shown at several instants during one free stream forcing cycle. In the figure it can be seen that as the free stream velocity decelerates, the flow separates along the upper surface of the membrane and a region of recirculating flow develops near the leading edge and is convected downstream. The periodic appearance and collapse of these flow features, along with an attendant adjustment in the membrane configuration, results in an aeroelastic response which may not be characterized as simple harmonic response at the free stream forcing frequency. Details of the calculation and a mesh refinement study can be found in Reference [20] and will not be repeated here.

5. CONCLUSION

A new method for solving the ALE form of the incompressible Navier–Stokes equations has been presented and was shown to be second-order-accurate in both space and time on a general time-dependent polygonal mesh. The method uses a staggered storage arrangement for the flow variables and was shown to be geometrically conservative to the precision of the arithmetic used. Edge formulas were developed for the momentum and pressure equations as well as for the non-linear free surface elevation equation. The method was applied to two rather different moving boundary problems using body-fitted mixed-element meshes. Overall, the method was shown to be stable and accurate for the problems investigated and can be readily extended to three dimensions.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support provided by the Independent Laboratory Independent Research (ILIR) Program at the Coastal System Station, Naval Surface Warfare Center, Dahlgren Division which is sponsored by the Office of Naval Research.

REFERENCES

1. Nomura T, Hughes TJR. An arbitrary Lagrangian–Eulerian finite element method for interaction of fluid and a rigid body. *Computer Methods in Applied Mechanics and Engineering* 1992; **95**:115–138.
2. Ramaswamy B, Kawahara M. Arbitrary Lagrangian–Eulerian finite element method for unsteady, convective, incompressible viscous free surface fluid flow. *International Journal for Numerical Methods in Fluids* 1987; **7**:1053–1075.
3. Nkonga B, Guillard H. Godunov-type method on non-structured meshes for three-dimensional moving boundary problems. *Computer Methods in Applied Mechanics and Engineering* 1994; **113**:183–204.
4. Thomas PD, Lombard CK. Geometric conservation law and its application to flow computations on a moving grid. *AIAA Journal* 1979; **17**:1030–1037.
5. Zhang H, Reggio M, Trepanier JY, Camaero R. Discrete form of the GCL for moving meshes and its implementation in CFD schemes. *Computers in Fluids* 1993; **22**:9–23.
6. Lesoinne M, Farhat C. Geometric conservation laws for flow problems with moving boundaries and deformable meshes and their impact on aeroelastic computations. *Computer Methods in Applied Mechanics and Engineering* 1996; **134**:71–90.
7. Wright JA, Smith RW. An edge-based method for the incompressible Navier–Stokes equations on polygonal meshes. *Journal of Computational Physics* 2001; **169**:24–43.
8. Marx YP. Time integration schemes for the unsteady incompressible Navier–Stokes equations. *Journal of Computational Physics* 1994; **112**:182–209.
9. Barth TJ, Jespersen DC. The design and application of upwind schemes on unstructured meshes. *AIAA Paper 89–0366*, 1989.
10. Barth TJ. Recent developments in high order k-exact reconstruction on unstructured meshes. *AIAA Paper 93–0668*, 1993.
11. Taylor GI. On the decay of vortices in a viscous fluid. *Philosophical Magazine* 1923; **46**:671–675.
12. Coutanceau M, Bouard R. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation, Part 2. Unsteady Flow. *Journal of Fluid Mechanics* 1977; **79**: 257–272.
13. Duncan JH. The breaking and non-breaking wave resistance of a two-dimensional hydrofoil. *Journal of Fluid Mechanics* 1983; **126**:507–520.
14. Hino T. An unstructured grid method for incompressible viscous flows with a free surface. *AIAA Paper 97–0862*, 1997.
15. Lohner R, Yang C, Onate E. Viscous free surface hydrodynamics using unstructured grids. *Twenty-Second Symposium on Naval Hydrodynamics*, Washington, D.C., 1998; pp. 476–490.
16. Smith RW, Wright JA. A new edge-based finite volume algorithm for viscous incompressible free surface flows. *Fifth World Congress on Computational Mechanics*, Vienna, Austria, 2002.
17. Yih CS. *Fluid Mechanics*. West River Press, Ann Arbor, MI, 1977; pp. 166–167.
18. Menter FR. A comparison of some recent eddy-viscosity turbulence models. *ASME Transactions* 1996; **118**: 514–519.
19. Menter FR. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal* 1994; **32**:1598–1605.
20. Smith RW, Wright JA. Simulation of the viscous aeroelastic response of a membrane airfoil to a harmonically varying free stream. *AIAA Paper 2001-0857*, 2001.
21. Smith RW, Shyy W. Computation of unsteady laminar flow over a flexible two-dimensional membrane wing. *Physics of Fluids* 1995; **7**:2175–2184.