AIAA-2003-1119

# Implicit Approaches for Moving Boundaries in a 3-D Cartesian Method

Scott M. Murman*
ELORET
Moffett Field, CA 94035
smurman@nas.nasa.gov

Michael J. Aftosmis†
NASA Ames Research Center
MS T27B Moffett Field, CA 94035

Marsha J. Berger*
Courant Institute
251 Mercer St.
New York, NY 10012

## 1 Introduction

This work considers numerical simulation of three-dimensional flows with time-evolving boundaries. Such problems pose a variety of challenges for numerical schemes, and have received a substantial amount of attention in the recent literature. Since such simulations are unsteady, time-accurate solution of the governing equations is required. In special cases, the body motion can be treated by a uniform rigid motion of the computational domain. For the more general situation of relative-body motion, however, this simplification is unavailable and the simulations require a mechanism for ensuring that the mesh evolves with the moving boundaries. This involves a "remeshing" of the computational domain (either localized or global) at each physical timestep, and places a premium on both the speed and robustness of the remeshing algorithms. This work presents a method which includes unsteady flow simulation, rigid domain motion, and relative body motion using a time-evolving Cartesian grid system in three dimensions.

*Member AIAA
†Senior Member AIAA

While 3-D moving-boundary simulations have been performed on body-fitted structured and unstructured grid systems for some time[1–4], the literature on Cartesian methods has been largely restricted to two dimensions and/or simplified configurations. Early approaches to the 3-D Cartesian moving-body problem included both the Volume of Fluid (VOF)[5] and level-set approaches[6]. More recently, there has been interest in the immersed-boundary[7–9] and cut-cell Cartesian approaches[10–13]. Non-body-fitted, Cartesian approaches like these are particularly interesting since they can be made both extremely fast and robust[14]. Moreover, they are comparatively insensitive to the complexity of the input geometry since the surface description is decoupled from the volume mesh, and can therefore handle complex geometries with relative ease.

This work builds upon the inviscid, Cartesian cut-cell solver in Ref. [15]. In this method, the cells which are cut by the boundary geometry can be arbitrarily small, making explicit update schemes overly restrictive for time-dependent problems. Approaches to overcome this restriction usually either extend the difference stencil of the spatial terms[16], or use a cell-merging approach [10], so that cut-cells can be advanced with the explicit timestep of a full *uncut* cell. Both approaches have been successful for two-dimensional simulations with moving boundaries[10, 13, 16], however, the coupling of cell size and allowable timestep with explicit methods implies that the boundary motion will be restricted by the size of the finest boundary intersecting Cartesian cell during a single timestep.

In a Cartesian moving-boundary scheme, the most delicate operation is the re-intersection of the body with Cartesian grid at each time step. Not only is this the most computationally expensive part of Cartesian mesh generation, but it requires special procedures to ensure that the floating-point intersection calculations are always robust[14]. In a two dimensional mesh with $O(N^2)$ cells, the boundary geometry only intersects $O(N)$ cells. In three dimensions, however, the boundary is a surface instead of a line and the number of intersection calculations is squared with respect to the 2-D case. Moreover, these intersection calculations are higher-dimensional, and therefore each is more expensive and more difficult to compute robustly. Arguments for both efficiency and robustness in three dimensions weigh heavily in favor of moving the body as rarely as possible to minimize the re-cutting of Cartesian cells. These arguments are amplified in parallel-computing environments, where mesh modifications often imply rebalancing the load distributed to the various processors by exchanging cells across subdomain boundaries.

Following this line of reasoning, the present work adopts a fully-implicit temporal operator to decouple the timestep from the local mesh scale. It leverages the same multigrid smoother used by the steady-state solver[15] by embedding it in a dual-time framework. The implicit approach means that finer mesh simulations do not automatically require more remeshings than coarse simulations. Simulations proceed at a timestep dictated by the appropriate physics rather than stability constraints.

As noted earlier, special subclasses of arbitrary body motion can be treated by rigid motion of the entire domain. The current work implements a rigid domain

motion within the Arbitrary Lagrangian-Eulerian (ALE) formulation of Hirt et al.[17]. In this *Lagrangian* framework, a single transformation matrix can be applied to all the faces in a Cartesian mesh, and since these faces all have the same orientations, the transformation can be inexpensively precomputed, stored and applied.

Relative boundary motion is superimposed on top of the rigid domain motion using an adaptive re-meshing strategy which does not rely on explicit cell merging. Since the geometry moves relative to the volume cells, this aspect of the implementation is *Eulerian*. A space-time analysis is used to ensure conservation, and to develop a hierarchy of approximations to the moving boundary. The basic analysis is presented in detail in 1-D, with the complexity which arises when extending to multiple dimensions also discussed. The final section presents initial numerical results in one, two, and three dimensions using the implicit, relative-motion scheme, including comparison with analytic solutions and experimental data.

# 2 Dual-time formulation

In order to leverage the infrastructure of the steady-state flow solver outlined in Ref. [15], a dual-time formulation (cf. Refs. [18, 19]) was developed for the time-dependent scheme,

$$\begin{aligned} \frac{d\mathbf{Q}}{d\tau} + R^*(\mathbf{Q}) &= 0 \\ R^*(\mathbf{Q}) &= \frac{\partial \mathbf{Q}}{\partial t} + R(\mathbf{Q}) \end{aligned} \tag{1}$$

where $\tau$ is referred to here as "pseudo-time", and is the iterative parameter, and $t$ is the physical time. $\mathbf{Q}$ is the vector of conserved variables, and $R(\mathbf{Q})$ is an appropriate numerical quadrature of the flux divergence, $\frac{1}{V}\oint_S \mathbf{f}\cdot\mathbf{n}dS$. As $\frac{d\mathbf{Q}}{d\tau} \rightarrow 0$ the time-dependent formulation is recovered. The multi-grid smoother described in [15] is used to converge the inner pseudo-time integration. An explicit, multi-stage, pseudo-time-integration scheme is utilized to converge the "inner loop" in Eqn. 1. This is similar to the scheme outlined by Jameson[20], however, the semi-implicit approach of Melson et al.[21] is used here for the physical time-derivative term.

Various time-dependent schemes can be constructed for Eqn. 1 by appropriately discretizing the time derivative. As noted in the Introduction, it's desirable to utilize an unconditionally-stable, implicit scheme to allow a large timestep to be chosen based upon physical considerations rather than a potentially smaller stability-limited timestep. Beam and Warming[22] outline a family of consistent time-dependent schemes that utilize three time levels, $\mathbf{Q}^{n-1}$, $\mathbf{Q}^n$, $\mathbf{Q}^{n+1}$. These are given by

$$\begin{aligned} R^*(\mathbf{Q}) = &\frac{(1+\xi)\mathbf{Q}^{n+1} - (1+2\xi)\mathbf{Q}^n + \xi\mathbf{Q}^{n-1}}{\Delta t} \\ &+ \theta R\left(\mathbf{Q}^{n+1}\right) + (1-\theta+\phi)R\left(\mathbf{Q}^n\right) - \phi R\left(\mathbf{Q}^{n-1}\right) \end{aligned} \tag{2}$$

| $\theta$ | $\xi$ | $\phi$ | Method | Order |
|---|---|---|---|---|
| 1 | 0 | 0 | Backward Euler | 1 |
| 1/2 | 0 | 0 | Trapezoidal | 2 |
| 1 | 1/2 | 0 | 2nd-order backward | 2 |
| 3/4 | 0 | -1/4 | Adam's type | 2 |
| 1/3 | -1/2 | -1/3 | Lee's type | 2 |
| 1/2 | -1/2 | -1/2 | Two-step trapezoidal | 2 |
| 5/9 | -1/6 | -2/9 | A-contractive | 2 |

**Table 1:** Partial list of A-stable, two- and three-level methods (Eqn. 2) from Beam and Warming[22].

Table 1 contains a partial list of the A-stable, three time level methods that can be formulated using Eqn. 2.

# 3 ALE formulation

In many applications with moving geometry, the motion of the geometry can be decomposed into a "uniform" rigid-body motion, with relative motion confined to a subset of the domain. Examples include rotating airframes with dithering canards[23], rotorcraft, or stage separation from space vehicles. It's desirable to simulate such a motion using a rigid-body motion of the entire computational domain, and treat the relative motion within the moving domain separately. This again limits the amount of computational work that is required to process the moving geometry.

An ALE formulation (cf. Hirt et al. [17]) was utilized in order to account for the rigid-body motion of the computational domain. This is accomplished by modifying the flux through a boundary to account for the motion of the boundary. For the inviscid flux vector used here, this becomes

$$\mathbf{f} \cdot \mathbf{n} = \left\{ \begin{array}{c} \rho u_n \\ \rho u_n \mathbf{u} + p\mathbf{n} \\ \rho u_n e + p\mathbf{u} \cdot \mathbf{n} \end{array} \right\} \tag{3}$$

where

$$u_n = (\mathbf{u} - \mathbf{u}_\Omega) \cdot \mathbf{n}$$

is the velocity relative to the moving boundary, and $\mathbf{u}_\Omega$ is the velocity of the moving domain. Hence the convective part of the flux is modified to account for the motion of the boundary, compared to the treatment for a fixed domain.

A modified form of van Leer's flux-vector splitting (FVS) [24] is used with the ALE formulation. This modification generalizes the scheme by using the Mach number relative to the moving boundary when determining the characteristic speeds of the system. If the relative Mach number is not used the scheme will not be Galilean invariant; simulations with a static domain will not provide the same numerical results as those which use a moving domain to compute the same physical problem. The

subsonic flux for van Leer's FVS can be written as a combination of convective and acoustic terms (for a static domain) as

$$\hat{\mathbf{f}}^{\pm} \cdot \mathbf{n} = M_n^{\pm} \begin{Bmatrix} \rho c \\ \rho c \mathbf{u} \\ \rho c H \end{Bmatrix} \pm M_n^{\pm} (2 \mp M_n) \begin{Bmatrix} 0 \\ p\mathbf{n} \\ 0 \end{Bmatrix} \mp \frac{\gamma}{\gamma - 1} M_n^{\pm} (2 \mp M_n)^2 \begin{Bmatrix} 0 \\ 0 \\ pc \end{Bmatrix} \quad (4)$$

where $M_n c = \mathbf{u} \cdot \mathbf{n}$ and $M_n^{\pm} = \pm \frac{1}{4} (1 \pm M_n)^2$. The numerical flux across a boundary which separates two domains (left and right) then becomes $\hat{\mathbf{f}}(\mathbf{Q}_L, \mathbf{Q}_R) = \hat{\mathbf{f}}^{+}(\mathbf{Q}_L) + \hat{\mathbf{f}}^{-}(\mathbf{Q}_R)$.

With an ALE formulation, the Mach number relative to the moving boundary is $M_n c = (\mathbf{u} - \mathbf{u}_\Omega) \cdot \mathbf{n}$. If this relative Mach number is simply substituted into van Leer's FVS when computing on a moving domain, the energy equation will not be consistent, in the sense of $\hat{\mathbf{f}}(\mathbf{Q}, \mathbf{Q}) = \mathbf{f}(\mathbf{Q})$. This is due to combining the total energy and pressure work terms into the total enthalpy $H$ in the numerical energy flux in Eqn. 4. Physically, the pressure work due to the domain motion ($p\mathbf{u}_\Omega$) is identically zero, and hence this term does not appear in the energy flux in Eqn. 3.

A modified form of van Leer's FVS was developed for the ALE scheme. The mass and momentum flux are unmodified from the original form, only the energy flux is changed. Examining the energy flux in Eqn. 4, it's seen that the last term disappears when $\mathbf{Q}_L = \mathbf{Q}_R$. Neglecting this last term in the energy flux, the total enthalpy is split into the total energy and pressure work, and these terms are treated separately in the same manner as the momentum equations. The modified scheme then becomes

$$\hat{\mathbf{f}}^{\pm} \cdot \mathbf{n} = M_n^{\pm} \begin{Bmatrix} \rho c \\ \rho c \mathbf{u} \\ \rho c e \end{Bmatrix} \pm M_n^{\pm} (2 \mp M_n) \begin{Bmatrix} 0 \\ p\mathbf{n} \\ p\mathbf{u} \cdot \mathbf{n} \end{Bmatrix} \quad (5)$$

where $M_n^{\pm}$ is now based upon the relative Mach number. The convective terms and acoustic terms are treated consistently in all 3 equations of the system. This numerical flux is similar to the WPS scheme developed by Agarwal and Halt[25]

The modified scheme for the energy flux maintains the smoothness property of van Leer's original FVS. It is continuously-differentiable through $M_n = \pm 1$, as can be seen in Fig. 1, where the normalized energy flux ($\frac{f}{\rho c^3}$) is plotted against the Mach number. Both FVS schemes smoothly approach the asymptotic limit of pure upwinding.

The geometry of the domain is expressed in the moving coordinate system, and must be transformed to the inertial system where the equations of motion are specified. With a Cartesian scheme, the faces of each computational cell have normals pointing in one of the Cartesian directions. These normals all transform to the inertial system similarly, and are simply pre-computed and stored prior to each timestep.

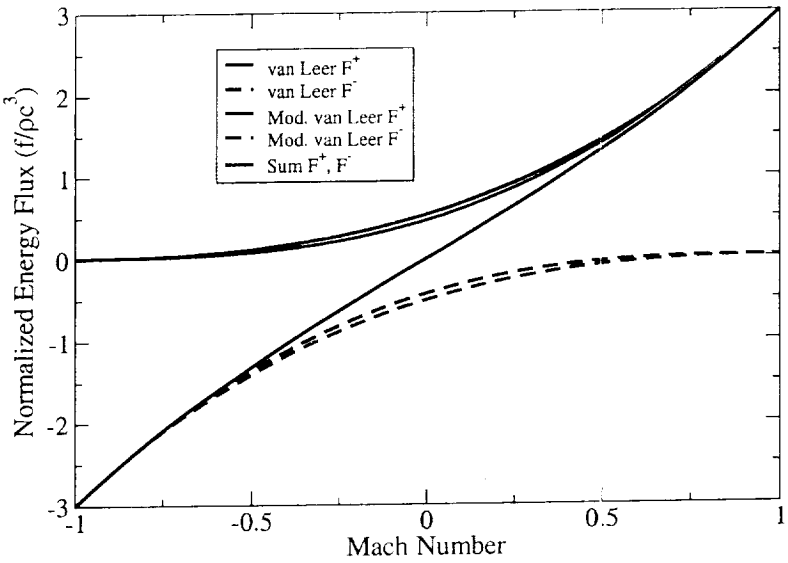



**Figure 1:** Normalized energy flux for the FVS schemes, Eqns. 4 and 5.

The ALE formulation was utilized to simulate a transonic NACA 0012 pitching airfoil (cf. AGARD Report 702 [26]). The 2nd-order backward time-integration scheme was used to compute the unsteady motion starting from a converged steady-state solution. The simulation uses 100 timesteps per complete cycle of the airfoil, and an isotropic mesh with a finest resolution of 0.001 chord. As the airfoil oscillates, the shock transitions between the upper and lower surfaces of the airfoil, and the fluid state is path-dependent. This hysteresis is evident in the normal force history plotted in Fig. 2. After an initial transient of approximately 1/2 cycle, the solution is periodic. At a given angle of attack, the normal force is multi-valued depending upon whether the airfoil is pitching up or down. The computed variation of normal force is in good agreement with the experimental data.

# 4 Relative motion

As an introduction to computing moving boundaries with a non-body-fitted Cartesian scheme, Fig. ?? contrasts three methods of moving geometry during a timestep: overset meshes, deforming meshes, and the current Cartesian approach. Both the overset and deforming mesh approaches use body-fitted volume meshes. Overset approaches (cf. [1, 27, 28]) are relatively easy to implement and efficient, as the volume mesh processing is done a priori; only (lower-dimensional) boundary interpolations are updated at each step. Overset methods cannot maintain conservation across composite grid boundaries without complex re-meshing of the overlap region. It also can be difficult and labor-intensive to maintain a compatible spatial resolution across composite grids, and to process the boundaries when components are close to-

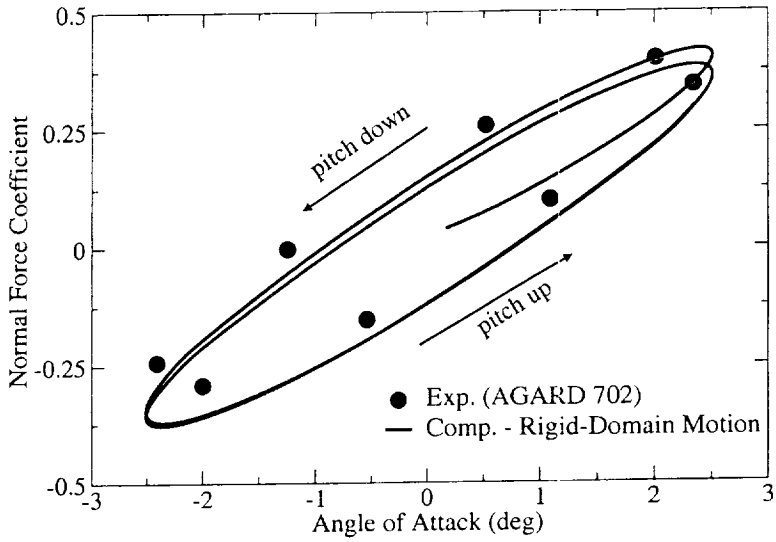



**Figure 2:** Variation of normal force coefficient with angle of attack for oscillating NACA 0012. ($M_\infty = 0.755$, $\alpha(t) = 0.016 + 2.51 \sin(2\pi 62.5t)$). The simulation uses 100 timesteps per complete cycle of the airfoil, and an isotropic mesh with a finest resolution of 0.001 chord. Experimental data from [26].

gether. With deforming meshes (cf. [3, 29, 30]), the volume mesh deforms in response to the surface motion, and after large deformations a volume re-meshing and (often non-conservative) interpolation to the new mesh is performed. Deforming-mesh approaches can be conservative over a timestep, making them attractive for small deformations, however for large timesteps (which lead to large boundary movements), the quality of the cell stencil can severely degrade due to the deformations. In the current Cartesian approach, the moving boundary "sweeps" through a fixed Eulerian mesh over a timestep. The cells within the swept region of the domain change volume and shape over the timestep, and cells can appear or disappear (or both) as well. The space-time geometry associated with these swept cells is complicated, however it is possible to formulate conservative schemes as the boundary is impermeable, hence the flux through the moving boundary is always known to some order of accuracy. Formulating a non-body-fitted Cartesian scheme which maintains conservation for large timesteps (large motions) of a moving boundary, while still retaining efficiency and robustness is the challenge for the current (and future) research.

## 4.1 Governing Equations

The motion of a solid body through an inviscid fluid discretized by a fixed Cartesian mesh is governed by the same ALE set of conservation equations (a Lagrangian body moves through an Eulerian mesh) as the rigid-domain motion of the previous

section,

$$\frac{d}{dt}\int_{V(t)} \mathbf{Q}dV = -\oint_{S(t)} \mathbf{f}\cdot\mathbf{n}dS \qquad S = \partial V$$
$$\mathbf{f}\cdot\mathbf{n} = \left\{ \begin{array}{c} \rho u_n \\ \rho u_n \mathbf{u} + p\mathbf{n} \\ \rho u_n e + p\mathbf{u}\cdot\mathbf{n} \end{array} \right\} \tag{6}$$
$$u_n = (\mathbf{u}-\mathbf{w})\cdot\mathbf{n}$$

Here $\mathbf{w}$ is the velocity of the moving boundary with respect to the Eulerian frame, and is used to differentiate from the rigid domain velocity $\mathbf{u}_\Omega$. For the current discussion the rigid-body motion of the domain will be ignored, and only the relative motion will be formulated. No changes to the current scheme are required when the rigid-domain motion is superposed.

For a cell in a Cartesian mesh swept by a moving boundary, Eqn. 6 can be simplified as the boundaries of the cell are fixed, hence $\mathbf{w}\cdot\mathbf{n} = 0$, except for the motion of the solid surface through the volume, for which $\mathbf{u}\cdot\mathbf{n} = \mathbf{w}\cdot\mathbf{n}$ holds (i.e. the convective portion of the flux is zero through an impermeable surface). Eqn. 6 is preferred over a simplified form however, as it emphasizes the deforming-cell nature of the problem.

Applying Liebniz' rule to the left side of Eqn. 6 gives

$$\frac{d}{dt}\int_{V(t)} \mathbf{Q}dV = \int_{V(t)} \frac{\partial \mathbf{Q}}{\partial t}dV + \oint_{S(t)} \mathbf{Q}\mathbf{w}\cdot\mathbf{n}dS \tag{7}$$

and the right-hand-side flux through the boundary surface can be decomposed as

$$-\oint_{S(t)} \mathbf{f}\cdot\mathbf{n}dS = -\oint_{S(t)} \left[\mathbf{Q}\left(\mathbf{u}-\mathbf{w}\right)+\mathbf{P}\right]\cdot\mathbf{n}dS \tag{8}$$

where $\mathbf{P} = \begin{pmatrix} 0 \\ p\boldsymbol{\delta} \\ p\mathbf{u} \end{pmatrix}$ is the acoustic portion of the flux. Balancing terms gives

$$\int_{V(t)} \frac{\partial \mathbf{Q}}{\partial t}dV + \oint_{S(t)} \left[\mathbf{Q}\mathbf{u}+\mathbf{P}\right]\cdot\mathbf{n}dS = 0 \tag{9}$$

Equation 9 can be recast into a space-time divergence form by applying Gauss's thereom to the spatial volume integral. Defining the 4-D space-time normal as $\widetilde{\mathbf{n}} = \{\hat{t}, \mathbf{n}\}$, where $\hat{t}$ is the normal in the time direction, $\mathbf{n}$ is the unit spatial normal, and $\widetilde{\mathbf{Q}} = \{\mathbf{Q}, \mathbf{Q}\mathbf{u}+\mathbf{P}\}$, the space-time conservation equation is

$$\oint \widetilde{\mathbf{Q}}\cdot\widetilde{\mathbf{n}}d\Omega = 0 \tag{10}$$

where $\Omega$ is the boundary of the space-time volume. Expanding $\widetilde{\mathbf{Q}}$ for clarity gives

$$\widetilde{\mathbf{Q}} = \left\{ \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ \rho e \end{pmatrix} \hat{t}, \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\mathbf{u}+p\boldsymbol{\delta} \\ \rho e\mathbf{u}+p\mathbf{u} \end{pmatrix} \mathbf{n} \right\} \tag{11}$$

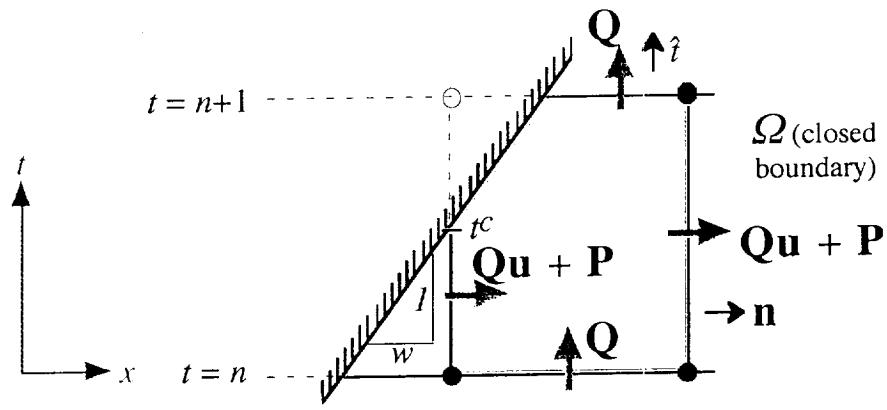



**Figure 3:** One-dimensional space-time cell. The impermeable moving boundary (shown in red) crosses the fixed left face of the cell at time $t^c$. The impermeable boundary has a normal with elements in both space and time.

Note that the velocity of the moving solid surface no longer explicitly appears in Eqn. 10 as the motion of the boundaries is accounted for in the direction and "area" of the space-time boundary. Fig. 3 shows a 1-D space-time cell volume for a Cartesian cell as a moving-boundary crosses the left cell face. The impermeable portion of the space-time cell boundary (red boundary in Fig. 3 with slope $1/w$) has a normal with elements in both space and time, and is analogous to the role of the boundary velocity in Eqn. 6. In essence, the mixed Lagrangian/Eulerian formulation of Eqn. 6 has been converted to an Eulerian formulation in space-time.

Equations 6 and 10 explicitly satisfy the so-called Geometric Conservation Law (GCL) (cf. Thomas and Lombard[31]), and no supplementary information is required. The GCL is usually presented as

$$\frac{d}{dt}\int_{V(t)} dV = \oint_{S(t)} \mathbf{w}\cdot\mathbf{n}dS \tag{12}$$

which states that the change in cell volume is equivalent to the area swept by the moving boundary. The supplementary information that the cell must close, $\oint \mathbf{n}dS = 0$, is also required. The space-time analog to the GCL is

$$\oint \tilde{\mathbf{n}}d\Omega = 0 \tag{13}$$

which states that the space-time cell must close.

Whether numerical schemes are built for Eqn. 6 or Eqn. 10 is largely a matter of convenience, as both are mathematically equivalent. In the current work, Eqn. 6 is discretized directly, since it maintains a similar formulation as the governing equations for both static and rigid-domain motions. This leverages the existing flow solver infrastructure for multigrid, dual-time, etc., for solving the relative motion equations.

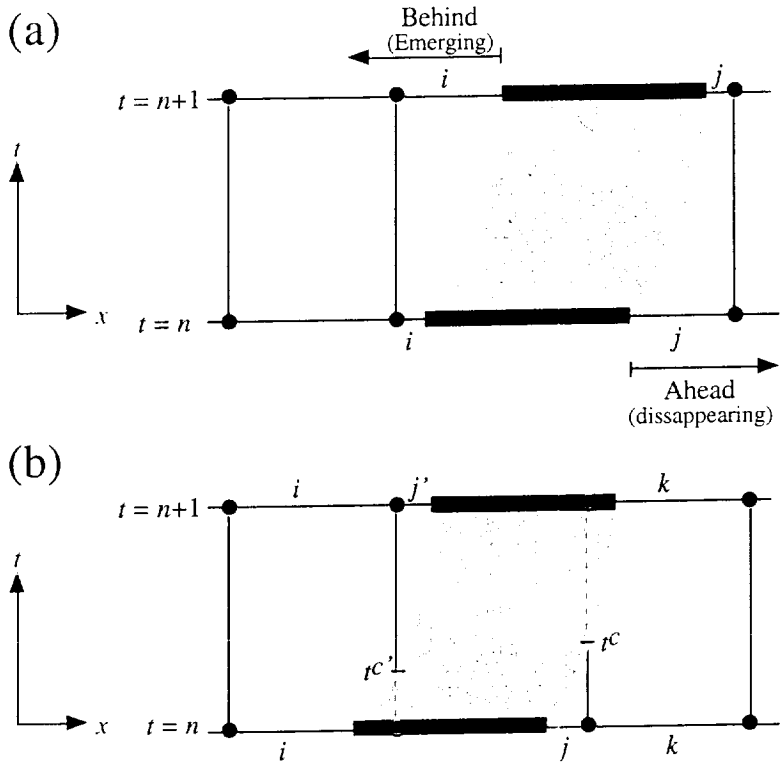



**Figure 4:** One-dimensional motions of an impermeable boundary for motion less than the uncut hexahedron spacing ($\mathrm{CFL}_w < 1$).

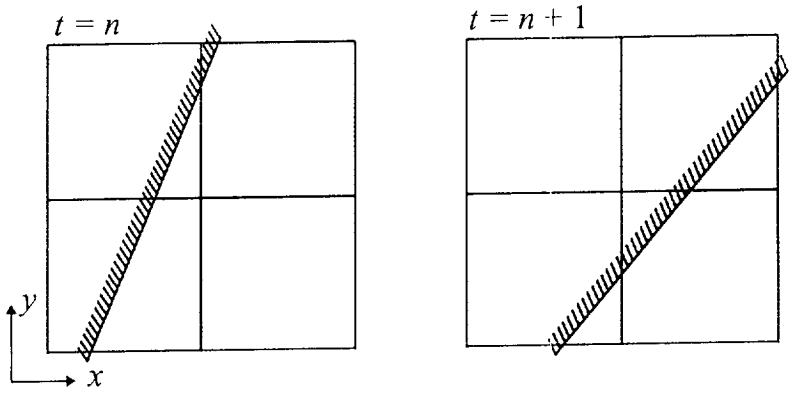
and spatial directions. The temporal (convective) contribution is identically zero, as $\mathbf{u}\cdot\mathbf{n} = \mathbf{w}\cdot\mathbf{n}$ for an impermeable boundary. This leaves only the pressure contribution to the moving-boundary flux, which in the current 1-D example becomes

$$\int_{t^n}^{t^{n+1}} \tilde{f}_j^L d\Omega = \int_{t^n}^{t^{n+1}} p_w dt \tag{16}$$

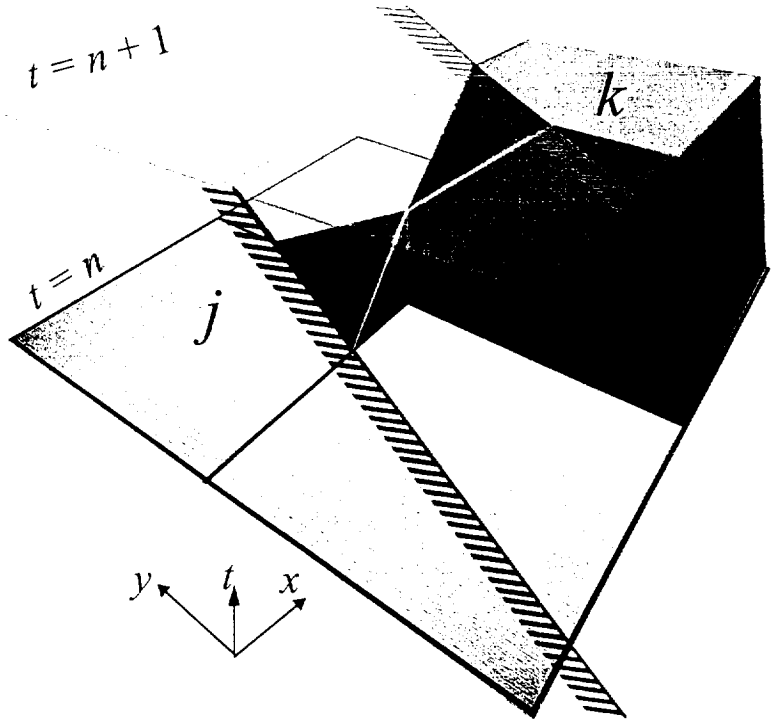
where $p_w$ represents an approximation for the wall pressure to some order of accuracy. In other words, pressure acts only in the spatial directions, and hence the space-time area over which the surface pressure acts is the projection along the spatial axes of the moving wall area. In the vector notation of the previous section

$$\left(\widetilde{\mathbf{Q}}\cdot\widetilde{\mathbf{n}}\right)_w = (\mathbf{P}\cdot\mathbf{n})_w \tag{17}$$

An example in multiple dimensions is shown in Fig. 5 where a boundary moves through a group of hexahedra in 2-D. Examining the space-time cell $k$, the pressure acts on the projection of the moving front along the $-x$ and $+y$ axes. The space-time areas over which the pressure acts are then the triangular projections in Fig. 5b.




(a)




(b)

**Figure 5:** 2-D cells swept by a moving boundary. The moving boundary is shaded in red, and begins at time level $n$ in hex $j$. The boundary both rotates and translates, moving in the $+x$ and $-y$ directions over the timestep. At time level $n+1$ hex $j$ is completely interior to the boundary, and cell $k$ is now cut.

For cells such as sketched in Fig. 4a, where the solid boundary does not cross a cell vertex over the timestep, it is unnecessary to calculate any space-time geometry or modify the numerical scheme in any way. Conservation is satisfied if the change in cell volume is known, and the accuracy of the scheme for Eqn. 14 is determined by the form of the flux quadrature through the space-time areas.

Fig. 4b again shows a boundary motion which is less than the uncut hexahedron spacing, however in this example the boundary crosses a cell vertex during the timestep. The cell labeled $j$ disappears ahead of the moving front, while the new cell $j'$ emerges behind the front. Examining the situation ahead of the moving front first, the cell $j$ does not exist on the mesh at time level $n+1$, hence nothing needs to be computed in this cell. The cell $k$ is the first cell at time level $n+1$ ahead of the front. This cell receives a contribution from the uncut face to the right, and two contributions on the left: a flow contribution from $t^n$ to $t^c$, the time of the vertex crossing, and a solid wall contribution from $t^c$ to $t^{n+1}$. In general, the flux through a space-time face is thus composed of a flow portion, $\Omega_f$, and a boundary contribution $\Omega_w$. The semi-discrete equations for cell $k$ are written as

$$(\rho uV)_k^{n+1} - (\rho uV)_k^n + \int_{t^n}^{t^{n+1}} \tilde{f}_k^R d\Omega - \int_{t^n}^{t^c} \tilde{f}_k^L d\Omega_f - \int_{t^c}^{t^{n+1}} p_w dt = 0 \qquad (18)$$

where the simplification for the solid wall contribution $\Omega_w$ from $t^c$ to $t^{n+1}$, Eqn. 16, has been made. The flux entering cell $k$ from $t^n$ to $t^c$ is identical to the flux out of cell $j$. The semi-discrete conservation equations for cell $j$ are

$$-(\rho uV)_j^n + \int_{t^n}^{t^c} \tilde{f}_j^R d\Omega_f - \int_{t^n}^{t^c} p_w dt = 0 \qquad (19)$$

solving for $\int \tilde{f}_j^R d\Omega_f$ and substituting for $-\int \tilde{f}_k^L d\Omega_f$ in Eqn. 18 gives

$$(\rho uV)_k^{n+1} - (\rho uV)_k^n - (\rho uV)_j^n + \int_{t^n}^{t^{n+1}} f^R d\Omega - \int_{t^n}^{t^{n+1}} p_w dt = 0 \qquad (20)$$

as the semi-discrete equations for cell $k$ in Fig. 4b. Comparing with Eqn. 14, the only required modification is a conservation correction from cell $j$. This "cell-merging" correction will be discussed further in the next section.

The conversion of the flux in cell $k$ from $t^n$ to $t^c$ to a pressure contribution in Eqn. 20 is not general, and in multiple-dimensions the spatial flux for a cell ahead of the front will contain both wall and flow contributions. This is seen in Fig. 5b, where the cell $j$ closes as the boundary moves, however cell $k$ ahead of the moving boundary has both wall and flow components of the space-time flux through its $-x$ and $+y$ space-time faces. In order for the flow component of the flux to telescope completely to a pressure component all of the spatial faces of a cell must become closed.

Cell $j'$ in Fig. 4b emerges behind the body at $t^{c'}$. Again, the temporal (convective) contribution due to the moving boundary is zero for an impermeable boundary, and

the only contribution due to pressure is a projection of the wall along the spatial axes. The left face of cell $j'$ receives a flux contribution beginning at $t^{c'}$ so that the semi-discrete governing equations for cell $j'$ are

$$(\rho uV)_{j'}^{n+1} + \int_{t^{c'}}^{t^{n+1}} p_w dt - \int_{t^{c'}}^{t^{n+1}} \tilde{f}_j^l d\Omega_f = 0 \tag{21}$$

Unlike the cell $k$ ahead of the front, for cells behind the moving boundary in 1-D it is necessary to determine the time $t^{c'}$ in some manner. Various levels of approximation for determining this vertex crossing time will be outlined in Sec. 4.2.3, after the current discussion is extended to arbitrarily-large boundary motions in the next section.

### 4.2.2 Arbitrary CFL

The implicit framework of the current method requires that the numerical scheme allow large geometry motions during a timestep. Figure 6 shows an impermeable boundary moving through several cells of a Cartesian mesh over a single timestep, again in 1-D. Hexahedra such as $j$, $k$, and $l$ in Fig. 6 are referred to as "time-split" hexahedra, in a similar manner as spatially-split hexahedra (cf. Aftosmis et al.[14]). The cells behind the moving front are processed as described in Eqn. 21 without modification. Examining the cells ahead of the moving front which no longer appear at time level $n+1$, the semi-discrete equation for cell $m$ can be written as

$$(\rho uV)_m^{n+1} - (\rho uV)_m^n - \sum_{ahead} (\rho uV)^n + \int_{t^n}^{t^{n+1}} \tilde{f}_m^R d\Omega - \int_{t^n}^{t^{n+1}} p_w dt = 0 \tag{22}$$

The term $\sum_{ahead}(\rho uV)^n$ represents a conservation correction for cell $m$ which is an agglomeration of the conserved quantity in all the cells ahead of the moving front. In 1-D, determining this conservation correction term is unambiguous. An example of the complexity in multi-dimensions is shown in Fig. 5b, where the correction from cell $j$ must be apportioned among its three neighbors (those sharing the $-y$ face, $+x$ face, and the diagonal neighbor $k$). This apportionment is similar to the flux redistribution used by Pember et al.[32].

The "flux telescoping" used here is analogous to the cell-merging technique used in explicit, 2-D simulations by Bayyuk et al.[10, 11]. In cell merging the cells surrounding the moving body at both time levels are physically merged into a single cell, and then integrated forward in time. After the timestep a reconstruction is performed back to the unmerged mesh. The complex implementation to physically merge the cells is avoided in the current method. Ahead of the front, the cell merging and flux telescoping techniques are mathematically equivalent, however behind the front they are not necessarily the same. For example, since the current method maintains the discretization through the timestep, a reconstruction step is not required behind the front. This reconstruction can be problematic for large boundary motions and in multi-dimensional simulations.

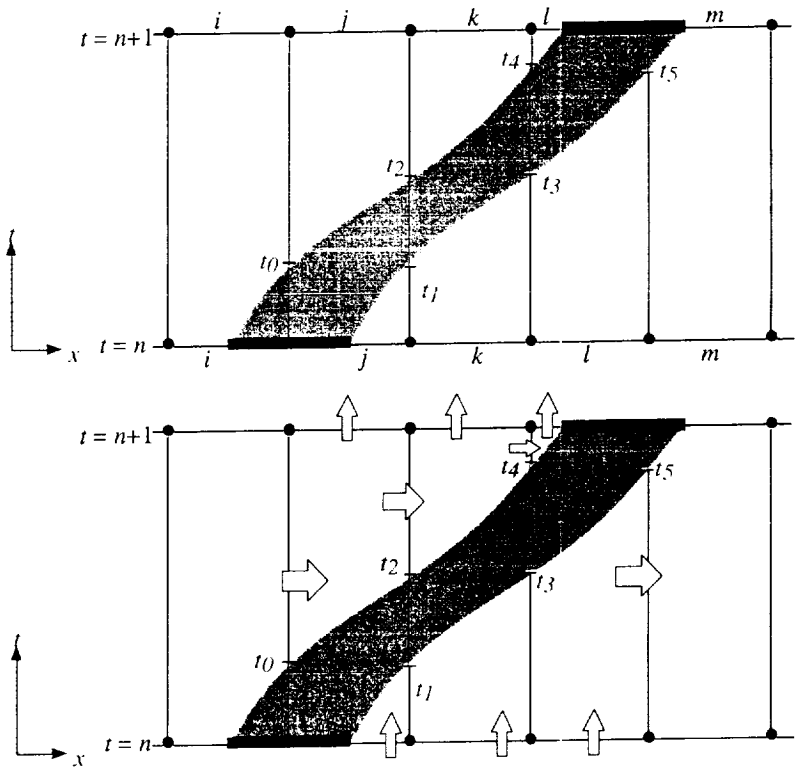



**Figure 6:** One-dimensional motions of an impermeable boundary for motion greater than the uncut hexahedron spacing ($\mathrm{CFL}_w > 1$).

### 4.2.3 Levels of Space-time Geometry Approximation

As described in the previous section, in general, some details of the space-time geometry must be determined in order to evaluate the spatial flux terms. As a first step, the current article focuses on two-time-level schemes with the flow and geometry states synchronous. These require only a single computation of the cut-cell intersection per timestep (time level $n$ simply being saved from $n+1$ of the previous step). A similar approach staggers the geometry and fluid states in time[8], and also only requires a single geometry intersection per step. As described previously, the change in the conserved quantity can be discretized directly since the cells volumes are known at time levels $n$ and $n+1$ as a result of the volume mesh generation. This section describes levels of approximation in determining the required space-time geometry, beginning with the simple and becoming more complex. Determining the space-time geometry affects the accuracy of the numerical scheme, however conservation is maintained discretely regardless of how the space-time geometry is approximated.

With the Cartesian approach, it is not explicitly required to determine the details of the moving boundary motion, and non-planar wall motions can potentially be evaluated. Fig. 7 shows an isolated view of the 2-D cell $k$ from Fig. 5. If the flow contributions to the space-time area ($\Omega_f$) are known, then the wall contribution can be determined by using the space-time GCL, Eqn. 13. Since Eqn. 13 is a vector equation it can be applied to each Cartesian direction independently. Thus in order to determine the wall contribution it is simply required to enforce that the space-time cell volume closes. In this manner it is not necessary to explicitly linearize the wall motion in order to determine the wall flux contribution, though with most methods an implicit linearization does occur as will be described.

The space-time GCL can also be used to simplify the pressure work due to the moving boundary, so that it is unnecessary to evaluate any space-time geometry. The pressure work term is given by

$$\int_w p\mathbf{u} \cdot \mathbf{n} d\Omega_w \tag{23}$$

where here the integral is taken over the moving boundary within the space-time cell. If the wall pressure is approximated with a constant value $p_w$, and the substitution $\mathbf{u} \cdot \mathbf{n} = \mathbf{w} \cdot \mathbf{n}$ for an impermeable boundary is made, then Eqn. 23 becomes

$$p_w \int_w \mathbf{w} \cdot \mathbf{n} d\Omega_w \tag{24}$$

The integral $\int_w \mathbf{w} \cdot \mathbf{n} d\Omega$ is the area swept by the moving boundary over the timestep, which is equivalent to the change in volume within the space-time cell (cf. Eqn. 7). Hence the pressure work term can be evaluated using the known cell volumes at $n$ and $n+1$ as

$$\int_w p\mathbf{u} \cdot \mathbf{n} d\Omega_w \approx p_w \Delta V \tag{25}$$

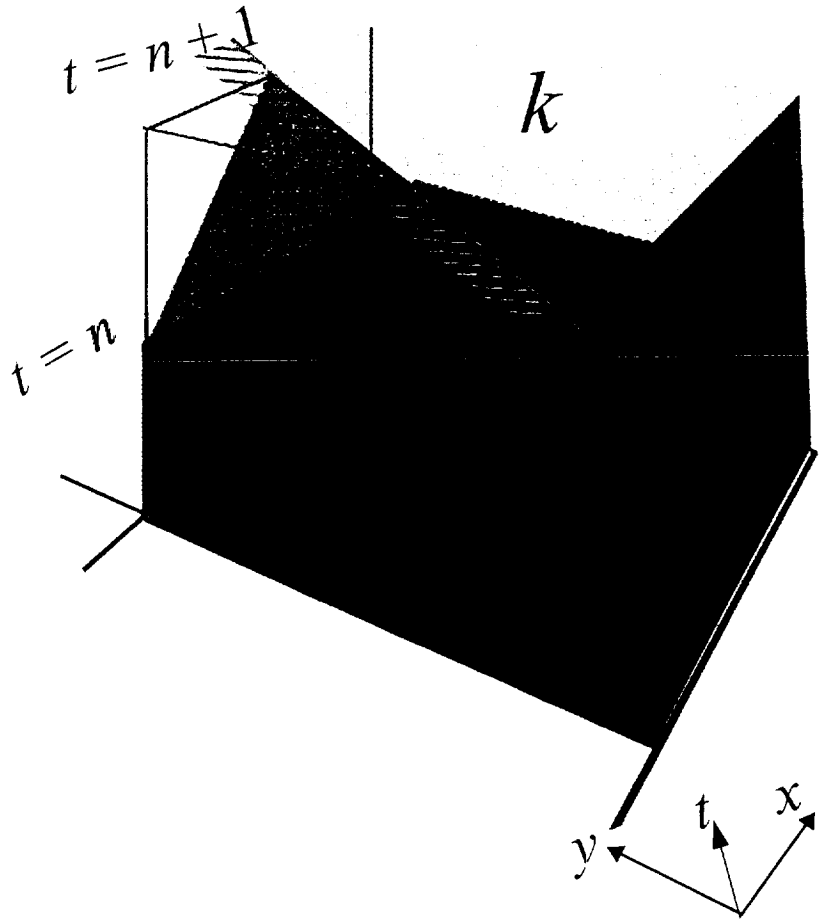



**Figure 7:** Isolated view of cell $k$ from the two-dimensional motion in Fig. 5.

Since the contribution to mass conservation due to an impermeable boundary is identically zero, this leaves only the pressure contribution to the momentum equations to be evaluated in the current scheme.

The lowest-fidelity approach for the space-time geometry is to simply ignore the time-dependent nature of the problem, and solve the governing equations with a steady-state solver at each time level. The steady-state solver can easily be augmented with a moving-wall boundary condition. While this sequential-static approach appears crude, there are many applications having timescales where this approach can be effective, and examples can be found in the literature. The advantage is that no specialized time-dependent or moving-body algorithms are required in order to perform the simulations, however the applicability and accuracy is limited (and unknown in general).

The sequential-static simulation approach can be improved by including a "history" of the fluid evolution, i.e. including the time derivative of the fluid state $\frac{\partial \mathbf{Q}}{\partial t}$. The simplest way to accomplish this is to use a staircase approximation to the space-time geometry, analogous to using a staircase geometry in space. The geometry is considered fixed over a timestep $[t^n, t^{n+1}]$, for example by holding the geometry in its state at $t^n$, or $t^{n+1/2}$, etc. In this manner the cell volume is held constant, so that a history of the motion is not provided, however the correct moving-wall boundary condition is still applied.

In the current scheme, the intersection of the geometry with a fixed Cartesian mesh is available at time levels $n$ and $n+1$ so that a history of the motion can be provided through the change in cell volume over a timestep. The integral of the flux through a face which is cut by the geometry during the timestep must still be evaluated however. As discussed in Sec. 4.2.1, in general the spatial flux through a space-time face is composed of both flow and wall contributions

$$\int_{t^n}^{t^{n+1}} \tilde{\mathbf{f}} \cdot \tilde{\mathbf{n}} d\Omega = \int_{t^n}^{t^{n+1}} \tilde{\mathbf{f}} \cdot \tilde{\mathbf{n}}_f d\Omega_f + \int_{t^n}^{t^{n+1}} \tilde{\mathbf{f}} \cdot \tilde{\mathbf{n}}_w d\Omega_w \tag{26}$$

where $\tilde{\mathbf{f}} = \mathbf{Q}\mathbf{u} + \mathbf{P}$ and the space-time normal $\tilde{\mathbf{n}}$ has elements in both space and time. The subscripts again refer to either a Cartesian flow face or a general impermeable boundary direction. Also note that in general the integrals are composed of multiple components. For example, the flow contribution on the $-x$ face in Fig. 7 can be broken into two integrals; one up to the vertex crossing and one after. The space-time flux terms in Eqn. 26 can be evaluated with a ($1^{\text{st}}$-order-in-time) backward-Euler quadrature, i.e.

$$\begin{aligned} \int_{t^n}^{t^{n+1}} \tilde{\mathbf{f}} \cdot \tilde{\mathbf{n}} d\Omega &\approx \hat{\mathbf{f}}^{n+1} \cdot \tilde{\mathbf{n}}_f \Omega_f^{n+1} + \hat{\mathbf{f}}_w^{n+1} \cdot \tilde{\mathbf{n}}_w^{n+1} \Omega_w^{n+1} \\ &\approx \hat{\mathbf{f}}^{n+1} \cdot \tilde{\mathbf{n}}_f S_f^{n+1} \Delta t + \hat{\mathbf{f}}_w^{n+1} \cdot \tilde{\mathbf{n}}_w^{n+1} S_w^{n+1} \Delta t \end{aligned} \tag{27}$$

In this manner the state of the flow at time $t^{n+1}$ is held over the entire timestep $[t^n, t^{n+1}]$. A $2^{\text{nd}}$-order spatial approximation for $\hat{\mathbf{f}}^{n+1}$ is used. The impermeable

moving-wall boundary condition is implemented in the wall flux term, i.e $\hat{\mathbf{f}}_w \cdot \tilde{\mathbf{n}}_w = \mathbf{P}_w \cdot \mathbf{n}_w$ (Eqn. 17). This provides a straightforward means to approximate the moving geometry and provide a history of the wall motion. Example results using this backward-Euler approximation will be presented in Sec. 5. Comparisons of the sequential-static and backward-Euler approximation for prescribed and 6-DOF motions are presented in [23] and [33] respectively.

An improvement can be made to the backward-Euler approximation, essentially at no cost, by improving the approximation to the space-time geometry. This approximate geometry approach is motivated by the observation that it isn't necessary to determine the complete details of the space-time geometry in order to implement a numerical scheme, it is only required to determine the space-time area for the spatial flux terms. If this area can be approximated in some manner (and an appropriate quadrature determined for the flux), an improvement in accuracy is possible. This approach is similar in spirit to using an agglomerated wall within the cut-cell geometry, rather than separately computing a contribution due to each polygonal contribution from the surface triangulation (cf. Aftosmis et al. [14]). A simple example is to approximate the flow portion of the space-time geometry using $\Omega_f \approx \frac{1}{2}\left(S_f^n + S_f^{n+1}\right)\Delta t$. The wall portion of the space-time geometry is then determined by forcing the space-time cell to close, as discussed above. Combining this approximation with a trapezoidal quadrature gives

$$\int_{t^n}^{t^{n+1}} \tilde{\mathbf{f}} \cdot \tilde{\mathbf{n}} d\Omega \approx \frac{1}{2}\left(\hat{\mathbf{f}}^n \cdot \tilde{\mathbf{n}}_f S_f^n + \hat{\mathbf{f}}^{n+1} \cdot \tilde{\mathbf{n}}_f S_f^{n+1}\right)\Delta t + \hat{\mathbf{f}}_w \cdot \tilde{\mathbf{n}}_w \hat{\Omega}_w \tag{28}$$

$\hat{\Omega}_w$ represents the approximation to the wall space-time area. More complex approximations for the space-time geometry are also possible. Evaluating the approximate space-time geometry approach for 3-D simulations is a focus of current research.

The highest-fidelity approach is to determine the actual space-time geometry. In 3-D, each spatial face is a 2-D area which evolves to a 3-D volume in space-time. If the motion of the moving boundary is planar in space-time, i.e. can be linearized, then the same techniques which are used to determine the cut-cell geometry for spatial meshes can be used to determine the space-time geometry of each spatial face. This requires that roughly six 3-D boundary/cell intersections must be computed for each Cartesian swept cell. While this approach is conceptually feasible, more experience is required to evaluate its utility.

# 5 Numerical Results

The previous sections described moving-boundaries within a Cartesian scheme, along with outlining implicit algorithms for solving such problems. The current section presents numerical results in one, two, and three dimensions to demonstrate the implicit approach. All results were obtained using the backward-Euler scheme dis-

cussed in the previous section. Before presenting the numerical results however, the full three-dimensional implementation is briefly discussed.

Currently, a fast global re-meshing is performed at each timestep with the same volume mesh generation package as used for static simulations. Work on incorporating solution and moving-geometry adaptation capability, similar to the static, steady-state method outlined in Aftosmis and Berger[34], is in progress. Note that if the motion is prescribed all the meshes can be processed a priori, and in parallel. Integrating the deforming-cell governing equations, Eqn. 6, for a representative cell $j$ using the backward-Euler scheme gives

$$(\mathbf{Q}V)_j^{n+1} - (\mathbf{Q}V)_j^n - \sum_{ahead} (\mathbf{Q}V)^n = -\Delta t \sum \left(\hat{\mathbf{f}} \cdot \mathbf{n} \Delta S\right)_j^{n+1} \tag{29}$$

The summation on the right side includes the wall and flow contributions within cut cells. This can be numerically integrated using the dual-time scheme outlined in Sec. 2. The terms $(\mathbf{Q}V)_j^n$ and $\sum_{ahead}(\mathbf{Q}V)^n$ become fixed source terms in the dual-time scheme. $(\mathbf{Q}V)^n$ however is only available on the mesh at time level $n$, while it is required on the mesh at time level $n+1$ in order to integrate Eqn. 29. $(\mathbf{Q}V)^n$ is conservatively transferred from the mesh at time level $n$ to the new mesh at $n+1$ external to the flow solver. The transfer of the solution between two volumes meshes takes advantage of the space-filling-curve ordering of the cells in the Cartesian meshes (cf. Ref. [15]). This allows the transfer to be performed very efficiently, requiring only two sweeps over the mesh cell list. A full discussion of this transfer algorithm is beyond the scope of the current work.

## 5.1 1-D Piston

A 1-D piston instantaneously moving at $M_p = 2.0$ into an initially quiescent fluid is simulated to demonstrate the conservation of the scheme. While the physical problem is one-dimensional, a 3-D mesh and solver are used, with suitable boundary conditions to ensure no lateral flow develops. The piston is originally centered at $x = 0$, and has width $4\Delta x$, where $\Delta x$ is the uncut hexahedron spacing. The piston moves in the $+x$ direction, and a shock forms ahead of the piston, with an expansion region to the rear (cf. Liepmann and Roshko Sec. 3.2[35]). If conservation is not maintained, a shock will not form ahead of the piston, or will have the wrong speed, as mass, momentum, and energy continually "leak" through the piston face. Numerical simulations moving the piston relative to a fixed domain are compared with the exact analytic solution. Figure 8 plots the pressure on the compression-side face of the piston as a function of time for two CFL numbers based upon the piston velocity, $\mathrm{CFL}_w = \frac{w_p \Delta t}{\Delta x} = 1.0$ and 10.0. After an initial transient, both numerical simulations reach the same pressure on the piston face, which is in agreement with the analytic solution. The pressure and density variation along the axis are plotted in Fig. 9 after the piston has traveled $80\Delta x$ and $320\Delta x$ for the $\mathrm{CFL}_w = 1.0$ and 10.0 simulations respectively. Both simulations predict the correct shock location ahead of the traveling piston, and the agreement in

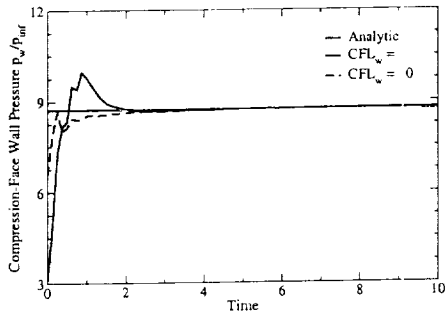
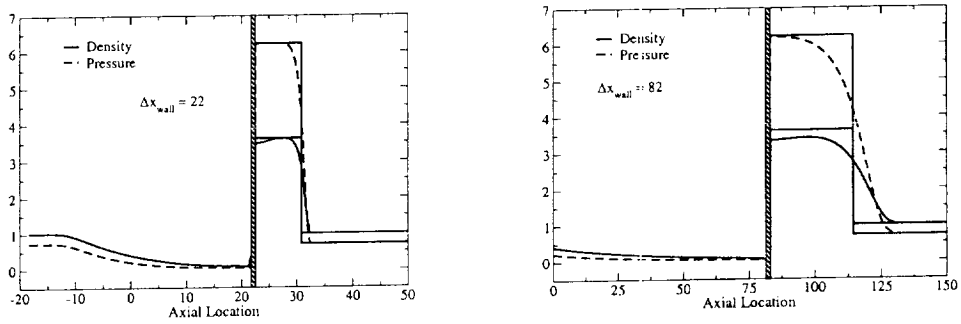



**Figure 8:** Pressure on the compression-side face of a piston moving at $M_p = 2.0$ into an initially quiescent fluid.




(a) $\mathrm{CFL}_w = \frac{w_p \Delta t}{\Delta x} = 1.0$ (b) $\mathrm{CFL}_w = \frac{w_p \Delta t}{\Delta x} = 10.0$

**Figure 9:** Density and pressure variation along the axis for a piston moving at $M_p = 2.0$ into an initially quiescent fluid. The mesh has a uniform spacing of $\Delta x = 0.25$.

the expansion region behind the shock is likewise very good. As expected, the shock is smeared over several cells using the backward-Euler time integration scheme, and the higher CFL results show greater dissipation near the shock than the CFL = 1.0 results.

## 5.2 2-D Oscillating Airfoil

The oscillating NACA 0012 airfoil presented in Sec. 3 is used to examine the behavior of the relative-motion scheme in 2-D. The experimental case was simulated using both the 2nd-order backward scheme with the moving-domain ALE scheme (cf. Sec. 3), and the 1st-order (in time) relative-motion scheme. Both schemes utilize the same spatially 2nd-order numerical flux formulation, and 100 timesteps per cycle were used in both simulations. The computed normal force variations with angle of attack are shown in Fig. 10. Both simulations capture the hysteresis caused by the unsteady shock formation, and are in good agreement with each other and the

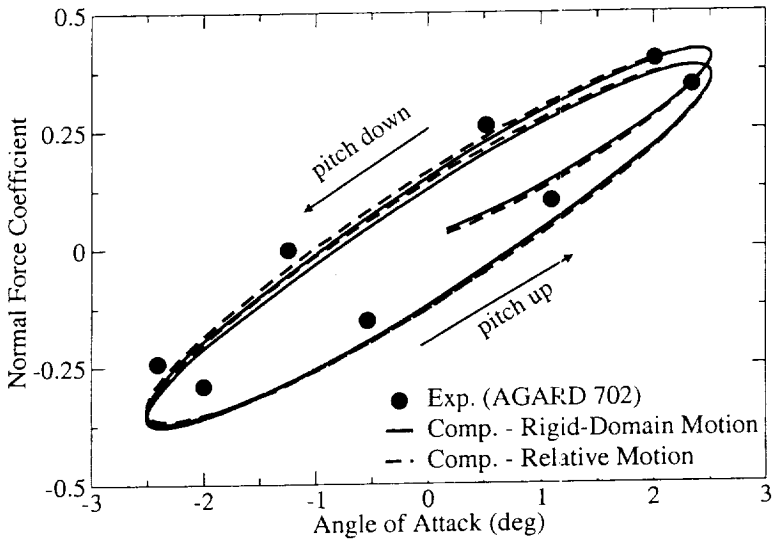



**Figure 10:** Variation of normal force coefficient with angle of attack for oscillating NACA 0012. Compare with pressure contours in Fig. 11. ($M_\infty = 0.755$, $\alpha(t) = 0.016 + 2.51\sin(2\pi 62.5t)$). The simulation uses 100 timesteps per complete cycle of the airfoil, and an isotropic mesh with a finest resolution of 0.001 chord. Experimental data from [26].

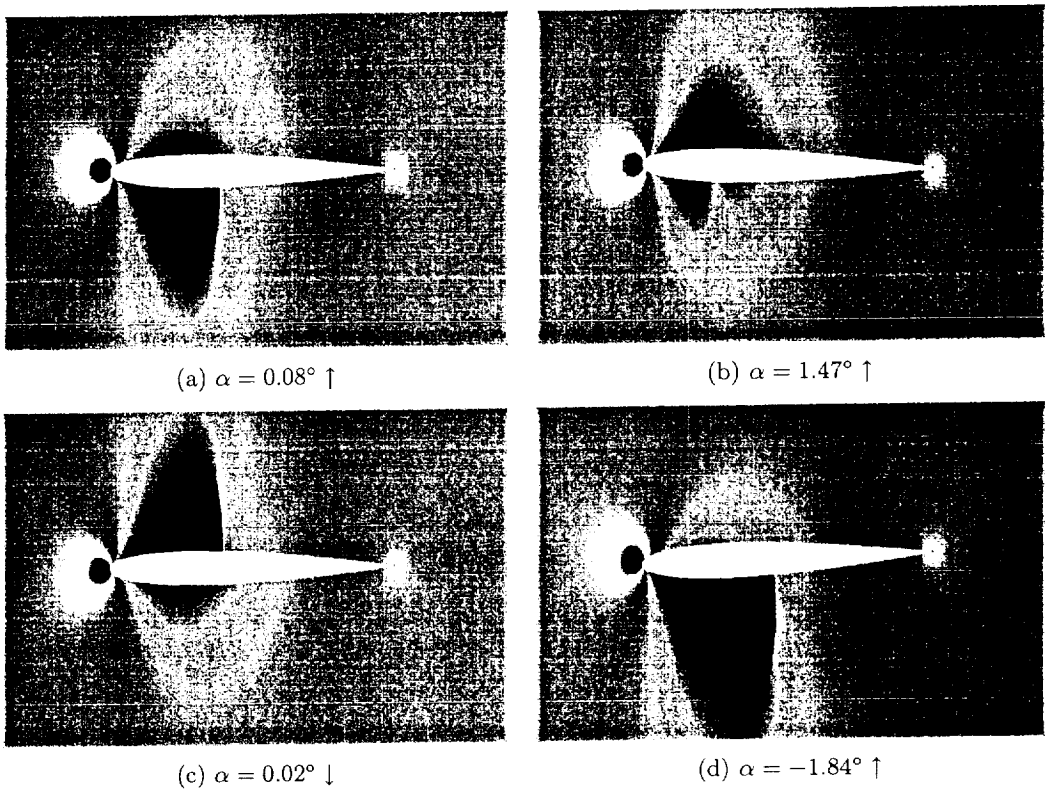
experimental data. The convergence and stability properties of the ALE and relative-motion formulations are similar for this problem. Using two coarsening levels of multigrid, each physical timestep converges roughly 2 orders of magnitude in the $L_1$-norm of density in 25 W-cycles using the dual-time formulation.

Snapshots of the 2-D pressure contours computed with the relative motion scheme are shown in Fig. 11 for the NACA 0012 oscillating airfoil. In general, the contours are smooth, with sharp definition of the shock location, i.e. no numerical artifacts from the relative motion scheme are visible. The hysteresis is evident comparing Fig. 11a and Fig. 11c, where the airfoil passes through zero angle of attack on the upstroke and downstroke respectively. The sonic "bump" which forms behind the upstream shock as it moves from the lower to upper surface is seen in frame (b). The maximum shock strength is shown in Fig. 11d, and the lag between maximum pitch angle and the angle of attack which produces the maximum shock strength is clear.

## 5.3 3-D Rolling Airframe

In order to determine the conservation correction term $\sum_{ahead}(\mathbf{Q}V)^n$ in multiple dimensions, it is necessary to simulate in some manner the physical convection process which is no longer discretized with a large timestep. Determining the conservation correction in 3-D, and the manner in which to distribute it, is an area of ongoing research. The 3-D results presented here do not include a conservation correction.

The complete three-dimensional time-dependent flow solver, using both a rigid-domain motion and localized relative motion, is demonstrated by simulating a rolling-airframe with dithering canards. The complete details of these numerical simulations



(a) $\alpha = 0.08° \uparrow$ (b) $\alpha = 1.47° \uparrow$

(c) $\alpha = 0.02° \downarrow$ (d) $\alpha = -1.84° \uparrow$

**Figure 11:** Snapshots of pressure during oscillation of NACA 0012 airfoil. Arrows in caption indicate whether airfoil is pitching nose up, or nose down. Compare with normal force variation in Fig. 10. ($M_\infty = 0.755$, $\alpha(t) = 0.016 + 2.51 \sin(2\pi 62.5t)$).

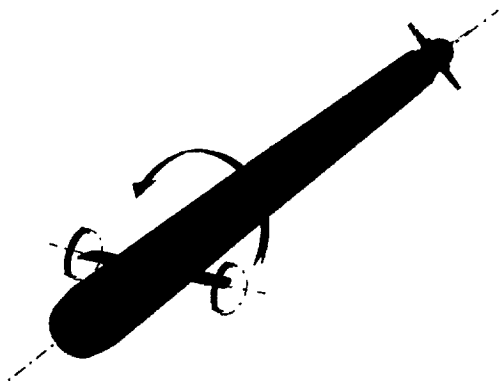
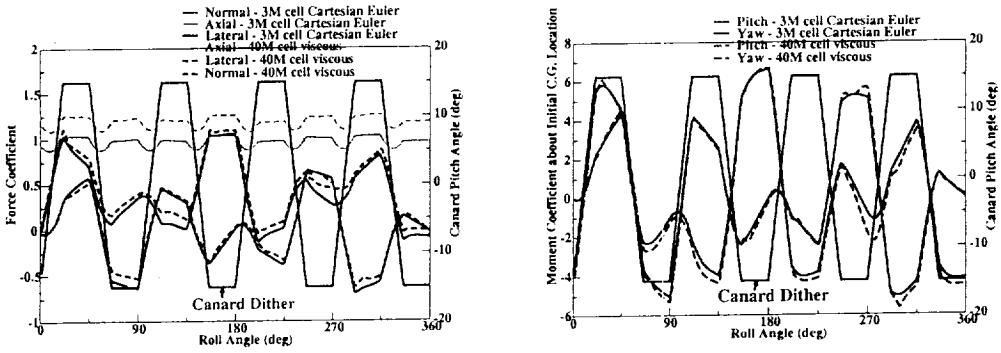


**Figure 12:** Rotating missile with dithering canards. The entire computational domain rotates at the body roll rate using the ALE formulation, while the canards rotate relative to the body using the relative-motion scheme.

are contained in [23], and only a brief overview will be presented here. The entire computational domain rotates with the body roll rate using the ALE formulation, while the canards concurrently rotate relative to the body using the relative-motion scheme. Again, only results for the backward-Euler relative motion scheme are presented. The missile body of Fig. 12 rotates at a constant prescribed rate of 8.75 Hz. As the body rolls, the two canards change positions synchronously to affect controlled movements, such as yaw or pitch. The computed force and moment variations with roll angle for one complete roll cycle are presented in Fig. 13, along with the canard deflection angles. The simulations use a mesh containing 3.4M cells, and a timestep which rolls the body 1° during a step (360 steps/cycle). The computed results are compared to high-resolution (40M cells, 10.000 steps/cycle) overset, viscous simulations of Nygaard and Meakin[36]. The current results compare well with the viscous results in terms of both roll-averaged values, and the roll-dependent variations. As expected, the viscous results predict a consistent axial force increment compared with the current inviscid results. Snapshots of the velocity magnitude at 5 axial cutting-planes along the body as the missile rolls are presented in Fig. 14. The canards change position from their maximum to minimum deflection through the three snapshots. The change in shock pattern on the canards as they pitch down, and the change in sense of rotation of the canard tip vortices are both visible. The twist in the canard tip vortices as the body rotates is also evident, though difficult to discern at this low spin rate.

Roll-averaged loads for the rolling airframe with dithering canards were measured experimentally[37]. Numerical simulations are compared to the experimental data in Table 2 for a body roll rate of 18Hz using the experimentally-measured canard dither schedule (cf. [23] for more details). The computed forces and moments are all within the 95% confidence level for the experimental data.




(a) Forces

(b) Moments

**Figure 13:** Force and moment comparison for rotating missile with dithering canards. The canard pitch angle is shown as a solid black line with scale at right. Viscous simulations by Nygaard and Meakin[36]. ($M_\infty = 1.6$, $\alpha = 3.0°$, $\dot{\phi} = 8.75$Hz).

| | $C_N$ | $C_Y$ | $C_l$ | $C_m$ | $C_n$ |
|---|---|---|---|---|---|
| Experiment | 0.45 – 0.61 | 0.15 – 0.20 | -0.036 – -0.019 | -1.5 – -0.40 | 0.93 – 1.5 |
| Computed | 0.55 | 0.20 | -0.034 | -0.48 | 1.46 |

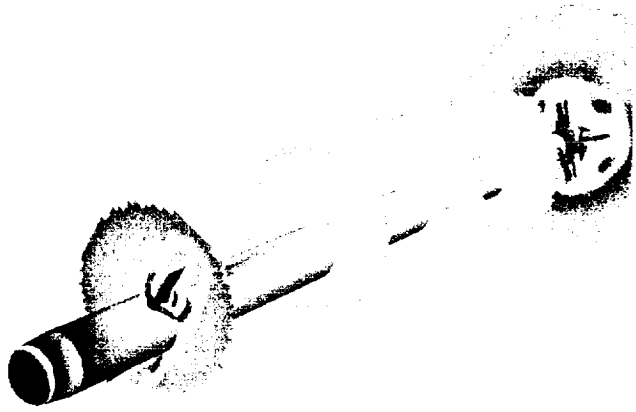
**Table 2:** Roll-averaged forces and moments for experimentally-measured dither schedule ($M_\infty = 1.6$, $\alpha = 3.0°$, $\dot{\phi} = 18$Hz). Experimental range corresponds to a 95% confidence level [37].

# 6 Summary

A method for simulating moving impermeable boundaries within a fixed Cartesian mesh has been developed and demonstrated. This scheme leverages the automated volume mesh generation process which has previously been demonstrated for static geometries. A major goal in this work is to limit the amount of geometry processing which is required during a complete simulation, in order to obtain an efficient and robust scheme. This is especially relevant for three-dimensional applications. An implicit dual-time method is used for the time advance, which allows a (large) timestep to be chosen based upon physical considerations and not stability restrictions. This limits the number of times the geometry must be intersected with the Cartesian volume mesh over a complete simulation. A general motion is decomposed into a rigid-body motion of the entire computational domain, with a relative-body motion superimposed. Since the rigid-domain motion can be treated using an ALE formulation, this confines the geometry processing only to the regions of relative motion within the domain.

The details of the relative-motion scheme are presented from a space-time analysis. This analysis presents the key ideas in 1-D, and features of relative motion in a Cartesian scheme are highlighted. A hierarchy of approximations to the boundary motion during a timestep are presented, along with preliminary results in one, two, and three dimensions. As a first step, only schemes which evaluate the geometry at

(a) $\phi = 45.5°$



(b) $\phi = 57.6°$

(c) $\phi = 71.8°$

**Figure 14:** Velocity magnitude contours for rotating missile with dithering canards. Red corresponds to a large magnitude, and blue low. ($M_\infty = 1.6$, $\alpha = 3.0^c$, $\dot{\phi} = 8.75$Hz).

two time-levels are considered for the relative motion.

## Acknowledgments

The authors would like to thank Dr. Tor Nygaard of ELORET and Dr. Robert Meakin of the U.S. Army AFDD for providing the viscous rolling-airframe results for comparison. Marsha Berger was supported by AFOSR grant F19620-00-0099 and DOE grants DE-FG02-00ER25053 and DE-FC02-01ER25472.

# References

[1] Meakin, R.L. and Suhs, N., "Unsteady Aerodynamic Simulation of Multiple Bodies in Relative Motion," AIAA Paper 89-1996-CP, June 1989.

[2] Batina, J., "Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex Aircraft Aeroelastic Analysis," AIAA Paper 89-1189, June 1989.

[3] Löhner, R., "Adaptive Remeshing for Transient Problems," *Computer Methods in Applied Mechanics and Engineering*, 75:195–214, 1989.

[4] Venkatakrishnan, V. and Mavriplis, D. J., "Computation of Unsteady Flows over Complex Geometries in Relative Motion," in *1st AFOSR Conference on Dynamic Motion CFD*, June 1996.

[5] Hirt, C.W. and Nichols, B.D., "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries," *Journal of Computational Physics*, 39:201–225, 1981.

[6] Osher, S. and Sethian, J.A., "Fronts Propagating with Curvature Dependent Speed: Algorithms based in Hamilton-Jacobi Formulations," *Journal of Computational Physics*, 79:12–49, 1988.

[7] Peskin, C.S., "Numerical Analysis of Blood Flow in the Heart," *Journal of Computational Physics*, 25:220–252, 1997.

[8] Forrer, H. and Berger, M.J., "Flow Simulations on Cartesian Grids Involving Complex Moving Geometries," in *7th International Conference on Hyperbolic Problems*, Zurich, Switzerland, 1998.

[9] Roma, A.M., Peskin, C.S., and Berger, M.J., "An Adaptive Version of the Immersed Boundary Method," *Journal of Computational Physics*, 153:509–534, 1999.

[10] Bayyuk, S.A., Powell, K.G., and van Leer, B., "A Simulation Technique for 2-D Unsteady Inviscid Flows around Arbitrarily Moving and Deforming Bodies of Arbitrary Geometry," AIAA Paper 93-3391-CP, 1993.

[11] Bayyuk, S.A., Powell, K.G., and van Leer, B., "Computation of Flows with Moving Boundaries and Fluid-structure Interactions," AIAA Paper 97-1771, 1997.

[12] Falcovitz, J., Alfandary, G., and Hanoch, G., "A Two-Dimensional Conservation Laws Scheme for Compressible Flows with Moving Boundaries," *Journal of Computational Physics*, 138:83–102, 1997.

[13] Lahur, P.R. and Nakamura, Y., "Simulation of Flow around Moving 3D Body on Unstructured Cartesian Grid," AIAA Paper 2001-2605, June 2001.

[14] Aftosmis, M.J., Berger, M.J., and Melton, J.E., "Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry," AIAA Paper 97-0196, Jan. 1997. Also *AIAA Journal* 36(6): 952–960,June 1998.

[15] Aftosmis, M.J., Berger, M.J., and Adomavicius, G., "A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries," AIAA Paper 2000-0808, Jan. 2000.

[16] Berger, M.J. and LeVeque, R., "Stable Boundary Conditions for Cartesian Grid Calculations," in *Symposium on Computational Technology for Flight Vehicles*, Pergamon Press, 1990. Also ICASE Report 90–37.

[17] Hirt, C.W., Amsden, A.A., and Cook, J.L., "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds," *Journal of Computational Physics*, 14: 227–253, 1974.

[18] Merkle, C.L. and Athavale, M., "A Time Accurate Unsteady Incompressible Algorithm Based on Artificial Compressibility," AIAA Paper 87-1137, June 1987.

[19] Rogers, S.E., Kwak, D., and Kiris, C., "Numerical Solution of the Incompressible Navier-Stokes Equations for Steady-State and Time-Dependent Problems," AIAA Paper 89-0463, 1989. Also *AIAA Journal* 29(4): 603–610, 1991.

[20] Jameson, A., "Time Dependent Calculations Using Multigrid with Applications to Unsteady Flows Past Airfoils," AIAA Paper 91-1596, June 1991.

[21] Melson, N.D., Sanetrik, M.D., and Atkins, H.L., "Time-accurate Navier-Stokes Calculations with Multigrid Acceleration," in *Proceedings of the Sixth Copper Mountain Conference on Multigrid Methods*, Apr. 1993.

[22] Beam, R.M. and Warming, R.F., "Implicit Numerical Methods for the Compressible Navier-Stokes and Euler Equations," VKI Lecture Series 81-01, Mar. 1981.

[23] Murman, S.M., Aftosmis, M.J., and Berger, M.J., "Numerical Simulation of Rolling-Airframes Using a Multi-Level Cartesian Method," AIAA Paper 2002-2798, June 2002.

[24] van Leer, B., "Flux-Vector Splitting for the Euler Equations," in *Lecture Notes in Physics*, vol. 170, pp. 507–512, Springer Verlag, Berlin, 1982.

[25] Agarwal, R.K. and Halt, D.W., "A Modified CUSP Scheme for Wave/Particle Split Form for Unstructured Grid Euler Flows," in *Frontiers of Computational Fluid Dynamics* (Caughey, D.A. and Hafez, M., ed.), pp. 155–168, John Wiley & Sons, 1994.

[26] Landon, R.H., "Compendium of Unsteady Aerodynamic Measurements," AGARD Report No. 702, Aug. 1982.

[27] Belk, D. and Maple, R., "Automated Assembly of Structured Grids for Moving Body Problems," AIAA Paper 95-1680-CP, June 1995.

[28] Löhner, R., Sharov, D., Luo, H., and Ramamurti, R, "Overlapping Unstructured Grids," AIAA Paper 2001-0439, Jan. 2001.

[29] Baker, T.J. and Cavallo, P.A., "Dynamic Adaptation for Deforming Tetrahedral Meshes," AIAA Paper 99-3253-CP, June 1999.

[30] Lesoinne, M. and 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*, 134:71–90, 1996.

[31] Thomas, P.D. and Lombard, C.K., "Geometric Conservation Law and its Application to Flow Computations on Moving Grids," *AIAA Journal*, 17:1030, 1979.

[32] Pember, R.B., Bell, J.B., Colella, P., Crutchfield, W.Y., and Welcome, M.L., "An Adaptive Cartesian Grid Method for Unsteady Compressible Flow in Irregular Regions," *Journal of Computational Physics*, 120:195–214, 1995.

[33] Murman, S.M., Aftosmis, M.J., and Berger, M.J., "Simulations of 6-DOF Motion with a Cartesian Method," AIAA Paper 2003-1246, Jan. 2003.

[34] Aftosmis, M.J. and Berger, M.J., "Multilevel Error Estimation and Adaptive h-Refinement for Cartesian Meshes with Embedded Boundaries," AIAA Paper 2002-0863, June 2002.

[35] Hans W. Liepmann and Anatol Roshko, *Elements of Gasdynamics*. John Wiley & Sons, 1960.

[36] Nygaard, T. and Meakin, R., "An Aerodynamic Analysis of a Spinning Missile with Dithering Canards," AIAA Paper 2002-2799-CP, June 2002.

[37] "Defensive Missile Wind Tunnel Test for the Validation and Verification of CFD Codes," Dynetics Technical Report, Jan. 2002.