INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 21, 807–835 (1995)

MIXED EXPLICIT/IMPLICIT TIME INTEGRATION OF COUPLED AEROELASTIC PROBLEMS: THREE-FIELD FORMULATION, GEOMETRIC CONSERVATION AND DISTRIBUTED SOLUTION

CHARBEL FARHAT, MICHEL LESOINNE AND NATHAN MAMAN

Department of Aerospace Engineering Sciences and Center for Aerospace Structures, University of Colorado at Boulder, Boulder, CO 80309-0429, U.S.A.

SUMMARY

A three-field arbitrary Lagrangian–Eulerian (ALE) finite element/volume formulation for coupled transient aeroelastic problems is presented. The description includes a rigorous derivation of a geometric conservation law for flow problems with moving boundaries and unstructured deformable meshes. The solution of the coupled governing equations with a mixed explicit (fluid)/implicit (structure) staggered procedure is discussed with particular reference to accuracy, stability, distributed computing, I/O transfers, subcycling and parallel processing. A general and flexible framework for implementing partitioned solution procedures for coupled aeroelastic problems on heterogeneous and/or parallel computational platforms is described. This framework and the explicit/implicit partitioned procedures are demonstrated with the numerical investigation on an iPSC-860 massively parallel processor of the instability of flat panels with infinite aspect ratio in supersonic airstreams.

KEY WORDS: CFD; structures; aeroelasticity; parallel processing

1. INTRODUCTION

In order to predict the dynamic response of a flexible structure in a fluid flow, the equations of motion of the structure and the fluid must be solved simultaneously. One difficulty in handling numerically the fluid/structure coupling stems from the fact that the structural equations are usually formulated with material (Lagrangian) co-ordinates while the fluid equations are typically written using spatial (Eulerian) co-ordinates. Therefore a straightforward approach to the solution of the coupled fluid/structure dynamic equations requires moving at each time step at least the portions of the fluid grid that are close to the moving structure. This can be appropriate for small displacements of the structure but may lead to severe grid distortions when the structure undergoes large motion. Several different approaches have emerged as alternatives to partial regridding in transient aeroelastic computations, among which we note the arbitrary Lagrangian/Eulerian (ALE) formulation,[1–3] the co-rotational approach,[4,5] dynamic meshes,[6] which are closely related to the ALE concept, and space–time formulations.[7] All these approaches treat a computational aeroelastic problem as a coupled two-field problem.

However, the moving mesh can be viewed as a pseudostructural system with its own dynamics[8] and therefore the coupled transient aeroelastic problem can be formulated as a three-rather than two-field problem: the fluid, the structure and the dynamic mesh. The semidiscrete equations governing this three-way coupled problem can be written as

CCC 0271–2091/95/100807–29

Received 17 June 1994
Revised 17 April 1995

$$\frac{\partial}{\partial t}[A(\xi,\ t)W(\xi,\ t)] + F^{c}(W(\xi,\ t),\ \xi,\ \dot{\xi}) = R(W(\xi,\ t)),$$

$$M\frac{\partial^2 q}{\partial t^2} + f^{int}(q) = f^{ext}(W(\xi,\ t)), \tag{1}$$

$$\tilde{M}\frac{\partial^2 \xi}{\partial t^2} + \tilde{D}\frac{\partial \xi}{\partial t} + \tilde{K}\xi = K_c q,$$

where t designates the time, ξ is the position of a moving fluid grid point, W is the fluid state vector, A results from the finite element/volume discretization of the fluid equations, F^c is the vector of convective ALE fluxes,[8] R is the vector of diffusive fluxes, q is the structural displacement vector, f^{int} denotes the vector of internal forces in the structure, f^{ext} is the vector of external forces, M is the finite element mass matrix of the structure, $\tilde{M}$, $\tilde{D}$ and $\tilde{K}$ are fictitious mass, damping and stiffness matrices associated with the fluid moving grid and constructed to avoid any parasitic interaction between the fluid and its grid or between the structure and the moving fluid grid,[8] and K_c is a transfer matrix that describes the action of the motion of the structural side of the fluid/structure interface on the fluid dynamic mesh. For example, $\tilde{M} = \tilde{D} = 0$ and $\tilde{K} = \tilde{K}^R$, where $\tilde{K}^R$ is a rotation matrix, correspond to a rigid mesh motion of the fluid grid around an oscillating aerofoil, and $\tilde{M} = \tilde{D} = 0$ includes the spring-based mesh motion scheme introduced in Reference 6 and the continuum-based updating strategy described in Reference 7 as particular cases.

Each of the three components of the coupled problem described by (1) has different mathematical and numerical properties and distinct software implementation requirements. For Euler and Navier–Stokes flows the fluid equations are non-linear. The structural equations and the semidiscrete equations governing the pseudostructural fluid grid system may be linear or non-linear. The matrices resulting from a linearization procedure are in general symmetric for the structural problem but are typically unsymmetric for the fluid problem. Moreover, the nature of the coupling in (1) is implicit rather than explicit, even when the fluid mesh motion is ignored. The fluid and the structure interact only at their interface via the pressure and the motion of the physical interface. However, for Euler and Navier–Stokes compressible flows the pressure variable cannot be easily isolated either from the fluid equations or from the fluid state vector W. Consequently, the numerical solution of (1) via a fully coupled monolithic scheme is computationally challenging and software-wise unmanageable.

Alternatively, equations (1) can be solved via partitioned procedures.[9–12] This approach offers several appealing features, including the ability to use well-established discretization and solution methods within each discipline, simplification of software development efforts and preservation of software modularity. Traditionally, transient aeroelastic problems have been solved via the simplest possible partitioned procedure whose cycle can be described as follows: (a) advance the structural system under a given pressure load; (b) update the fluid mesh accordingly; (c) advance the fluid system and compute a new pressure load.[13–16] Occasionally, some investigators have advocated the introduction of a few predictor/corrector iterations within each cycle of this three-step staggered integrator in order to improve accuracy,[17] especially when the fluid equations are non-linear and treated implicitly.[18] In this paper we focus on the design of a broader family of partitioned procedures where the fluid flow is integrated using an explicit scheme and the structural response is advanced using an implicit one. We address important issues pertaining to numerical stability, subcycling, accuracy versus speed trade-offs, implementation on heterogeneous computing platforms and interfield as well as intrafield parallel processing.

We begin in Section 2 with the discussion of a geometric conservation law for finite element and finite volume ALE fluid formulations and its implications on the time integration of the semidiscrete equations (1) that govern the three-way coupled aeroelastic problem. Next we introduce in Section 3 a partitioned solution procedure where the fluid flow is time integrated using an explicit scheme and the structural response is advanced using an implicit one. This particular choice of mixed time integration is motivated by the following facts: (a) the aeroelastic response of a structure is often dominated by low-frequency dynamics and therefore is most efficiently predicted by an implicit time integration scheme; (b) we have previously developed a massively parallel explicit finite element/volume Navier–Stokes solver[19–22] that we wish to reuse for aeroelastic computations. In practice the stability limit of this partitioned procedure has proved to be governed only by the critical time step of the explicit fluid solver. In Section 4 we describe a subcycling procedure that does not limit the stability properties of a partitioned time integrator. In Section 5 we address important issues related to interfield parallelism and design variants of the algorithm presented in Section 3 that allow simultaneous advancing of the fluid and structural systems. Section 6 focuses on the implementation of staggered procedures on distributed and/or heterogeneous computational platforms. Section 7 describes the application of the work presented in this paper to the numerical investigation of the instability of flat panels with infinite aspect ratio in supesonic airstreams. Finally, Section 8 concludes this paper.

Most of the theoretical foundation of the staggered time integrators discussed herein can be found in a companion paper.[12] In particular we refer the reader to Reference 12 for the theoretical proofs of the propositions made in this paper.

2. A GEOMETRIC CONSERVATION LAW FOR UNSTRUCTURED ALE METHODS

For the sake of brevity we present the mathematical derivations for the case of two-dimensional Euler flows and finite volume formulations and give only the final results for the case of three-dimensional problems.

2.1. A three-field ALE formulation

Let $\Omega \subset \mathbb{R}^2$ be the flow domain of interest and Γ be its moving and deforming boundary. The conservative law form of the equations describing Euler flows can be written in ALE form as

$$\left.\frac{\partial(JW)}{\partial t}\right|_a + J\nabla_\xi \cdot \mathscr{F}^c(W) = 0, \qquad \mathscr{F}^c(W) = \mathscr{F}(W) - \dot{\xi}W, \tag{2}$$

where a and ξ denote respectively the Lagrangian and instantaneous Eulerian positions of a fluid grid point, $J = \det(\mathrm{d}\xi/\mathrm{d}a)$ is the Jacobian of the frame transformation $a \to \xi$, W is the fluid state vector, $\mathscr{F}^c$ denotes the convective ALE fluxes and $\dot{\xi}$ is the ALE grid velocity, which may be different from the fluid velocity and from zero. Whether the finite volume or finite element method is used for the spatial discretization of (2), the resulting semidiscrete equation can be written as

$$\frac{\partial}{\partial t}[A(\xi, t)W(\xi, t)] + F^c(W(\xi, t), \xi, \dot{\xi}) = 0, \tag{3}$$

where A denotes the area of a cell or a finite element and F^c denotes the numerical approximation of the convective ALE fluxes.

Several procedures have been proposed for updating the ALE variables ξ and $\dot{\xi}$, most of which can be summarized as viewing the fluid domain (or its grid) as a pseudostructural system governed by

$$\tilde{M}\ddot{\xi} + \tilde{D}\dot{\xi} + \tilde{K}\xi = \tilde{f}, \tag{4}$$

where a dot superscript designates a time derivative, $\tilde{M}$, $\tilde{D}$ and $\tilde{K}$ are fictitious mass, damping and stiffness matrices and $\tilde{f}$ is a vector of generalized forces that controls the fluid grid motion. For example, Batina's[6] spring-based mesh updating scheme corresponds to setting $\tilde{M}=\tilde{D}=0$ and attaching a fictitious spring between each pair of connected grid points which determines $\tilde{K}$. Alternatively, $\tilde{M}$ and $\tilde{D}$ can be constructed from $\tilde{K}$ as

$$\tilde{M} = \alpha\tilde{K}, \qquad \tilde{D} = \beta\tilde{K}, \tag{5}$$

where the two scalars α and β are such that the pseudostructural system (4) is critically damped. In all cases equation (4) is usually driven by enforcing the continuity between the grid motion and the structural displacement and/or velocity at the fluid/structure boundary:

$$\xi(t) = q(t) \quad \text{on} \quad \Gamma_{F/S} \subset \Gamma, \qquad \dot{\xi}(t) = \dot{q}(t) \quad \text{on} \quad \Gamma_{F/S} \subset \Gamma. \tag{6}$$

Therefore $\tilde{f}$ can be written in general as

$$\tilde{f}(t) = K_c q(t), \tag{7}$$

where K_c is a transfer matrix that describes the action of the motion of the structural side of the fluid/structure interface on the fluid dynamic mesh and $q(t)$ is determined from the structural equation of dynamic equilibrium

$$M\ddot{q} + f^{\text{int}}(q) = f^{\text{ext}}(W(\xi, t)). \tag{8}$$

In general the governing parameters of $\tilde{K}$, e.g. the spring coefficients in Batina's scheme,[6] are chosen and optimized to avoid mesh distortion as much as possible. However, this is not sufficient for correctly computing flow problems with moving boundaries and deformable meshes via an ALE algorithm. Next we show that the ALE fluid grid must obey a geometric conservation law (GCL) equivalent to that developed in Reference 23 for moving body-fitted curvilinear grids and the Beam–Warming[24] finite difference scheme.

2.2. *Conservation law for the grid displacement and velocity fields*

Let $\mathcal{T}_h$ be a standard triangulation of Ω. A vertex of a triangle T is denoted by S_i and the set of its neighbouring vertices by $K(i)$. At each vertex S_i a cell C_i is constructed as the union of the subtriangles resulting from the subdivision by means of the medians of each triangle of $\mathcal{T}_h$ that is connected to S_i

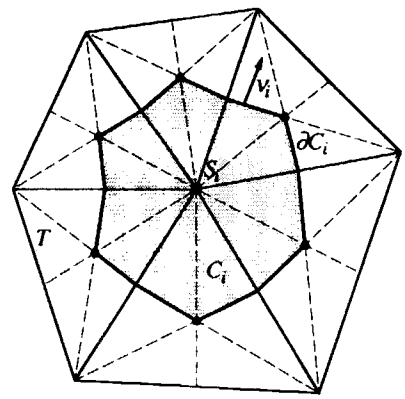

Figure 1. Control volume in an unstructured mesh

(Figure 1). The boundary of C_i is denoted by ∂C_i and the unit vector of the outward normal to ∂C_i by v_i. The union of all these N_c control volumes constitutes a discretization of the domain Ω:

$$\Omega = \bigcup_{i=1}^{N_c} C_i.$$

Let also Δt and $t^n = n\Delta t$ denote respectively the chosen time step and the nth time station. Integrating equation (3) between t^n and t^{n-1} leads to

$$\int_{t^n}^{t^{n+1}} \frac{\partial}{\partial t}[A(\xi,\ t)W(\xi,\ t)]\ \mathrm{d}t + \int_{t^n}^{t^{n+1}} F^c(W(\xi,\ t),\ \xi,\ \dot{\xi})\ \mathrm{d}t$$

$$= A(\xi^{n+1})W^{n+1} - A(\xi^n)W^n + \int_{t^n}^{t^{n+1}} F^c(W(\xi,\ t),\ \xi,\ \dot{\xi})\ \mathrm{d}t = 0. \tag{9}$$

In a finite volume formulation A is a diagonal matrix that contains the areas (volumes) A_i of the cells and F^c results from the spatial integration of the convective fluxes on the cell boundaries. For each cell C_i, F_i^c can be written as

$$F_i^c(W,\ \dot{\xi}) = \sum_{j\in K(i)} \int_{\partial C_{i,j}(\xi)} [\mathscr{F}^+(W_i) + \mathscr{F}^-(W_j) - \dot{\xi}W] \cdot v_{i,j}\ \mathrm{d}s, \tag{10}$$

where $\mathscr{F}^+$ and $\mathscr{F}^-$ designate the classical flux splitting[25] in the absence of a moving mesh and are given by

$$\mathscr{F}^+ = \frac{\mathscr{F} + |\mathscr{F}|}{2}, \qquad \mathscr{F}^- = \frac{\mathscr{F} - |\mathscr{F}|}{2}. \tag{11}$$

The most important issue in the solution of (1) via an ALE method is the proper evaluation of $\int_{t^n}^{t^{n+1}} F^c(W(\xi,\ t),\ \xi,\ \dot{\xi})\ \mathrm{d}t$ in (9). In particular it is crucial to establish where the fluxes must be integrated: on the mesh configuration at $t = t^n$ (ξ^n), on that at $t = t^{n+1}$ (ξ^{n+1}) or in between these two configurations.

Let W^* denote a given uniform state of the flow. Clearly a proposed solution method cannot be acceptable unless it conserves a uniform flow. Substituting $W^n = W^{n+1} = W^*$ in (9) and rewriting this equation at the cell level gives

$$[A_i(\xi^{n+1}) - A_i(\xi^n)]W^* + \int_{t^n}^{t^{n+1}} F_i^c(W^*,\ \xi,\ \dot{\xi})\ \mathrm{d}t = 0. \tag{12}$$

From (10) it follows that

$$F_i^c(W^*,\ \dot{\xi}) = \sum_{j\in K(i)} \int_{\partial C_{i,j}(\xi)} [\mathscr{F}^+(W^*) + \mathscr{F}^-(W^*) - \dot{\xi}W^*] \cdot v_{i,j}\ \mathrm{d}s = \int_{\partial C_i(\xi)} [\mathscr{F}(W^*) - \dot{\xi}W^*] \cdot v_i\ \mathrm{d}s. \tag{13}$$

Given that the integral on a closed boundary of the flux of a constant function is identically zero, i.e.

$$\int_{\partial C_i(\xi)} \mathscr{F}(W^*) \cdot v_i\ \mathrm{d}s = 0, \tag{14}$$

it follows that

$$F_i^c(W^*,\ \dot{\xi}) = -\int_{\partial C_i(\xi)} \dot{\xi}W^* \cdot v_i\ \mathrm{d}s. \tag{15}$$

Hence substituting (15) into (12) yields

$$[A_i(\xi^{n+1}) - A_i(\xi^n)]W^* - \left(\int_{t^n}^{t^{n+1}} \dot{\xi} \cdot \nu_i \, \mathrm{d}s\right) W^* = 0, \tag{16}$$

which can be rewritten as

$$\boxed{A_i(\xi^{n+1}) - A_i(\xi^n) = \int_{t^n}^{t^{n+1}} \int_{\partial C_i(\xi)} \dot{\xi} \cdot \nu_i \, \mathrm{d}s \, \mathrm{d}t.} \tag{17}$$

Equation (17) defines a GCL that must be verified by any proposed ALE mesh updating scheme. This law states that the change in area (volume) of each control volume between t^n and t^{n+1} must be equal to the area (volume) swept by the cell boundary during $\Delta t = t^{n+1} - t^n$ (Figure 2). Therefore the updating of ξ and $\dot{\xi}$ cannot be based on mesh distortion issues alone when using AL solution schemes.

2.3. *Implications of the GCL*

From the analysis presented in the previous subsection, it follows that an appropriate scheme for evaluating $\int_{t^n}^{t^{n+1}} F^c(W(\xi, t), \xi, \dot{\xi}) \, \mathrm{d}t$ in (9) is a scheme that respects the GCL (17). Note that once a mesh updating scheme is given, the left-hand side of (17) is always exactly computed. Hence a proper method for evaluating $\int_{t^n}^{t^{n+1}} F^c(W(\xi, t), \xi, \dot{\xi}) \, \mathrm{d}t$ is a method that obeys the GCL and therefore computes exactly the right-hand side of (17), i.e. $\int_{t^n}^{t^{n+1}} \int_{\partial C_i(\xi)} \dot{\xi} \cdot \nu_i \, \mathrm{d}s \, \mathrm{d}t$. Given that $\partial C_i(\xi)$ is the union of segments, it suffices to consider the integration of $\dot{\xi} \cdot \nu_i$ along a segment $[ab]$:

$$I_{[ab]} = \int_{t^n}^{t^{n+1}} \int_{[ab]} \dot{\xi} \cdot \nu \, \mathrm{d}s \, \mathrm{d}t. \tag{18}$$

Let ξ_a and ξ_b denote the instantaneous positions of two connected vertices a and b (Figure 3). The position of any point on the edge $[ab]$ during the time interval $[t^n, t^{n+1}]$ can be parametrized as:

$$\xi(t) = \gamma\xi_a(t) + (1-\gamma)\xi_b(t), \quad \gamma \in [0, 1], \quad t \in [t^n, t^{n+1}], \quad \dot{\xi}(t) = \gamma\dot{\xi}_a(t) + (1-\gamma)\dot{\xi}_b(t), \tag{19}$$

where

$$\xi_a(t) = \delta(t)\xi_a^{n+1} + [1-\delta(t)]\xi_a^n, \quad t \in [t^n, t^{n+1}], \quad \xi_b(t) = \delta(t)\xi_b^{n+1} + [1-\delta(t)]\xi_b^n \tag{20}$$

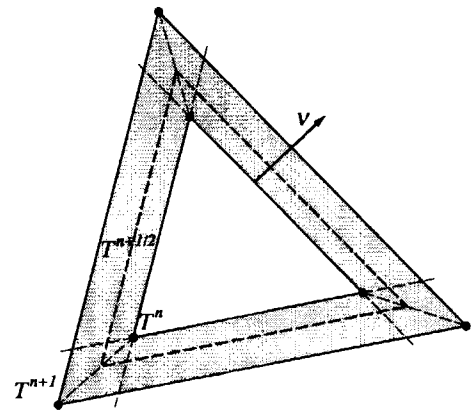

Figure 2. GCL for ALE formulations (two-dimensional case)

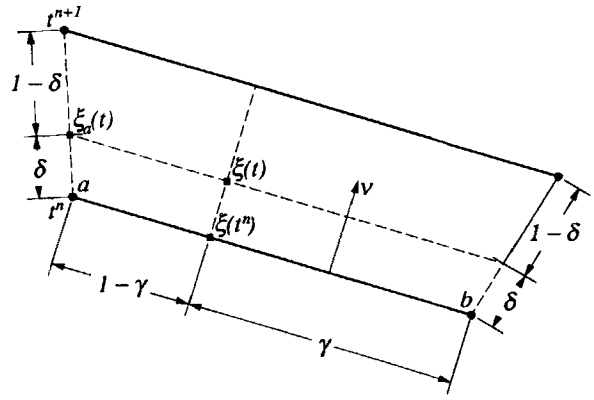

Figure 3. Parametrization of a grid edge

and $\delta(t)$ is a real function that satisfies

$$\delta(t^n) = 0, \qquad \delta(t^{n+1}) = 1. \tag{21}$$

Substituting (19) and (20) into (18) yields

$$\begin{aligned} I_{[ab]} &= \int_{t^n}^{t^{n+1}} \int_0^1 [\gamma\dot{\xi}_a + (1-\gamma)\dot{\xi}_b] \cdot vl \, d\gamma \, dt \\ &= \int_{t^n}^{t^{n+1}} \tfrac{1}{2}(\dot{\xi}_a + \dot{\xi}_b) \cdot vl \, dt \\ &= \int_{t^n}^{t^{n+1}} \tfrac{1}{2}(\dot{\xi}_a + \dot{\xi})H(\xi_a - \xi_b) \, dt \\ &= \int_{t^n}^{t^{n+1}} \tfrac{1}{2}(\dot{\xi}_a + \dot{\xi}_b)H[\delta(\xi_a^{n+1} - \xi_b^{n+1}) + (1-\delta)(\xi_a^n - \xi_b^n)] \, dt, \end{aligned} \tag{22}$$

where l is the length of the edge $[ab]$ and

$$H = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

The mesh velocities $\dot{\xi}_a$ and $\dot{\xi}_b$ can be obtained from the differentiation of (20) as

$$\dot{\xi}_a = \dot{\delta}(t)(\xi_a^{n+1} - \xi_a^n), \qquad \dot{\xi}_b = \dot{\delta}(t)(\xi_b^{n+1} - \xi_b^n) \tag{23}$$

and $I_{[ab]}$ can be finally written as

$$\begin{aligned} I_{[ab]} &= \tfrac{1}{2}\int_{t^n}^{t^{n+1}} \dot{\delta}[(\xi_a^{n+1} - \xi_a^n) + (\xi_b^{n+1} - \xi_b^n)]H[\delta(\xi_a^{n+1} - \xi_b^{n+1}) + (1-\delta)(\xi_a^n - \xi_b^n)] \, dt \\ &= \tfrac{1}{2}\int_0^1 [(\xi_a^{n+1} - \xi_a^n) + (\xi_b^{n+1} - \xi_b^n)]H[\delta(\xi_a^{n+1} - \xi_b^{n+1}) + (1-\delta)(\xi_a^n - \xi_b^n)] \, d\delta. \end{aligned} \tag{24}$$

Clearly the integrand of $I_{[ab]}$ is linear in δ. Therefore $I_{[ab]}$ can be exactly computed using the *midpoint rule* provided that equations (23) hold, i.e.

$$\dot{\xi} = \dot{\delta}(t)(\xi^{n+1} - \xi_n) = \frac{\Delta\delta}{\Delta t}(\xi^{n+1} - \xi^n), \tag{25}$$

which in view of (21) can also be written as

$$\boxed{\dot{\xi} = \frac{\xi^{n+1} - \xi^n}{\Delta t}} \tag{26}$$

In summary, the GCL derived herein shows that for two-dimensional problems the integrand of $\int_{t^n}^{t^{n+1}} F^c(W(\xi, t), \xi, \dot{\xi})\, dt$ in (9) must be evaluated at the midpoint configuration and this integral must be computed as

$$\boxed{\begin{gathered}\int_{t^n}^{t^{n+1}} F^c(W(\xi, t), \xi, \dot{\xi})\, dt = \Delta t F^c(W^k, \xi^{n+1/2}, \dot{\xi}^{n+1/2}), \\ W^{n+1/2} = \frac{W^n + W^{n+1}}{2}, \qquad \xi^{n+1/2} = \frac{\xi^n + \xi^{n+1}}{2}, \qquad \dot{\xi}^{n+1/2} = \frac{\xi^{n+1} - \xi^n}{\Delta t},\end{gathered}} \tag{27}$$

where the superscript k depends on the time discretization of the fluid flow equation.

Similarly it can be shown that in the three-dimensional case the integrand of $I_{[ab]}$, where $[abc]$ denotes a facet of a tetrahedron (Figure 4), is quadratic in δ and therefore $I_{[abc]}$ can be exactly computed using the *two-point rule* provided that equations (23) hold. Hence the only method for evaluating $\int_{t^n}^{t^{n+1}} F^c(W(\xi, t), \xi, \dot{\xi})\, dt$ that respects the GCL (17) in the three-dimensional case is

$$\boxed{\begin{gathered}\int_{t^n}^{t^{n+1}} F^c(W(\xi, t), \xi, \dot{\xi})\, dt = \frac{\Delta t}{2}[F^c(W^{k1}, \xi^{m1}, \dot{\xi}^{n+1/2}) + F^c(W^{k2}, \xi^{m2}, \dot{\xi}^{n+1/2})], \\ m1 = n + \tfrac{1}{2} = 1/2\sqrt{3}, \qquad m2 = n + \tfrac{1}{2} + 1/2\sqrt{3}, \\ W^{n+\zeta} = \zeta W^{n+1} + (1-\zeta)W^n, \qquad \xi^{n+\zeta} = \zeta\xi^{n+1} + (1-\zeta)\xi^n, \qquad \dot{\xi}^{n+1/2} = \frac{\xi^{n+1} - \xi^n}{\Delta t},\end{gathered}} \tag{28}$$

where the superscripts $k1$ and $k2$ depend on the time discretization of the fluid flow equation.

Note that the proper mesh configurations and computational procedures for evaluating $\int_{t^n}^{t^{n+1}} F^c(W(\xi, t), \xi, \dot{\xi})\, dt$ are not the same for two- and three-dimensional problems (equations (27) and (28)). However, in both cases the proper evaluation of the mesh velocity field is the same:

$$\dot{\xi}^{n+1/2} = \frac{\xi^{n+1} - \xi^n}{\Delta t}. \tag{29}$$

The above formula for updating $\dot{\xi}^{n+1/2}$ is intuitive and has certainly been 'naturally' used by several investigators when $\tilde{M} = \tilde{D} = 0$, independently from any conservation law issue (see e.g. Reference 6). However, when the pseudostructural fluid grid system is characterized by $M \neq 0$ and/or $\tilde{D} \neq 0$, the mesh velocity field $\tilde{\dot{\xi}}^{n+1/2}$ computed by the time integrator applied to the solution of (4) is not guaranteed to be equal to $\dot{\xi}^{n+1/2} = (\xi^{n+1} - \xi^n)/\Delta t$. In that case, satisfying the GCL requires

(a) using the mesh velocity $\tilde{\dot{\xi}}^{n+1/2}$ computed by the time integrator applied to (4) only for evaluating ξ^{n+1}.

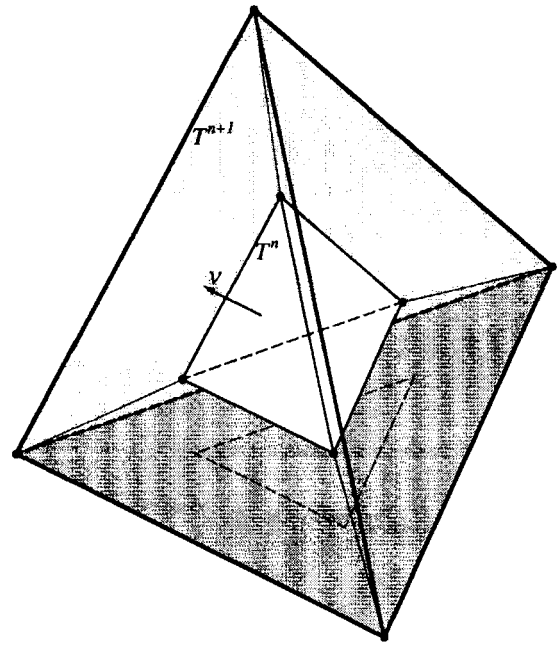

Figure 4. GCL for ALE formulations (three-dimensional case)

(b) using the mesh velocity $\dot{\xi}^{n+1/2} = (\xi^{n+1} - \xi^n)/\Delta t$ in the evaluation of $\int_{t^n}^{t^{n+1}} F^c(W(\xi, t), \xi, \dot{\xi})\, dt$ via (27) or (28).

On the other hand, when a pseudostructural fluid grid system is constructed with $\tilde{M} = \tilde{D} = 0$ (e.g. Batina's network of springs[6]), it is not always possible to satisfy both continuity equations (6) without violating the GCL. For example, if the displacement continuity condition $\xi(t) = q(t)$ is enforced at the fluid/structure interface $\Gamma_{F/S}$, which is usually the case, then respecting the GCL implies computing a mesh velocity field on $\Gamma_{F/S}$ that is equal to

$$\dot{\xi}^{n+1/2} = \frac{\xi^{n+1} - \xi^n}{\Delta t} = \frac{q^{n+1} - q^n}{\Delta t} \quad \text{on} \quad \Gamma_{F/S}. \tag{30}$$

In that case, satisfying also the velocity continuity condition $\dot{\xi}(t) = \dot{q}(t)$ on $\Gamma_{F/S}$ requires that

$$\dot{q}^{n+1/2} = \frac{q^{n+1} - q^n}{\Delta t} \quad \text{on} \quad \Gamma_{F/S}, \tag{31}$$

which is not enforced by all structural time integrators. Therefore it is not always possible to satisfy the continuity between both the displacement and velocity of the structure and those of the fluid grid without violating the GCL.

Unfortunately, a discontinuity between the velocity of the structure and that of the fluid mesh at the fluid/structure interface can perturb the energy exchange between the fluid and the structure. Luckily, it can be shown[26] that when the implicit midpoint rule is used for advancing the structure and the displacement condition $\xi(t) = q(t)$ is enforced on $\Gamma_{F/S}$, the following equalities hold:

$$\dot{\xi}^{n+1/2} = \frac{\xi^{n+1} - \xi^n}{\Delta t} = \frac{q^{n+1} - q^n}{\Delta t} = \dot{q}^{n+1/2} \quad \text{on} \quad \Gamma_{F/S}. \tag{32}$$

Hence, if the structural equations are integrated with the implicit midpoint rule, it becomes possible to satisfy both continuity equations (6) while still respecting the GCL. Among others, this important result justifies the choice of the trapezoidal rule as the implicit time integrator for the low-frequency-dominated structural equations (8).

Finally, in order to highlight the impact of the GCL on transient coupled aeroelastic computations, we report in Figure 5 two computed histories of the non-dimensional lift for an elastic panel in a transonic regime (freestream Mach number $M_\infty = 0{\cdot}84$) using an ALE solution scheme that satisfies the GCL and one that violates it at the fluid/structure interface. In both cases the time integration is carried out with an explicit (fluid)/implicit (structure) partitioned procedure and a time step that respects the stability limit of the coupled aeroelastic problem. For this example, violating the GCL is shown to introduce undesirable spurious oscillations in the lift prediction.

3. A STAGGERED EXPLICIT/IMPLICIT TIME INTEGRATOR

3.1. Background

When the structure undergoes small displacements, the fluid mesh can be frozen and 'transpiration' fluxes can be introduced at the fluid side of the fluid/structure boundary to account for the motion of the structure. In that case the transient aeroelastic problem is simplified from a three- to a two-field coupled problem. Furthermore, if the structure is assumed to remain in the linear regime and the fluid flow is linearized around an equilibrium position W_0 (note that most fluid/structure instability problems are analysed by investigating the response of the coupled system to a perturbation around a steady state), the semidiscrete equations governing the coupled aeroelastic problem become (see Reference 12 for details)

$$\begin{pmatrix} \delta\dot{W} \\ Q \end{pmatrix} = \begin{pmatrix} A^* & B \\ C & D^* \end{pmatrix} \begin{pmatrix} \delta W \\ Q \end{pmatrix}, \qquad \begin{pmatrix} \delta W \\ Q \end{pmatrix}_{(t=0)} = \begin{pmatrix} \delta W \\ Q \end{pmatrix}_0 \tag{33}$$

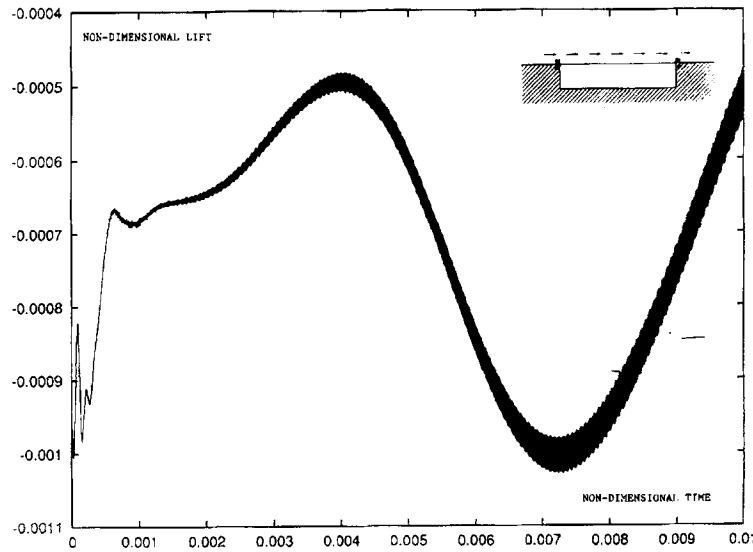

Figure 5. Flat panel in a transonic regime: effect of GCL on lift

Here δW is the perturbed fluid state vector,

$$Q = \begin{pmatrix} q \\ \dot{q} \end{pmatrix}$$

is the structure state vector, A^* results from the spatial discretization of the flow equations (2), B is the matrix induced by the transpiration fluxes at the fluid/structure boundary $\Gamma_{F/S}$, C is the matrix that transforms the fluid pressure on $\Gamma_{F/S}$ into prescribed structural forces and

$$D^* = \begin{pmatrix} 0 & I \\ -M^{-1}K & -M^{-1}D \end{pmatrix},$$

where M, D and K are the structural mass, damping and stiffness matrices.

A mathematical discussion of the time integration of (33) via implicit/implicit and explicit/implicit partitioned procedures can be found in Reference 12. Here we focus on the more general three-way coupled aeroelastic problem (1). Based on the mathematical results established in Reference 12 for solving (33), we design a family of explicit/implicit staggered procedures for time integrating (1) and address important issues pertaining to accuracy, stability, distributed computing, I/O transfers, subcycling and parallel processing.

3.2. ALG0: a partitioned explicit/implicit algorithm for the three-field formulation

For simplicity and without any loss of generality we focus on the case of two-dimensional Euler flows and linear structural vibrations. For three-dimensional inviscid flows, equations (28) should be used instead of equations (27). From the results established in Section 3, it follows that the semidiscrete equations governing the three-way coupled aeroelastic problem can be written in that case as

$$\begin{gathered} A(\xi^{n+1})W^{n+1} - A(\xi^n)W^n + \Delta t F^c(W^k,\ \xi^{n+1/2},\ \dot{\xi}^{n+1/2}) = 0, \\ \xi^{n+1/2} = \frac{\xi^n + \xi^{n+1}}{2}, \qquad \dot{\xi}^{n+1/2} = \frac{\xi^{n+1} - \xi^n}{\Delta t}, \\ M\ddot{q}^{n+1} + D\dot{q}^{n+1} + Kq^{n+1} = f^{\text{ext}}(W^{n+1}(\xi,\ t)), \\ \tilde{M}\ddot{\xi}^{n+1} + \tilde{D}\dot{\xi}^{n+1} + \tilde{K}\xi^{n+1} = K_c q^{n+1}, \end{gathered} \tag{34}$$

where the superscript k depends on the time discretization of the fluid flow equations.

In many aeroelastic problems, such as wing flutter, a steady flow is first computed around a structure in equilibrium. Next the structure is perturbed via an initial displacement and/or velocity and the aeroelastic response of the coupled fluid/structure system is analysed. This suggests that a natural sequencing for the staggered time integration of (34) is as follows.

1. Perturb the structure via some initial conditions.
2. Update the fluid grid to conform to the new structural boundary.
3. Advance the flow with the new boundary conditions.
4. Advance the structure with the new pressure load.
5. Repeat from step (2) until the goal of the simulation is reached.

An important feature of partitioned solution procedures is that they allow the use of existing single-discipline software modules. In this effort we are particularly interested in reusing the massively parallel explicit flow solver described in References 19 and 20. Therefore we select to time integrate the semidiscrete fluid equations with a three-step variant of the explicit Runge–Kutta algorithm. On the other hand, the aeroelastic response of a structure is often dominated by low-frequency dynamics. Hence the structural equations are most efficiently solved by an implicit time integration scheme. Here we select to time integrate the structural motion with the implicit midpoint rule because it allows the enforcement of both continuity equations (6) while still respecting the GCL (see Section 2).[26] Consequently, we propose the following explicit/implicit solution algorithm for solving the three-field coupled problem (34):

Given: a steady flow and initial conditions for the structure

1. Update the fluid grid to conform to the new structural boundary:

$$\text{Solve } \tilde{M}\ddot{\xi}^{n+1} + \tilde{D}\dot{\xi}^{n+1} + \tilde{K}\xi^{n+1} = K_c q^n$$

$$\xi^{n+1/2} = \frac{\xi^n + \xi^{n+1}}{2}$$

$$\dot{\xi}^{n+1/2} = \frac{\xi^{n+1} - \xi^n}{\Delta t}$$

2. Advance the fluid system using RK3:

$$\begin{aligned} W_i^{n+1(0)} &= W_i^n \\ W_i^{n+1(k)} &= \frac{A_i(\xi^n)}{A_i(\xi^{n+1})} W_i^{n+1(0)} \\ &\quad + \frac{1}{A_i(\xi^{n+1})} \frac{1}{4-k} \Delta t F_i^c(W^{(k-1)}, \xi^{n+1/2}, \dot{\xi}^{n+1/2}), \quad k = 1, 2, 3 \\ W_i^{n+1} &= W_i^{n+1(3)} \end{aligned} \tag{35}$$

3. Advance the structural system using the midpoint rule:

$$M\ddot{q}^{n+1} + D\dot{q}^{n+1} + Kq^{n+1} = f^{\text{ext}}(W^{n+1})$$

$$q^{n+1} = q^n + \frac{\Delta t}{2}(\dot{q}^n + \dot{q}^{n+1})$$

$$\dot{q}^{n+1} = \dot{q}^n + \frac{\Delta t}{2}(\ddot{q}^n + \ddot{q}^{n+1})$$

In the sequel we refer to the above explicit/implicit staggered procedure as ALG0. It is depicted graphically in Figure 6. Extensive experiments with this solution procedure have shown that its stability limit is governed by the critical time step of the explicit fluid solver (and therefore is not worse than that of the underlying fluid explicit time integrator).

The three-step Runge–Kutta algorithm is third-order-accurate for linear problems and second-order-accurate for non-linear ones. The midpoint rule is second-order-accurate. A simple Taylor expansion shows that the partitioned procedure ALG0 is first-order-accurate when applied to the linearized equations (33). When applied to (34), its accuracy depends on the solution scheme selected for solving

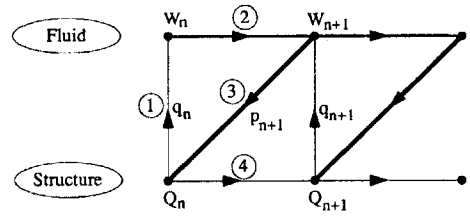

Figure 6. ALG0: basic staggered procedure

the mesh equations (4). As long as the time integrator applied to (4) is consistent, ALG0 is guaranteed to be at least first-order-accurate.

4. SUBCYCLING

The fluid and structure fields often have different time scales. For problems in aeroelasticity the fluid flow usually requires a smaller temporal resolution that the structural vibration. Therefore, if ALG0 is used to solve (34), the coupling time step Δt_c will be typically dictated by the stability time step of the fluid system, Δt_F, and not the time step $\Delta t_S > \Delta t_F$ that meets the accuracy requirements of the structural field.

Using the same time step Δt in both fluid and structure computational kernels presents only minor implementational advantages. On the other hand, subcycling the fluid computations with a factor $n_{S/F} = \Delta t_S / \Delta t_F$ can offer substantial computational advantages, including

(a) savings in the overall simulation CPU time, because in that case the structural field will be advanced fewer times
(b) savings in I/O transfers and/or communication costs when computing on a heterogeneous platform, because in that case the fluid and structure kernels will exchange information fewer times.

However, the computational advantages highlighted above are effective only if subcycling does not restrict the stability region of the staggered algorithm to values of the coupling time step Δt_c that are small enough to offset these advantages. In Reference 12 it is shown that for the linearized problem (33) the straightforward conventional subcycling procedure—that is, the scheme where at the end of each $n_{F/S}$ fluid subcycles only the interface pressure computed during the last fluid subcycle is transmitted to the structure—lowers the stability limit of ALG0 to a value that is less than the critical time step of the fluid explicit time integrator. On the other hand, it is also shown in Reference 12 that when solving (33), the stability limit of ALG0 can be preserved if (a) the deformation of the fluid mesh between t^n and t^{n+1} is evenly distributed among the $n_{S/F}$ subcycles and (b) at the end of each $n_{S/F}$ fluid subcycles the average interface pressure field $\bar{p}_{\Gamma_{F/S}}$ computed during the subcycles between t^n and t^{n+1} is transmitted to the structure. Hence we propose the following explicit/implicit fluid-subcycled partitioned procedure for solving (34):

1. Construct $\{q^{n^{(s)}}\}_{s=1}^{s=n_{S/F}}$ such that $q^{(n^{S/F})} = q^n$

$\bar{p}_{\Gamma_{F/S}} = 0$

For $s = 1, \ldots, n_{S/F}$

{

1a. Update in stages the fluid grid to conform to the new structural boundary:

$$\text{Solve } \tilde{M}\ddot{\xi}^{n+1^{(s)}} + \tilde{D}\dot{\xi}^{n+1^{(s)}} + \tilde{K}\xi^{n+1^{(s)}} = K_c q^{n^{(s)}}$$

$$\xi^{n+1/2^{(s)}} = \frac{\xi^{n^{(s)}} + \xi^{n+1^{(s)}}}{2}$$

$$\dot{\xi}^{n+1/2^{(s)}} = \frac{\xi^{n+1^{(s)}} - \xi^{n^{(s)}}}{\Delta t}$$

2. Advance the fluid system using RK3:

$$W_i^{n+1^{(0)^{(s)}}} = W_i^{n^{(s)}}$$

$$w_i^{n+1^{(k)^{(s)}}} = \frac{A_i(\xi^{n^{(s)}})}{A_i(\xi^{n+1^{(s)}})} W_i^{n+1^{(0)^{(s)}}} + \frac{\Delta t F_i^c(W^{(k-1)^{(s)}}, \xi^{n+1/2^{(s)}}, \dot{\xi}^{n+1/2}(s))}{(4-k)A_i(\xi^{n+1^{(s)}})} \quad k = 1, 2, 3 \qquad (36)$$

$$W_i^{n+1^{(s)}} = W_i^{n+1^{(3)^{(s)}}}$$

$$\bar{p}_{\Gamma_{F/s}} = \bar{p}_{\Gamma_{F/S}} + p_{\Gamma_{F/S}}^{n+1^{(s)}}$$

}

$\bar{p}_{\Gamma_{F/S}}^{n+1} = \bar{p}_{\Gamma_{F/S}} / n_{S/F}$

3. Advance the structural system using the midpoint rule:

$$M\ddot{q}^{n+1} + D\dot{q}^{n+1} + Kq^{n+1} = f^{ext}(\bar{p}_{\Gamma_{F/S}}^{n+1})$$

$$q^{n+1} = q^n + n_{S/F}\Delta t \frac{\dot{q}^n + \dot{q}^{n+1}}{2}$$

$$\dot{q}^{n+1} = \dot{q}^n + n_{S/F}\Delta t \frac{\ddot{q}^n + \ddot{q}^{n+1}}{2}$$

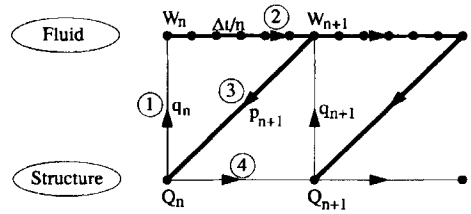

Figure 7. ALG1: subcycling

In the sequel we refer to the above explicit/implicit fluid-subcycled staggered procedure as ALG1. It is depicted graphically in Figure 7. Extensive numerical experiments have shown that for small values of $n_{S/F}$ the stability limit of ALG1 is governed by the critical time step of the explicit fluid solver.

However, experience has also shown that there exists a maximum subcycling factor beyond which ALG1 becomes numerically unstable.

From the theory developed in Reference 12 for the linearized equations (33), it follows that ALG1 is first-order-accurate and that, as one would have expected, subcycling amplifies the fluid errors by the factor $n_{S/F}$.

5. INTERFIELD PARALLELISM

ALG0 and ALG1 are inherently sequential. In both partitioned procedures the fluid system must be updated before the structural system can be advanced. Of course, ALG0 and ALG1 allow intrafield parallelism (parallel computations within each discipline) but inhibit interfield parallelism. Advancing the fluid and structural systems simultaneously is appealing because it can reduce the total simulation time.

A simple variant ALG2 of ALG1 (or ALG0 if subcycling is not desired) that allows interfield parallel processing is given below:

1. Construct $q^{n(s)}{}_{s=1}^{s=n_{S/F}}$ such that $q^{(n_{S/F})} = q^n$

$\bar{p}_{\Gamma_{F/S}} = 0$

For $s = 1, \ldots, n_{S/F}$

{

1a. Update in stages the fluid grid to conform to the new structural boundary:

$$Solve \quad \tilde{M}\ddot{\xi}^{n+1^{(s)}} + \tilde{D}\dot{\xi}^{n+1^{(s)}} + \tilde{K}\xi^{n+1^{(s)}} = K_c q^{n^{(s)}}$$

$$\xi^{n+1/2^{(s)}} = \frac{\xi^{n^{(s)}} + \xi^{n+1^{(s)}}}{2}$$

$$\dot{\xi}^{n+1/2^{(s)}} = \frac{\xi^{n+1^{(s)}} - \xi^{n^{(s)}}}{\Delta t}$$

2. Advance the fluid system using RK3:

$$W_i^{n+1^{(0)(s)}} = W_i^{n^{(s)}}$$

$$W_i^{n+1^{(k)(s)}} = \frac{A_i(\xi^{n^{(s)}})}{A_i(\xi^{n+1^{(s)}})} W_i^{n+1^{(0)(s)}} + \frac{\Delta t F_i^c(W^{(k-1)^{(s)}}, \xi^{n+1/2^{(s)}}, \dot{\xi}^{n+1/2^{(s)}})}{(4-k)A_i(\xi^{n+1^{(s)}})} \quad k = 1, 2, 3 \tag{37}$$

$$W_i^{n+1^{(s)}} = W_i^{n+1^{(3)(s)}}$$

$$\bar{p}_{\Gamma_{F/S}} = \bar{p}_{\Gamma_{F/S}} + p^{n+1^{(s)}}_{\Gamma_{F/S}}$$

}

$\bar{p}^{n+1}_{\Gamma_{F/S}} = \bar{p}_{\Gamma_{F/S}}/n_{S/F}$

3. Advance the structural system using the midpoint rule:

$$M\ddot{q}^{n+1} + D\dot{q}^{n+1} + Kq^{n+1} = f^{ext}(\bar{p}^{n}_{\Gamma_{F/S}})$$

$$q^{n+1} = q^n + n_{S/F}\Delta t \frac{\dot{q}^n + \dot{q}^{n+1}}{2}$$

$$\dot{q}^{n+1} = \dot{q}^n + n_{S/F}\Delta t \frac{\dot{q}^n + \dot{q}^{n+1}}{2}$$

Clearly the fluid and structure kernels can run in parallel during the time interval $[t_n, t_{n+n_{S/F}}]$. Interfield communication or I/O transfer is needed only at the beginning of each time interval.

The basic steps of ALG2 are depicted graphically in Figure 8. The theory developed in Reference 12 shows that for the linearized equations (33), ALG2 is first-order-accurate and parallelism in ALG2 is achieved at the expense of amplified errors in the fluid and structure responses.

In order to improve the accuracy of the basic parallel time integratgor ALG2, we propose to exchange information between the fluid and structure kernels at half-step in the following specific manner (ALG3):

$\bar{p}_{\Gamma_{F/S}} = 0$

For $s = 1, \ldots, n_{S/F}/2 - 1$

{

Solve $\tilde{M}\ddot{\xi}^{n+1^{(s)}} + \tilde{D}\dot{\xi}^{n+1^{(s)}} + \tilde{K}\xi^{n+1^{(s)}} = K_c q^{n^{(s)}}$

$$\xi^{n+1/2^{(s)}} = \frac{\xi^{n^{(s)}} + \xi^{n+1^{(s)}}}{2}$$

$$\dot{\xi}^{n+1/2^{(s)}} = \frac{\xi^{n+1^{(s)}} - \xi^{n^{(s)}}}{\Delta t}$$

$$W_i^{n+1^{(0)^{(s)}}} = W_i^{n^{(s)}}$$

$$W_i^{n+1^{(k)^{(s)}}} = \frac{A_i(\xi^{n^{(s)}})}{A_i(\xi^{n+1^{(s)}})} W_i^{n+1^{(0)^{(s)}}} + \frac{\Delta t F_i^c(W^{(k-1)^{(s)}}, \xi^{n+1/2^{(s)}}, \dot{\xi}^{n+1/2^{(s)}})}{(4-k)A_i(\xi^{n+1^{(s)}})} \quad k = 1, 2, 3$$

$$W_i^{n+1^{(s)}} = W_i^{n+1^{(3)^{(s)}}}$$

$$\bar{p}_{\Gamma_{F/S}} = \bar{p}_{\Gamma_{F/S}} + p^{n+1^{(s)}}_{\Gamma_{F/S}}$$

}

$$\bar{p}_{\Gamma_{F/S}}^{n+1/2} = \bar{p}_{\Gamma_{F/S}}/(n_{S/F}/2)$$

$$M\ddot{\bar{q}}^{n+1} + D\dot{\bar{q}}^{n+1} + K\bar{q}^{n+1} = f^{ext}(\bar{p}_{\Gamma_{F/S}}^{n+1/2})$$

$$\bar{q}^{n+1} = q^n + n_{S/F}\Delta t \frac{\dot{q}^n + \dot{\bar{q}}^{n+1}}{2}$$

$$\dot{\bar{q}}^{n+1} = \dot{q}^n + n_{S/F}\Delta t \frac{\ddot{q}^n + \ddot{\bar{q}}^{n+1}}{2} \tag{38}$$

$$\bar{\Gamma}_{F/S} = 0$$

For $s = n_{S/F}/2, \ldots, n_{S/F}$

{

Solve $\tilde{M}\ddot{\xi}^{n+1^{(s)}} + \tilde{D}\dot{\xi}^{n+1^{(s)}} + \tilde{K}\xi^{n+1^{(s)}} = K_c\bar{q}^{n+1^{(s)}}$

$$\xi^{n+1/2^{(s)}} = \frac{\xi^{n^{(s)}} + \xi^{n+1^{(s)}}}{2}$$

$$\dot{\xi}^{n+1/2^{(s)}} = \frac{\xi^{n+1^{(s)}} - \xi^{n^{(s)}}}{\Delta t}$$

$$W_i^{n+1^{(0)(s)}} = W_i^{n^{(s)}}$$

$$W_i^{n+1^{(k)(s)}} = \frac{A_i(\xi^{n^{(s)}})}{A_i(\xi^{n+1^{(s)}})} W_i^{n+1^{(0)(s)}} + \frac{\Delta t F_i^c(W^{(k-1)^{(s)}}, \xi^{n+1/2^{(s)}}, \dot{\xi}^{n+1/2^{(s)}})}{(4-k)A_i(\xi^{n+1^{(s)}})} \quad k = 1, 2, 3$$

$$W_i^{n+1^{(s)}} = W_i^{n+1^{(3)(s)}}$$

$$\bar{p}_{\Gamma_{F/S}} = \bar{p}_{\Gamma_{F/S}} + p_{\Gamma_{F/S}}^{n-1^{(s)}}$$

}

$$\bar{p}_{\Gamma_{F/S}}^{n+1} = \bar{p}_{\Gamma_{F/S}}/(n_{S/F}/2)$$

$$M\ddot{q}^{n+1} + D\dot{q}^{n+1} + Kq^{n+1} = f^{ext}(\bar{p}_{\Gamma_{F/S}}^{n+1}$$

$$q^{n+1} = q^n + n_{S/F}\Delta t \frac{\dot{q}^n + \dot{q}^{n+1}}{2}$$

$$\dot{q}^{n+1} = \dot{q}^n + n_{S/F}\Delta t \frac{\ddot{q}^n + \ddot{q}^{n+1}}{2}$$

ALG3 is illustrated in Figure 9. The first half of the computation is identical with that of ALG2, except that the fluid system is subcycled only up to $t^{n+n_{S/F}}/2$ while the structure is advanced in one shot up to $t^{n+n_{S/F}}$. At $t^{n+n_{S/F}}/2$ the fluid and structure kernels exchange pressure, displacement and velocity information. In the second half of the computation the fluid system is subcycled from $t^{n+n_{S/F}}/2$ to $t^{n+n_{S/F}}$ using the new structural information and the structural behaviour is recomputed in parallel using the newly received pressure distribution. Note that the first evaluation of the structural state vector $\bar{Q}^{n+1}$ can be interpreted as a predictor.

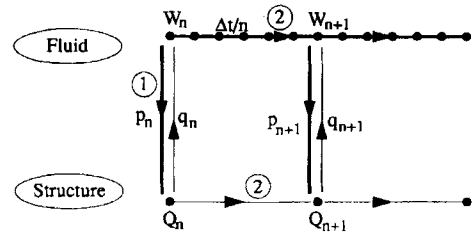

Figure 8. ALG2: subcycling and interfield parallelism

It can be shown that when applied to the linearized equations (33), ALG3 is first-order-accurate and reduces the errors of ALG2 by the factor $n_{S/F}$ at the expense of one additional communication step or I/O transfer during each coupled cycle (see Reference 12 for a detailed error analysis).

6. IMPLEMENTATIONAL ISSUES

6.1. *Subcycling*

We have pointed out in Section 4 that when subcycling is desired, the deformation of the fluid mesh between t^n and t^{n+1} should not be entirely applied during the first fluid subcycle, but evenly distributed across all subcycling stages. There are many ways this can be accomplished, including the following one.

At the beginning of time step t^{n+1} the fluid code has access to the component of the structural state vector $(q^n, \dot{q}^n)_{\Gamma_{F/S}}$ that relates to the degrees of freedom located at the fluid/structure interface. The objective of any mesh updating strategy is to exploit this information and compute a fluid mesh position ξ^{n+1} that satisfies the continuity equations (6):

$$\xi^{n+1} = q^n \quad \text{on} \quad \Gamma_{F/S}, \qquad \dot{\xi}^{n+1} = \dot{q}^n \quad \text{on} \quad \Gamma_{F/S}. \tag{39}$$

At this point it should be noted that the difference in the superscripts between the left- and right-hand sides of (39) is due to the staggered nature of the solution scheme and that the second equation of (39) should be enforced only if it does not violate the GCL. Using equations (39) as constraints, an updated fluid mesh position ξ^{n+1} can be obtained e.g. via the solution of (4). Then at every subcycling stage a new constraint can be generated for computing the subcycled mesh position $\xi^{n+1^{(2)}}_{\Gamma_{F/S}}$ via the solution of (4) as follows:

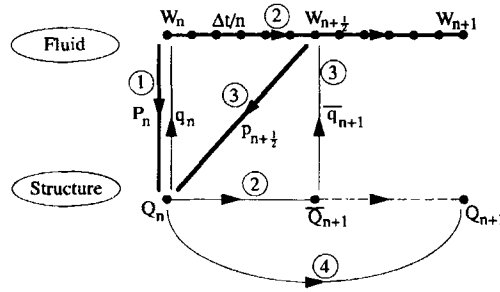

Figure 9. ALG3: subcycling, interfield parallelism and improved accuracy

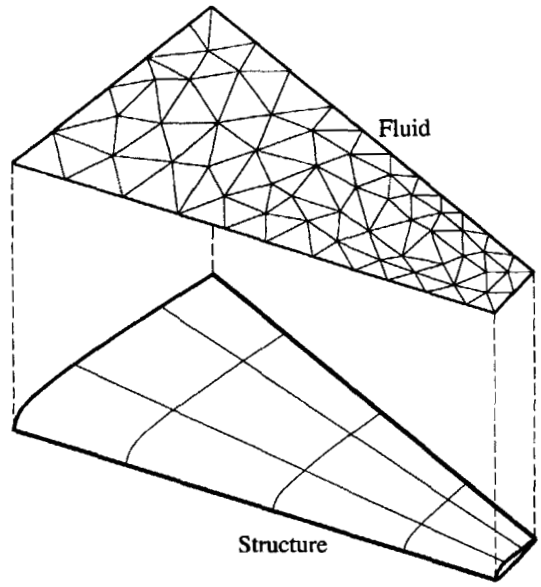

Figure 10. Incompatible fluid and structural meshes

$$\begin{aligned}
&\xi^{n+1^{(0)}}_{\Gamma_{F/S}} = q^{n-1}_{\Gamma_{F/S}},\\
&\xi^{n+1^{(s)}}_{\Gamma_{F/S}} = I^d(\xi^{n+1^{(s-1)}}_{\Gamma_{F/S}},\ q^n_{\Gamma_{F/S}},\ \dot{q}^n_{\Gamma_{F/S}},\ \Delta t), \quad s = 1,\ \ldots,\ n_{F/S}-1,\\
&\xi^{n+1^{(n_{F/S})}}_{\Gamma_{F/S}} = q^n_{\Gamma_{F/S}},
\end{aligned} \tag{40}$$

where I^d is an interpolation scheme of order $d = 0, 1, 2$. More specifically, I^d is given by and leads to

$$\begin{aligned}
\xi^{n+1^{(s)}}_{\Gamma_{F/S}} &= q^n_{\Gamma_{F/S}} \quad (d=0),\\
\xi^{n+1^{(s)}}_{\Gamma_{F/S}} &= \xi^{n+1^{(s-1)}}_{\Gamma_{F/S}} + \frac{q^n_{\Gamma_{F/S}} - \xi^{n+1^{(s-1)}}_{\Gamma_{F/S}}}{n_{F/S}-s+1} \quad (d=1),\\
\xi^{n+1^{(s)}}_{\Gamma_{F/S}} &= \xi^{n+1^{(s-1)}}_{\Gamma_{F/S}} + \frac{2q^n_{\Gamma_{F/S}} - 2\xi^{n+1^{(s-1)}}_{\Gamma_{F/S}} - (n_{F/S}-s+1)\Delta t\dot{q}^n_{\Gamma_{F/S}}}{n_{F/S}-s+1}\\
&\quad + \frac{\dot{q}^n_{\Gamma_{F/S}} - (n_{F/S}-s+1)\Delta t - q^n_{\Gamma_{F/S}} + \xi^{n+1^{(s-1)}}_{\Gamma_{F/S}}}{(n_{F/S}-s+1)^2} \quad (d=2).
\end{aligned} \tag{41}$$

In practice we have found that the $I^1(\xi^{n+1^{(s-1)}}_{\Gamma_{F/S}},\ q^n_{\Gamma_{F/S}})$ interpolation scheme is the best choice. This scheme does not require the transmission of any structural velocity information to the fluid computational kernel. In all cases the fluid mesh velocity $\dot{\xi}$ is computed via (26) to satisfy the GCL.

6.2. *Incompatible mesh interfaces*

In general the fluid and structure meshes have two independent representations of the physical fluid/structure interface. When these representations are identical, e.g. when every fluid grid point on $\Gamma_{F/S}$ is

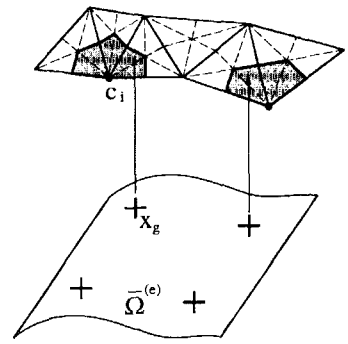

Figure 11. Gauss point–fluid cell pairing

also a structural node and vice versa, the evaluation of the pressure forces and the transfer of the structural motion to the fluid mesh are trivial operations. However, analysts usually prefer to

(a) use a fluid mesh and a structural model that have been independently designed and validated
(b) refine each mesh independently from the other.

Hence most realistic aeroelastic simulations will involve handling fluid and structural meshes that are incompatible at their interface boundaries (Figure 10). In Reference 27 we have addressed this issue and proposed a preprocessing 'matching' procedure that does not introduce any approximation other than those intrinsic to the fluid and structure solution methods. This procedure can be summarized as follows.

The nodal forces induced by the fluid pressure on the 'wet' surface of a structural element e can be written as

$$F_i = -\int_{\bar{\Omega}^{(e)}} N_i p v \, d\sigma, \tag{42}$$

where $\bar{\Omega}^{(e)}$ denotes the geometrical support of the wet surface of the structural element e, p is the pressure field, v is the unit normal to $\bar{\Omega}^{(e)}$ and N_i is the shape function associated with node i. Most if

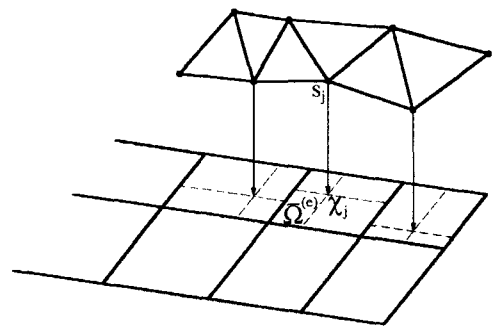

Figure 12. Fluid grid point–wet structural element pairing

not all finite element structural codes evaluate the integral in (42) via a quadrature rule

$$F_i = -\sum_{g=1}^{g=n_g} w_g N_i(X_g) p(X_g), \tag{43}$$

where w_g is the weight of the Gauss point X_g. Hence a structural code needs to know the values of the pressure field only at the Gauss points of its wet surface. This information can be easily made available once every Gauss point of a wet structural element is paired with a fluid cell (Figure 11). It should be noted that in (43) the X_g are not necessarily the same Gauss points as those used for stiffness evaluation. For example, if a high pressure gradient is anticipated over a certain wet area of the structure, a larger number of Gauss points can be used for the evaluation of the pressure forces F_i on that area.

On the other hand, when the structure moves and/or deforms, the motion of the fluid grid points on $\Gamma_{F/S}$ can be prescribed via the regular finite element interpolation

$$\xi(S_j) = \sum_{k=1}^{k=wne} N_k(\chi_j) q_k^{(e)}, \tag{44}$$

where S_j, wne, χ_j and q_k denote respectively a fluid grid point on $\Gamma_{F/S}$, the number of wet nodes in its 'nearest' structural element e, the natural co-ordinates of S_j in $\bar{\Omega}^{(e)}$ and the structural displacement at the kth node of element e. From (44) it follows that each fluid grid point on $\Gamma_{F/S}$ must be matched with one wet structural element (Figure 12).

Given a fluid grid and a structural model, the Matcher programme described in Reference 27 generates all the data structures needed to evaluate the quantities in (43) and (44) in a single preprocessing step.

6.3. *Intrafield parallel processing*

Aeroelastic simulations are known to be computationally intensive and therefore can benefit from the parallel processing technology. An important feature of a partitioned solution procedure is preservation of software modularity. Hence all the solution procedures ALG0, ALG1, ALG2 and ALG3 can use existing computational fluid dynamics and computational structural mechanics parallel algorithms. The solution of the mesh motion equations (4) can be easily incorporated into an existing fluid code and its parallelization is no more difficult than that of a finite element structural algorithm.

Our approach to parallel processing is based on the mesh partitioning/message passing paradigm, which leads to a portable software design. Using an automatic mesh partitioning algorithm,[28,29] we decompose both fluid and structural meshes into subdomains. The same 'old' serial code is executed within every subdomain. The 'gluing' or assembly of the subdomain results is implemented in a separate software module. This approach enforces data locality[21] and therefore is suitable for all parallel hardware architectures. Note that in this context, message passing refers to the assembly phase of the subdomain results. It does not imply that messages have to be explicitly exchanged between the

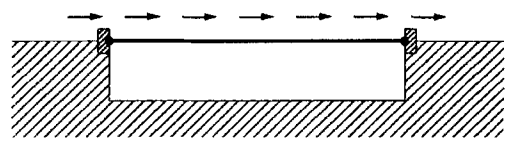

Figure 13. Flat panel with infinite aspect ratio

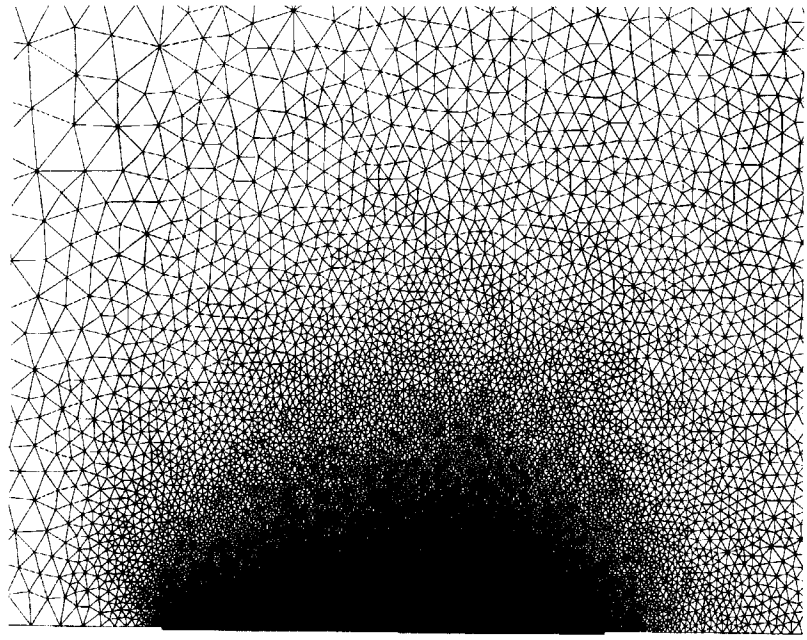

Figure 14. Discretization of flow domain (partial view)

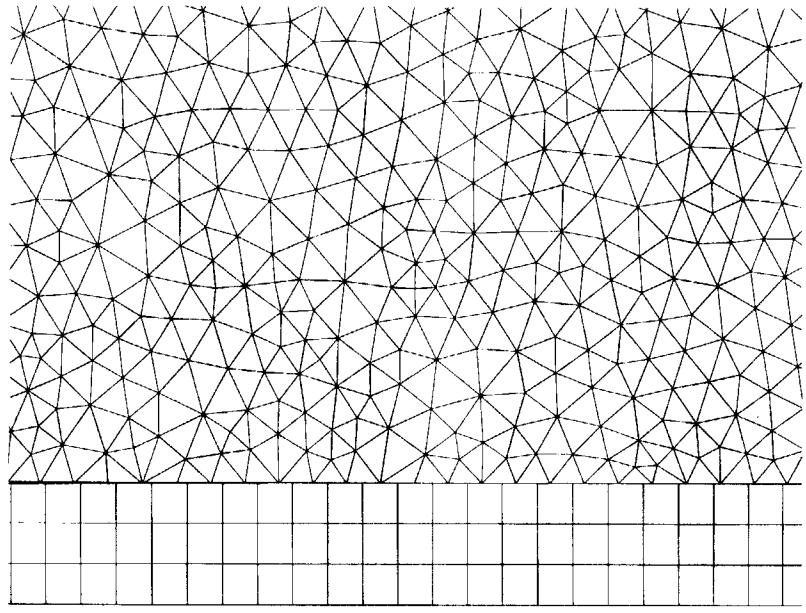

Figure 15. Mesh incompatibility

subdomains. For example, message passing can be implemented on a shared memory multiprocessor as a simple access to a shared buffer or as a duplication of one buffer into another.

6.4. *Interfield parallel processing*

Using the message passing paradigm, interfield parallel processing can be implemented in the same manner as intrafield multiprocessing. The fluid and structure codes can run either on different sequential or parallel machines or on a different partition of the same multiprocessor. Any software product such as PVM[30] can be used to implement message passing between the two computational kernels.

7. APPLICATION TO FLAT PANELS IN SUPERSONIC AIRSTREAMS

Here we demonstrate the aeroelastic computational methodology described in the previous sections with the numerical investigation on an iPSC-860 massively parallel processor of the instability of flat panels with infinite aspect ratio in supersonic airstreams.[31]

The flat panel with infinite aspect ratio (Figure 13) is assumed to have a length $L = 0{\cdot}5$ m, a uniform thickness $h = 1{\cdot}35 \times 10^{-3}$ m, a Young modulus $E = 7{\cdot}728 \times 10^{10}$ N/m^{-2}, a Poisson ratio $\mu = 0{\cdot}33$, a density $\rho = 2710$ kg m^{-3} and to be clamped at both ends. Its rectangular cross-section is discretized into 1111×3 plane strain four-node elements. This fine discretization, which generates 3333 elements with perfect aspect ratios and 4448 nodes, is not needed for accuracy; we have designed this structural mesh only because we are also interested in assessing some computational and I/O performance issues.

The two-dimensional flow domain above the panel is discretized into 32,568 triangles and 16,512 vertices (Figure 14). A slip condition is imposed at the fluid/structure boundary.

Because the fluid and structural meshes are not compatible at their interface (Figure 15), the Matcher software[27] is used to generate in a single preprocessing step the data structures required for transferring the pressure load to the structure and the structural deformations at the upper surface of the panel to the fluid.

We consider several supersonic flows at different Mach numbers and discuss the performances of ALG0, ALG1, ALG2, and ALG3. Whenever subcycling is used, the I^1 interpolation scheme is used to prescribe the motion of the fluid grid points on $\Gamma_{F/S}$.

7.1. *The fluid solver*

The Euler flow equations (2) are solved with a second-order-accurate finite volume monotonic upwinding scheme for conservation laws (MUSCL)[32] on fully unstructured grids. The resulting semidiscrete equations are time integrated with a second-order low-storage explicit Runge–Kutta method. The details of this explicit unstructured flow solver can be found in References 19 and 20.

7.2. *The mesh motion solver*

In this work the unstructured dynamic fluid mesh is represented by the pseudostructural model of Batina ($\tilde{M} = \tilde{D} = 0$). A fictitious linear spring is assigned to each edge connecting two fluid grid points S_i and S_j and is attributed the stiffness

$$k_{ij} = \frac{1}{\sqrt{[(\xi_{j_x} - \xi_{i_x})^2 + (\xi_{j_y} - \xi_{i_y})^2]}}. \tag{45}$$

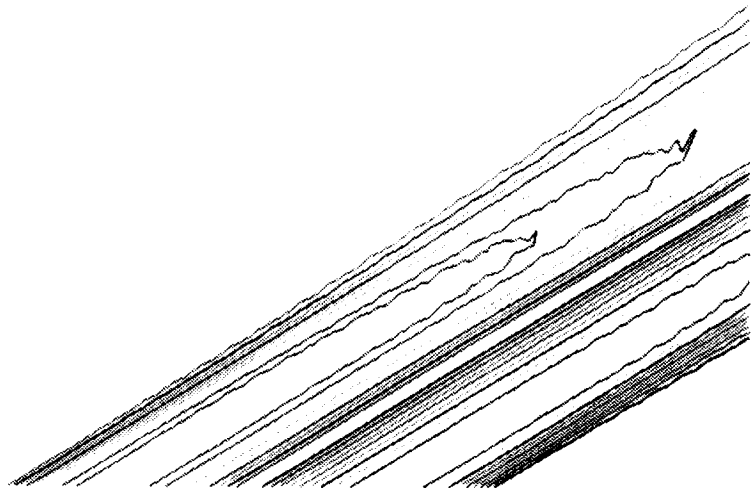

Figure 16. Pressure isovalues for steady state flow solution

The grid points located on the upstream and downstream boundaries are held fixed. The motion of those points located on $\Gamma_{F/S}$ is determined from the panel surface motion and/or deformation. At each time step t^{n+1} the new position of the interior grid points is determined from the solution of the displacement-driven pseudostructural problem (4) via a two-step iterative procedure. First the displacements of the interior grid points are predicted by extrapolating the previous displacements at time steps t^n and t^{n-1} in the following manner:

$$\Delta\xi = 2\Delta^n\xi - \Delta^{n-1}\xi, \tag{46}$$

where $\Delta^n\xi = \xi^{n+1} - \xi^n$. Next the predicted values are corrected with a few explicit Jacobi relaxations on the static equilibrium equations:

$$\Delta^{n+1}\xi_i = \frac{\sum_{j \in K(i)} k_{ij}\Delta\xi_j}{\sum_{j \in K(i)} k_{ij}}. \tag{47}$$

Finally the position of the fluid grid points at t^{n+1} is computed as follows:

$$\xi^{n+1} = \xi^n + \Delta^{n+1}\xi. \tag{48}$$

Figure 17. Initial perturbation of panel displacement field

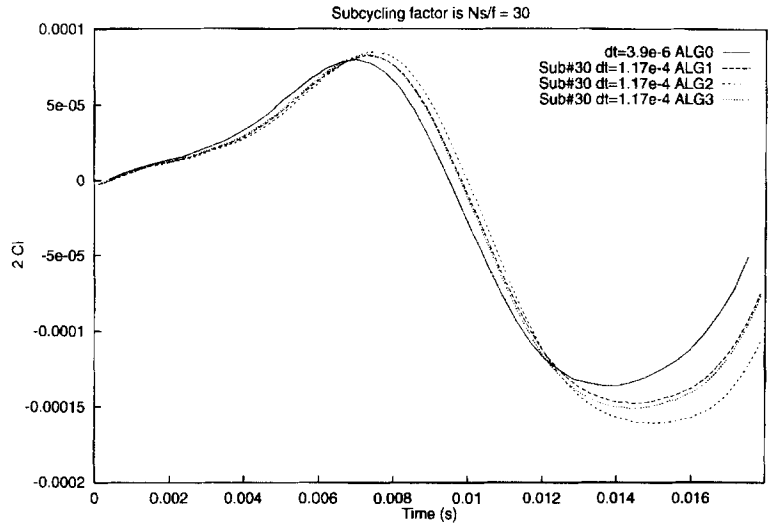

Figure 18. Lift coefficient history for $n_{S/F} = 30$

7.3. *The structural solver*

The structural equations of dynamic equilibrium (8) are solved with the parallel implicit transient finite element tearing and interconnecting (FETI) method.[33] Because it is based on a midpoint rule formulation, this method allows the enforcement of both continuity equations (6) while still respecting the GCL. The resistance of the structure to displacements in the plane of the skin is assumed to be small. Consequently, all panel flutter computations can be performed with a linearized structural theory. Since the FETI solver is a domain-decomposition-based iterative solver, we also use the special restarting procedure proposed in Reference 34 for the efficient iterative solution of linear systems with repeated right-hand sides.

7.4. *The computational platform*

All computations are performed on an iPSC-860 parallel processor using double-precision arithmetic. The fluid and structure solvers are implemented as separate programmes that communicate via the intercube communication procedures described in Reference 35.

Table I. Performance results on iPSC-860

Algorithm	Elapsed time for 4102 fluid time steps (s)			
	Fluid (64 processors)	Structure (two processors)	Fluid − wait + ICC	Total CPU
ALG0	2617·23	1267·93	1283·10	3900·33
ALG1 ($n_{S/F} = 10$)	2625·11	126·67	127·90	2753·01
ALG2 ($n_{S/F} = 5$)	2643·57	253·34	1·67	2645·24
ALG3 ($n_{S/F} = 10$)	2603·56	253·23	1·37	2604·93

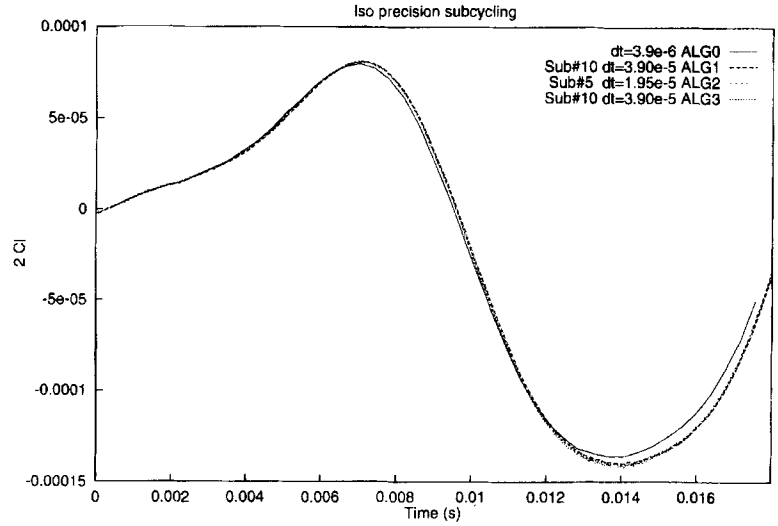

Figure 19. Lift coefficient history for a fixed level of accuracy

7.5. *Assessment of the partitioned procedures*

In order to illustrate the relative merits of the partitioned procedures ALG0, ALG1, ALG2 and ALG3, we first consider two different series of transient aeroelastic simulations at Mach number $M_\infty = 1{\cdot}90$ that highlight

(a) the relative accuracy of these algorithms for a fixed subcycling factor $n_{S/F}$
(b) the relative speed of these algorithms for a fixed level of accuracy on both sequential and parallel computational platforms.

In all cases 64 processors are allocated to the fluid system and two processors are assigned to the structural solver. Initially a steady state flow is computed above the panel at $M_\infty = 1{\cdot}90$ (Figure 16), a speed at which the panel described above is not supposed to flutter. Then the aeroelastic response of the coupled system is triggered by a displacement perturbation of the panel along its first mode (Figure 17).

First the subcycling factor is fixed at $n_{S/F} = 30$ and the lift coefficient is computed using the time step $\Delta t = 3{\cdot}9 \times 10^{-6}$ corresponding to the stability limit of the explicit flow solver in the absence of coupling with the structure. The obtained results are depicted in Figure 18 for the first 4102 time steps. For $n_{S/F} = 30$, ALG1 and ALG3 exhibit essentially the same accuracy. In the long run their amplitude and phase errors are less important than those of ALG2. Clearly this highlights the superiority of ALG3, which, despites it interfield parallelism and unlike ALG2, is capable of delivering the same accuracy as the sequential algorithm ALG1.

Next the relative speed of the focus partitioned solution procedures is assessed by comparing their CPU performance for a certain level of accuracy dictated by ALG0. It turns out that in order to meet the accuracy requirements of ALG0, both ALG1 and ALG3 can use a subcycling factor as large as $n_{S/F} = 10$ but ALG2 can subcycle only up to $n_{S/F} = 5$ (Figure 19).

The performance results measured on the iPSC-860 are reported in Table I, where ICC denotes the intercube communication time. Note that ICC is measured in the fluid kernel and includes idle time when the flow and structural communications do not overlap.

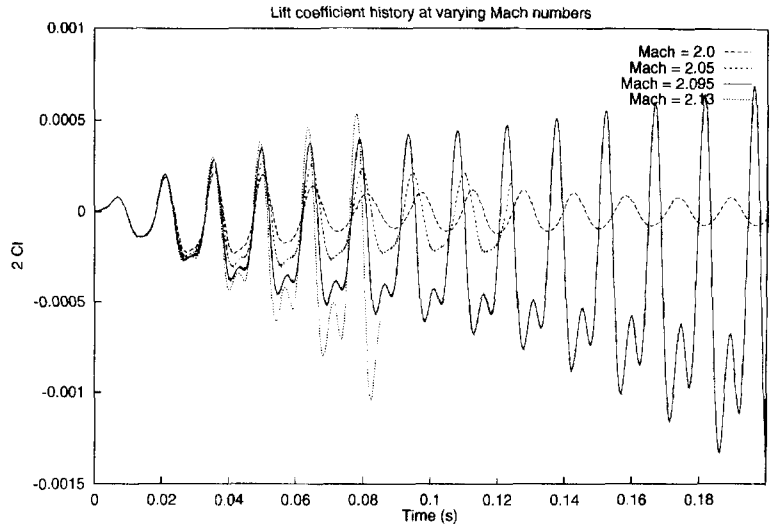

Figure 20. Flutter analysis

From the results reported in Table I, the following observations can be made.

1. The fluid computations dominate the simulation time. This is partly because the structural model is simple in this case and a linear elastic behaviour is assumed for the panel.
2. Considering that the iPSC-860 has 128 processors and that only clusters of 2^n processors can be defined on this machine, allocating 64 processors to the fluid and two processors to the structure achieves the minimum possible interfield load imbalance for this coupled problem.
3. The effect of subcycling on intercube communication costs is clearly demonstrated. Because the flow solution time is dominating, the effect of subcycling on the total CPU time is less important for ALG2 and ALG3 which feature interfield parallelism in addition to intrafield multiprocessing than for ALG1 which features intrafield parallelism only (note that ALG1 with $n_{S/F} = 1$ is identical with ALG0).
4. ALG2 and ALG3 allow a perfect overlap of interfield communications, which reduces intercube communication and idle time to less than 0·3 per cent of the amount corresponding to ALG0.
5. The superiority of ALG3 over ALG2 is not clearly demonstrated for this problem because of the simplicity of the structural model and the subsequent load imbalance between the fluid and structure computations.

7.6. *Panel flutter*

The classical and analytical solution of the instability problem of flat panels with infinite aspect ratio in supersonic airstreams assumes a shallow shell theory for the structure and a linearized formulation for the flow problem (piston theory). Within this analytical approach the dynamics of the focus coupled fluid/structure system are governed by a fourth-order partial differential equation (Reference 31, p. 419) and the flutter condition is obtained by analysing the roots of the corresponding characteristic equation. For the panel described at the beginning of Section 7, the classical linear theory predicts flutter at the critical Mach number $M_\infty^{cr} = 1{\cdot}98$. The objective of this subsection is to validate the aeroelastic simulation capability presented in this paper by reproducing the theoretical critical Mach number for the given panel. Note that in order to compare the analytical and finite element approaches,

the coefficients of the shallow shell equation described in Reference 31, p. 419, must be computed to represent the same equation as that corresponding to the finite element model used in this paper.

Four different runs at $M_\infty = 2{\cdot}0$, $2{\cdot}05$, $2{\cdot}095$ and $2{\cdot}13$ are performed using ALG3. For each run a steady state flow is first computed. Then a displacement perturbation of the panel along its first mode (Figure 17) is imposed and the aeroelastic response of the coupled system is computed. The predicted time histories of the lift coefficient are depicted in Figure 20 for all four cases.

From the results reported in Figure 20, it follows that the flutter speed predicted by our formulation verifies $2{\cdot}05 \leqslant M_\infty^{cr} \leqslant 2{\cdot}095$. Hence this flutter speed is 4·5 per cent higher than that predicted by the piston theory. This is a rather good agreement given that the piston theory and the computational approach presented herein do not share exactly the same approximations.

8. CLOSURE

In this paper we have presented several explicit (fluid)/implicit (structure) partitioned procedures for time integrating transient coupled aeroelastic problems, including one parallel coupling strategy with superior accuracy properties. We have discussed their merits in terms of accuracy, stability, heterogeneous computing, I/O transfers, subcycling and parallel processing. We have also described a general and flexible framework for implementing these solution procedures on heterogeneous and/or parallel computational platforms. This framework and the explicit/implicit algorithms have been demonstrated with the numerical investigation on an iPSC-860 massively parallel processor of the instability of flat panels with infinite aspect ratio in supersonic airstreams.

ACKNOWLEDGEMENTS

The authors acknowledge partial support by NASA Lewis Research Center under Grant NAG3-1273, partial support by the National Science Foundation under Grant ASC-9217394 and partial support by Dassault Aviation, Saint Cloud, France. They would like to thank Po-Shu Chen and Stephane Lanteri for their help in parallel implementations on the iPSC-860.

REFERENCES

1. J. Donea, 'An arbitrary Lagrangian–Eulerian finite element method for transient fluid–structure interactions', *Comput. Methods Appl. Mech. Eng.*, **33**, 689–723 (1989).
2. T. J. R. Hughes, W. K. Liu and T. K. Zimmermann, 'Lagrangian–Eulerian finite element formulation for incompressible viscous flows', *Proc. U.S.–Japan Seminar on Interdisciplinary Finite Element Analysis*, Ithaca, NY, August 1978.
3. T. Belytschko and J. M. Kennedy, 'Computer models for subassembly simulation', *Nucl. Eng. Design*, **49**, 17–38 (1978).
4. O. A. Kandil and H. A. Chuang, 'Unsteady vortex-dominated flows around manoeuvering wings over a wide range of mach numbers', *AIAA Paper 88-0317*, 1988.
5. C. Farhat and T. Y. Lin, 'Transient aeroelastic computations using multiple moving frames of reference', *AIAA Paper 90-3053*, 1990.
6. J. T. Batina, 'Unsteady Euler airfoil solutions using unstructured dynamic meshes', *AIAA Paper 89–0115*, 1989.
7. T. Tezduyar, M. Behr and J. Liou, 'A new strategy for finite element computations involving moving boundaries and interfaces—the deforming spatial domain/space–time procedure: I. The concept and the preliminary numerical tests, *Comput. Methods Appl. Mech. Eng.*, **94**, 339–351 (1992).
8. M. Lesoinne and C. Farhat, 'Stability analysis of dynamic meshes for transient aeroelastic computations', *AIAA Paper 93-3325*, 1993.
9. K. C. Park and C. A. Felippa, 'Partitioned analysis of coupled systems', in T. Belytschko and T. J. R. Hughes (eds), *Computational Methods for Transient Analysis*, North-Holland, Amsterdam, 1983, pp. 157–219.
10. T. Belytschko, P. Smolenski and W. K. Liu, 'Stability of multi-time step partitioned integrators for first-order finite element systems', *Comput. Methods Appl. Mech. Eng.*, **49**, 281–297 (1985).
11. C. Farhat, K. C. Park and Y. D. Pelerin, 'An unconditionally stable staggered algorithm for transient finite element analysis of coupled thermoelastic problems', *Comput. Methods Appl. Mech. Eng.*, **85**, 349–365 (1991).
12. S. Piperno, C. Farhat and B. Larrouturou, 'Partitioned procedures for the transient solution of coupled aeroelastic problems', *Comput. Methods Appl. Mech. Eng.*, **124**, 79–112 (1995).

13. C. J. Borland and D. P. Rizzetta, 'Nonlinear transonic flutter analysis', *AIAA J.*, **20**, 1606–1615 (1982).
14. V. Shankar and H. Ide, 'Aeroelastic computations of flexible configurations', *Comput. Struct.*, **30**, 15–28 (1988).
15. R. D. Rausch, J. T. Batina and T. Y. Yang, 'Euler flutter analysis of airfoils using unstructured dynamic meshes', *AIAA Paper 89-13834*, 1989.
16. M. Blair, M. H. Williams and T. A. Weisshaar, 'Time domain simulations of a flexible wing in subsonic compressible flow', *AIAA Paper 90-1153*, 1990.
17. T. W. Strganac and D. T. Mook, 'Numerical model of unsteady subsonic aeroelastic behaviour', *AIAA J.*, **28**, 903–909 (1990).
18. E. Pramono and S. K. Weeratunga, 'Aeroelastic computations for wings through direct coupling on distributed-memory MIMD parallel computers', *AIAA Paper 94-0095*, 1994.
19. C. Farhat, S. Lanteri and L. Fezoui, 'Mixed finite volume/finite element massively parallel computations: Euler flows, unstructured grids, and upwind approximations', in P. Mehrotra, J. Saltz and R. Voigt (eds), *Unstructured Scientific Computation on scalable Multiprocessors*, MIT Press, Boston, MA, 1992, pp. 253–283.
20. C. Farhat, L. Fezoui and S. Lanteri, 'Two-dimensional viscous flow computations on the Connection Machine: unstructured meshes, upwind schemes, and massively parallel computations', *Comput. Methods Appl. Mech. Eng.*, **102**, 61–88 (1993).
21. S. Lanteri and C. Farhat, 'Viscous flow computations on MPP systems: implementational issues and performance results for unstructured grids', in R. F. Sincovec *et al.* (eds), *Parallel Processing for Scientific Computing*, SIAM, Philadelphia, PA, 1993, pp. 65–70.
22. C. Farhat and S. Lanteri, 'Simulation of compressible viscous flows on a variety of MPPs: computational algorithms for unstructured dynamic meshes and performance results', *Comput. Methods Appl. Mech. Eng.*, **119**, 35–60 (1994).
23. P. D. Thomas and C. K. Lombard, 'Geometric conservation law and its application to flow computations on moving grids', *AIAA J.*, **17**, 1030–1037 (1979).
24. R. Beam and R. F. Warming, 'An implicit scheme for the compressible Navier–Stokes equations', *AIAA J.*, **16**, 393–402 (1978).
25. J. L. Steger and R. F. Warming, 'Flux vector splitting of the inviscid gasdynamic equations with applications to finite difference methods', *J. Comput. Phys.*, **40**, 263–293 (1981).
26. M. Lesoinne and C. Farhat, 'A geometric conservation law for flow problems with moving boundaries and unstructured deformable meshes, and its implications, *AIAA Paper 95–1709*, 1995.
27. N. Maman and C. Farhat, 'Matching fluid and structure meshes for aeroelastic computations: a parallel approach', *Comput. Struct.*, **54**, 779–785 (1995).
28. C. Farhat and M. Lesoinne, 'Automatic partitioning of unstructured meshes for the parallel solution of problems in computational mechanics', *Int. j. numer. methods eng.*, **36**, 745–764 (1993).
29. C. Farhat, S. Lanteri and H. D. Simon, 'TOP/DOMDEC, a software tool for mesh partitioning and parallel processing', *J. Comput. Syst. Eng.*, **6**, 13–26 (1995).
30. A. Geist, A. Beguelin, J. Dongarra, R. Mancheck and V. Sunderam, 'PVM 3.0 user's guide and reference manual', *Tech. Rep. ORNL/TM-12187*, Oak Ridge National Laboratory, 1993.
31. R. L. Bisplinghoff and H. Ashley, *Principles of Aeroelasticity*, Dover, New York, 1962.
32. B. Van Leer, 'Towards the ultimate conservative difference scheme. V: A second-order sequel to Godunov's method', *J. Comput. Phys.*, **32**, (1979).
33. L. Crivelli and C. Farhat, 'Implicit transient finite element structural computations on MIMD systems: FETI vs. direct solvers', *AIAA Paper 93-1310*, 1993.
34. C. Farhat, L. Crivelli and F. X. Roux, 'Extending substructure based iterative solvers to multiple load and repeated analyses', *Comput. Methods Appl. Mech. Eng.*, **117**, 195–209 (1994).
35. E. Barszcz, 'Intercube communication on the iPSC/860', *Proc. Scalable High Performance Computing Conf.*, Williamsburg, VA, April 1992.