NASA Contractor Report 195355

High-Performance Parallel Analysis of Coupled Problems for Aircraft Propulsion

C.A. Felippa, C. Farhat, S. Lanteri,
N. Maman, S. Piperno,
and U. Gumaste
University of Colorado
Boulder, Colorado

July 1994

Prepared for
Lewis Research Center
Under Grant NAG3–1425

National Aeronautics and
Space Administration

Summary

This research program deals with the application of high-performance computing methods for the analysis of complete jet engines. We have initited this program by applying the two-dimensional parallel aeroelastic codes to the interior gas flow problem of a by-pass jet engine. The fluid mesh generation, domain decomposition and solution capabilities were successfully tested. We have then focused attention on methodology for the partitioned analysis of the interaction of the gas flow with a flexible structure and with the fluid mesh motion that results from these structural displacements. This is treated by a new ALE technique that models the fluid mesh motion as that of a fictitious mass-spring network. New partitioned analysis procedures to treat this coupled 3-component problem are developed. These procedures involved delayed corrections and subcycling. Preliminary results on the stability, accuracy and MPP computational efficiency are reported.

1. OVERVIEW

The present program deals with the application of high-performance parallel computation for the analysis of complete jet engines, considering the interaction of fluid, thermal and mechanical components. The research is driven by the simulation of advanced aircraft propulsion systems, which is a problem of primary interest to NASA Lewis.

The coupled problem involves interaction of structures with gas dynamics, heat conduction and heat transfer in aircraft engines. The *methodology* issues to be addressed include: consistent discrete formulation of coupled problems with emphasis on coupling phenomena; effect of partitioning strategies, augmentation and temporal solution procedures; sensitivity of response to problem parameters; and methods for interfacing multiscale discretizations. The *computer implementation* issues to be addressed include: parallel treatment of coupled systems; domain decomposition and mesh partitioning strategies; data representation in object-oriented form and mapping to hardware driven representation, and tradeoff studies between partitioning schemes with differing degree of coupling.

2. STAFF

Two graduate students have been partly supported by this grant. M. Ronaghi (U.S. citizen) began his graduate studies at Colorado on January 1993. He has completed a M.Sc. in Aerospace Engineering on May 1994 but plans to pursue his Ph. D. in Mechanical Engineering under a different research program that fits his background and experience better.

U. Gumaste (permanent U.S. resident) began his graduate studies at Colorado in the Fall semester, but worked in this project during June and July 1993 as an hourly research assistant. Mr. Gumaste has a B.Tech in Civil Engineering from the Indian Institute of Technology, Bombay, India. He completed his Ph. D. course requirement this semester with the transfer of graduate credit units from the University of Maryland. During this period he was partly supported as a Teaching Assistant. He plans to familiarize himself with our aeroelastic codes during May-June and will visit NASA Lewis for 6 weeks during July and August.

The methods development for this project has been greatly benefitted from the presence of three visiting Post-Docs. Dr. S. Lanteri conducted extensive experimentation on several computational algorithms for compressive viscous flow simulation on the iPSC-860, CM-5 and KSR-1 as reported in Appendix II. Dr. N. Maman has implemented "mesh matching" techniques that connect separately generated fluid and structural meshes. Dr. S. Piperno has developed and evaluated implicit and subcycled partitioned analysis procedures for the interaction of structure, fluid and fluid-mesh motion. A new approach to augmentation of the governing semi-discrete equations that improves stability while keeping communications overhead modest was investigated. Results from this study are presented in Appendix I.

3. DEVELOPMENT OF PARTITIONED ANALYSIS METHODS

The first parallel computations of a jet engine, presented in the first Progress Report, dealt with the fluid flow within a jet engine structure that was considered rigid and hence provides only guiding boundary conditions for the gas flow. When the structural flexibility is accounted for two complications occur:

1. The engine simulation algorithm must account for the structural flexibility though periodic transfer of interaction information, and
2. The fluid mesh must smoothly follow the relative structural motions through an ALE (Adaptive Lagrangian Eulerian) scheme. The particular ALE scheme selected for the present work makes use of Batina's proposed pseudo-mechanical model of springs and masses overlaid over the fluid mesh.

Research work during the period July 1993 through January 1994 was dominated by the treatment of two subjects: partitioned analysis of fluid-structure interaction (FSI) and accounting for fluid mesh motions. The partitioned analysis algorithm developed for the FSI problem is always implicit in the structure (because of its larger time scale of significant vibratory motions) and either explicit or implicit for the gas flow modeled by the Navier-Stokes equations. Subcycling, in which the integration stepsize for the fluid may be smaller than that used in the structure, was also studied.

General Requirements

The fundamental practical considerations in the development of these methods are: (1) numerical stability, (2) fidelity to physics, (3) accuracy, and (4) MPP efficiency. Numerical stability is fundamental in that an unstable method, no matter how efficient, is useless. There are additional considerations:

1. Stability degradation with respect to that achievable for the *uncoupled* fields should be minimized. For example, if the treatment is implicit-implicit (I-I) we would like to maintain unconditional stability. If the fluid is treated explicitly we would like to maintain the same CFL stability limit.
2. Masking of physical instability should be avoided. This is important in that flutter or divergence phenomena should not be concealed by numerical dissipation. For this reasons all time integration algorithms considered in this work must exclude the use of artificial damping.

Stability vs. Communication-Overhead Tradeoff

The degradation of numerical stability degradation is primarily influenced by the nature of information exchanged every time step among the coupled subsystems during the course of partitioned integration. A methodology called *augmentation* that systematically exploits this idea was developed by Park and Felippa in the late 1970s. The idea is to modify the governing equations of one subsystem with system information from connected subsystems. The idea proved highly successful for the sequential computers of the time. A fresh look must be taken to augmentation, however, in light of the communications overhead incurred in massively parallel processing. For the present application three possibilities were considered:

No augmentation. The 3 subsystems (fluid, structure and ALE mesh) exchange only minimal interaction state information such as pressures and surface-motion velocities, but no information on system characteristics such as mass or stiffness. The resulting algorithm has minimal MPP communication overhead but poor stability characteristics. In fact the stability of an I-I scheme becomes conditional and not too different from that of a less expensive I-E scheme. This degradation in turns significantly limits the stepsize for both fluid and structure.

Full augmentation. This involves transmission of inverse-matrix-type data from one system to another. Such data are typified by terms such as a a structure-to-fluid coupling-matrix times the inverse of the structural mass. Stability degradation can be reduced or entirely eliminated; for example I-I unconditional stability may be maintained. But because the transmitted matrix combinations tend to be much less sparse than the original system matrices, the MPP communications overhead can become overwhelming, thus negating the benefits of improved stability characteristics.

Partial augmentation. This new approach involves the transmission of coupling matrix information which does not involve inverses. It is efficiently implemented as a delayed correction to the integration algorithm by terms proportional to the squared stepsize. The MPP communication requirements are modest in comparison to the fully-augmented case, whereas stability degradation can be again eliminated with some additional care.

The partial augmentation scheme was jointly developed by S. Piperno and C. Farhat. Its derivation is fully reported in Appendix I, which has been submitted for publication. The general methodology is first applied to a staggered FSI algorithm (staggering is a special form of partition) with common timestep, and then extended to cover subcycling. The reduced communications overhead has been recently verified on simple problems with preliminary tests on a iPSC860 Hypercube. Tests of this new scheme for more complex geometries will be carried out this summer.

Algorithmic Effects of Dynamic Fluid Mesh

The first one-dimensional results on the effect of a dynamic fluid mesh on the stability and accuracy of the staggered integration were obtained by C. Farhat and S. Piperno and are also discussed in Appendix I. A doctoral student, M. Lesoinne, supported by NSF is extending these calculations to the multidimensional case.

Appendix II contains a report by Charbel Farhat and S. Lanteri that summarizes recent experiences with the application of the two-dimensional Navier Stokes equations combined with the pseudo-mechanical network approach to dynamic mesh motion. Performance results obtained on the iPSC-860, KSR-1 and CM-5 parallel computers are reported for both fixed and moving fluid meshes.

4. FUTURE WORK

The present work undertaken since the renewal of the grant on March 1994 is focused on axisymmetrizing the fluid-structure interaction codes, in which the new partially-augmented staggered schemes and mesh motion techniques are implemented. We still need to introduce more physical effects in the gas flow, namely compression, diffusion and combustion. The modeling experience that Mr. Gumaste will acquire during his visit to NASA Lewis should help with the prosecution of these items.

APPENDIX I

Partitioned Procedures for the Transient Solution of Coupled Aeroelastic Problems

S. PIPERNO AND C. FARHAT
Department of Aerospace Engineering Sciences
and Center for Aerospace Structures
University of Colorado at Boulder
Boulder, CO 80309-0429

January 1994

Abstract

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. In this paper, we present several partitioned procedures for time-integrating this focus coupled problem and discuss their merits in terms of accuracy, stability, heterogeneous computing, I/O transfers, subcycling, and parallel processing. All theoretical results are derived for a one-dimensional piston model problem with a compressible flow, because the complete three-dimensional aeroelastic problem is difficult to analyze mathematically. However, the insight gained from the analysis of the coupled piston problem and the conclusions drawn from its numerical investigation are confirmed with the numerical simulation of the two-dimensional transient aeroelastic response of a flexible panel in a transonic nonlinear Euler flow regime.

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 distorsions when the structure undergoes large motion. Several different approaches have emerged as an alternative to partial re-gridding in transient aeroelastic computations, among which we note the arbitrary Lagrangian-Eulerian (ALE) formulation [1–3], the co-rotational approach [4,5], and dynamic meshes [6] (see also [28] for a review). All of these approaches treat a computational aeroelastic problem as a coupled two-field problem.

However, the moving mesh itself can be formulated as a pseudo-structural system with its own dynamics [7], 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 semi-discrete equations governing this three-way coupled problem can be written as follows:

$$\begin{aligned} \frac{\partial}{\partial t}(A(x,t)\ W(x,t)) + \tilde{F}_c^c(W(x,t),x,\dot{x}) &= \tilde{F}^d(W(x,t)) \\ M\frac{\partial^2 q}{\partial t^2} + f^{int}(q) &= f^{ext}(W(x,t)) \\ \tilde{M}\frac{\partial^2 x}{\partial t^2} + \tilde{D}\frac{\partial x}{\partial t} + \tilde{K}x &= K_c\ q \end{aligned} \tag{1}$$

where x 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, $\tilde{F}_c^c$ is the convected vector of convective fluxes [7], $\tilde{F}^d$ 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} 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 the structure and the moving fluid grid [7], 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 $\tilde{K} = I$ corresponds to a rigid mesh motion of the fluid grid around an oscillating airfoil, and $\tilde{M} = \tilde{D} = 0$ corresponds to the spring-based mesh motion scheme introduced in [6].

Each of the three components of the coupled problem described by Eqs. (1) has different mathematical and numerical properties, and distinct software implementation requirements. For Euler and Navier-Stokes flows, the fluid equations are nonlinear. The structural equations may be linear or nonlinear. The semi-discrete equations governing the pseudo-structural fluid grid system are linear. The matrices resulting from a linearization procedure are in general symmetric

for the structural problem, but they are typically unsymmetric for the fluid problem. Morevoer, the nature of the coupling in Eqs. (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, the pressure variable cannot be easily isolated neither from the fluid equations nor from the fluid state vector W. Consequently, the numerical solution of Eqs. (1) via a fully coupled monolithic scheme is computationally challenging and software-wise unmanageable.

Alternatively, Eqs. (1) can be solved via partitioned procedures [8]. 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, and c) advance the fluid system and compute a new pressure load [9–12]. 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 [13], especially when the fluid equations are nonlinear and treated implicitly [14].

The objective of this paper is the investigation of a broader range of partitioned procedures for the transient solution of coupled aeroelastic problems, with particular attention to accuracy and stability issues, subcycling schemes, accuracy v.s. speed trade-offs, implementation on heterogeneous computing platforms, and inter-field as well as intra-field parallel processing. The complete three-dimensional aeroelastic problem is difficult to analyze because it mixes linear and nonlinear operators, symmetric and unsymmetric matrices, explicit and implicit coupling, and can become physically unstable. Therefore, we begin our investigation with the design and analysis of partitioned integrators for a simplified one-dimensional aeroelastic problem that turned out to be a good model problem for the more complex aeroelastic systems that we wish to gain some intuition about. We focus on implicit time-integration schemes for the structural field, because the aeroelastic response of a structure is often dominated by low frequency dynamics. However, we consider both implicit/implicit and explicit/implicit fluid/structure partitioned procedures, with and without nontrivial prediction schemes. We discuss the computational and implementation aspects of each procedure and contrast their respective merits and shortcomings. Finally, we validate all the conclusions drawn from the investigation of the model problem with the simulation of the two-dimensional transient aeroelastic response

of a flexible panel in a transonic nonlinear Euler flow.

2. A one-dimensional aeroelastic model problem with an Euler flow

2.1. The piston problem: ALE formulation and linearization

As a model problem, we consider the one-dimensional piston depicted in Fig. 1. The equilibrium state of this coupled system is defined by a uniform pressure p_0 inside and outside the piston chamber, a uniform gas density ρ_0, a zero flow velocity $u_0 = 0$, and a chamber length equal to l_0. The gas is assumed to be perfect, and the flow isentropic. Hence, the pressure p is function of the density ρ only and obeys:

$$\frac{dp}{d\rho} = c^2 \tag{2}$$

where c denotes the sound speed. The cross sectional area of the chamber is assumed to be constant and equal to one.

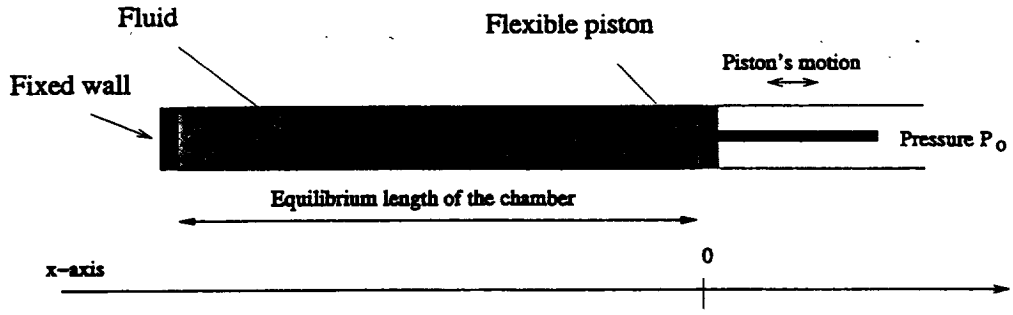

Fig. 1. The one-dimensional piston problem

For this model aeroelastic problem, the one dimensional mass and momentum conservation equations for the fluid are:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) &= 0 \\ \frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}((\rho u^2 + p)) &= 0 \end{aligned} \tag{3}$$

The linear dynamic equilibrium of the piston is governed by:

$$m\ddot{q} + d\dot{q} + kq = p(l_0 + q) - p_0 \tag{4}$$

where m, d, k, and q denote respectively the piston mass, damping, stiffness and displacement. A dot superscript designates a derivative with respect to time.

The boundary conditions for this coupled fluid/structure problem are given by:

$$\begin{aligned} \rho u(0) &= 0 \\ u(l_0 + q) &= \dot{q} \end{aligned} \tag{5}$$

Eqs. (5) above state that the fluid velocity is zero at the fixed wall, and equal to the piston speed at the other end of the chamber.

Clearly, the fluid flow has one moving boundary. Therefore, it is convenient to re-write Eqs. (3) with respect to a moving frame characterized by a velocity $\dot{\xi}$ that may be different from the fluid velocity u and from zero. Let $\mathcal{J} = det(dx/d\xi)$ denote the jacobian of the frame transformation $x \rightarrow \xi$. The ALE form of Eqs. (3) goes as follows:

$$\begin{aligned} \frac{1}{\mathcal{J}}\frac{\partial}{\partial t}(\mathcal{J}\rho) + \frac{\partial}{\partial x}(\rho(u - \dot{\xi})) &= 0 \\ \frac{1}{\mathcal{J}}\frac{\partial}{\partial t}(\mathcal{J}\rho u) + \frac{\partial}{\partial x}((\rho u(u - \dot{\xi}) + p)) &= 0 \end{aligned} \tag{6}$$

The above equations can be re-written in vector form as:

$$\frac{1}{\mathcal{J}}\frac{\partial}{\partial t}(\mathcal{J}W) + \frac{\partial}{\partial x}(F^c(W)) = 0 \tag{7}$$

where W and F^c are respectively the fluid state and fluid convected flux vectors:

$$\begin{aligned} W &= \begin{pmatrix} \rho \\ \rho u \end{pmatrix} \\ F^c &= \begin{pmatrix} \rho(u - \dot{\xi}) \\ (\rho u(u - \dot{\xi}) + p) \end{pmatrix} \end{aligned} \tag{8}$$

The convection matrix associated with the above convected flux vector is:

$$J(\dot{\xi}) = \frac{\partial F^c}{\partial W} = \begin{pmatrix} -\dot{\xi} & 1 \\ -u^2 + c^2 & 2u - \dot{\xi} \end{pmatrix} \tag{9}$$

We consider the response of the aeroelastic coupled system to small perturbations around the equilibrium position (ρ_0, $u_0 = 0$, p_0, c_0). First, we note that the fluid state vector at equilibrium $W_0 = \begin{pmatrix} \rho_0 \\ 0 \end{pmatrix}$ satisfies:

$$\frac{1}{\mathcal{J}}\frac{\partial}{\partial t}(\mathcal{J}W_0) + \frac{\partial}{\partial x}(F^c(W_0)) = 0 \tag{10}$$

and the convection matrix at equilibrium is:

$$J_0(0) = \begin{pmatrix} 0 & 1 \\ c_0^2 & 0 \end{pmatrix} \tag{11}$$

Then, we linearize the convected flux vector around W_0:

$$F^c(W) = F^c(W_0) + J_0(0)\delta W - \delta\dot{\xi} W_0 \tag{12}$$

Finally, from Eqs. (7-12) it follows that the linearized fluid flow equations are:

$$\frac{1}{\mathcal{J}}\frac{\partial}{\partial t}(\mathcal{J}\delta W) + \frac{\partial}{\partial x}(J_0(0)\delta W - \delta\dot{\xi} W_0) = 0 \tag{13}$$

2.2. Spatial discretization: finite volume formulation and upwinding

The one-dimensional chamber region is discretized into N grid points and N cells (Fig. 2). Integrating Eqs. (13) between $x_{j-\frac{1}{2}}$ and $x_{j+\frac{1}{2}}$ and using a finite volume formulation with upwinding leads to:

$$\begin{aligned} \Delta x_j\, \delta\dot{W} + (\widetilde{F}^c_{j+\frac{1}{2}} - \widetilde{F}^c_{j-\frac{1}{2}}) &= 0 \\ \text{where} & \\ \Delta x_j &= \frac{x_{j+1} - x_{j-1}}{2} \end{aligned} \tag{14}$$

Here, $\widetilde{F}^c_{j+\frac{1}{2}}$ and $\widetilde{F}^c_{j-\frac{1}{2}}$ are the convected numerical fluxes [7] associated with the classical $+/-$ flux splitting:

$$\begin{aligned} \widetilde{F}^c_{j+\frac{1}{2}} &= J_0^+(0)\delta W_j + J_0^-(0)\delta W_{j+1} - \delta\dot{\xi}_{j+\frac{1}{2}} \frac{(W_{0_j} + W_{0_{j+1}})}{2} \\ \widetilde{F}^c_{j-\frac{1}{2}} &= J_0^+(0)\delta W_{j-1} + J_0^-(0)\delta W_j - \delta\dot{\xi}_{j-\frac{1}{2}} \frac{(W_{0_{j-1}} + W_{0_j})}{2} \end{aligned} \tag{15}$$

and $J_0^+(0)$ and $J_0^-(0)$ satisfy $J_0(0) = J_0^+(0) + J_0^-(0)$ and are given by:

$$\begin{aligned} J_0^+(0) &= \frac{1}{2}\begin{pmatrix} c_0 & 1 \\ c_0^2 & c_0 \end{pmatrix} \\ J_0^-(0) &= \frac{1}{2}\begin{pmatrix} -c_0 & 1 \\ c_0^2 & -c_0 \end{pmatrix} \end{aligned} \tag{16}$$

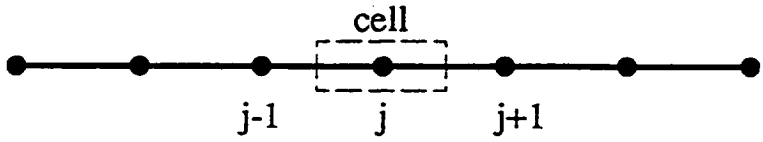

Fig. 2. Discretization of the one-dimensional flow

Substituting Eqs. (15) into Eqs. (14) gives:

$$\begin{aligned} \Delta x_j\ \delta\dot{W} - J_0^+(0)\delta W_{j-1} + (J_0^+(0) - J_0^-(0))\delta W_j + J_0^-(0)\delta W_{j+1} \\ - \delta\dot{\xi}_{j+\frac{1}{2}}\frac{(W_{0_j} + W_{0_{j+1}})}{2} - \delta\dot{\xi}_{j-\frac{1}{2}}\frac{(W_{0_{j-1}} + W_{0_j})}{2} \ = \ 0 \end{aligned} \tag{17}$$

2.3. Transpiration

From Fig.2 and the second of Eqs. (5), it follows that:

$$\delta\dot{\xi}_{N+\frac{1}{2}} \ = \ \dot{q} \tag{18}$$

All other ALE grid velocity perturbations $\delta\dot{\xi}_j$, $j = 1$, ..., N are arbitrary. In order to simplify the piston problem from a three- to a two-field coupled problem, we assume that these velocity perturbations are small compared to the unperturbed sound speed c_0. Consequently, Eqs. (17) become:

$$\begin{aligned} \Delta x_j\ \delta\dot{W} - J_0^+(0)\delta W_{j-1} + (J_0^+(0) - J_0^-(0))\delta W_j + J_0^-(0)\delta W_{j+1} \ &= \ 0 \quad (j \neq N) \\ \Delta x_N\ \delta\dot{W} - J_0^+(0)\delta W_{N-1} + (J_0^+(0) - J_0^-(0))\delta W_N - \dot{q}W_{0_N} \ &= \ 0 \end{aligned} \tag{19}$$

The quantity $\dot{q}W_{0_N} = \begin{pmatrix} \rho_0\dot{q} \\ 0 \end{pmatrix}$ corresponds to a "transpiration" flux. The reader can check that except for the presence of this transpiration flux, Eqs. (19) are identical to the semi-discrete linearized equations governing a one-dimensional fluid flow with fixed boundaries.

2.4. The semi-discrete aeroelastic model problem

We define the structure state vector as:

$$Q = \begin{pmatrix} q \\ \dot{q} \end{pmatrix} \tag{20}$$

Using Q, the structural Eqs. (4) can be re-written as:

$$\dot{Q} = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{d}{m} \end{pmatrix} Q + \begin{pmatrix} 0 \\ \frac{p(l_0+q)-p_0}{m} \end{pmatrix} \tag{21}$$

For the linearized piston problem, the forcing term $\begin{pmatrix} 0 \\ \frac{p(l_0+q)-p_0}{m} \end{pmatrix} = \frac{c_0^2}{m} \begin{pmatrix} 0 \\ \delta\rho_N \end{pmatrix}$ is a linear function of the fluid state vector $\delta W = \begin{pmatrix} \delta\rho \\ \delta(\rho u) \end{pmatrix}$. Hence, Eq. (21) can be re-written as:

$$\begin{aligned} \dot{Q} &= DQ + C\delta W \\ &\text{where} \\ D &= \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{d}{m} \end{pmatrix} \end{aligned} \tag{22}$$

Also, Eqs. (19) can be re-arranged in matrix form as follows:

$$\delta\dot{W} = A\delta W + BQ \tag{23}$$

where B is the matrix induced by the transpiration flux $\dot{q}W_{0_N}$.

In summary, the semi-discrete coupled system associated with the one-dimensional aeroelastic model problem is completely defined by:

$$\boxed{\begin{aligned} \begin{pmatrix} \delta\dot{W} \\ \dot{Q} \end{pmatrix} &= \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \delta W \\ Q \end{pmatrix} \\ \begin{pmatrix} \delta W \\ Q \end{pmatrix}_{(t=0)} &= \begin{pmatrix} \delta W \\ Q \end{pmatrix}_0 \end{aligned}} \tag{24}$$

In the remainder of this paper, we focus on developing, analyzing, and validating partitioned procedures for solving Eqs. (24). Because the aeroelastic response of a structure is often dominated by low frequency dynamics, we consider only implicit schemes for time-integrating the structural field. However,

we consider both explicit and implicit time-integrators for advancing the fluid field, as both approaches are popular in computational fluid dynamics. Elegant methods for analyzing the stability of partitioned integrators with and without subcycling can be found in [15,16]. However, both of these references deal with symmetric fields only. We have found that the extension of these analysis methods to mixed symmetric/unsymmetric problems such as those described by Eqs. (24) is difficult — if not impossible — which has also motivated us to investigate first a simplified aeroelastic model problem.

REMARK 1. Eqs. (24) are also valid for two-dimensional and three-dimensional linearized aeroelastic problems.

3. Mathematical preliminaries

3.1. Physical v.s. numerical instabilities

Transient fluid (gas)/structure interaction problems have one particularity: they possess a wide variety of self-excited vibrations and instabilities. For example, at speeds of flow somewhat above the *critical flutter speed* [17], the structural system extracts energy from the flow system and a small accidental disturbance of the airfoil can serve as a trigger to initiate an oscillation of great violence. Physical instabilities can also occur in the linear regime. An example of a linear dynamic instability is vibrations due to von Kármán vortices [18]. If the frequency of the structure loading caused by the vortices is close or equal to the natural frequency of the body, then a resonance effect is present and large amplitudes of vibrations result. Therefore, when it comes to analyzing the numerical stability of a proposed algorithm for time-integrating fluid/structure interaction problems, it is essential to consider the case where the coupled system is physically stable — that is, when Eqs. (1) or even Eqs. (24) have a solution that does not grow indefinitely in time.

The objectives of this section are to present a mathematical framework for the stability analysis of the solution of the semi-discrete Eqs. (24), and to show that for the aeroelastic model problem introduced and discretized in Section 2, these equations have always a stable solution. Hence, the fluid/structure interaction model problem presented in this paper is also a good problem for analyzing partitioned time-integrators with particular reference to numerical stability.

REMARK 2. Intuitively, one can expect Eqs. (24) to admit a stable solution for the aeroelastic model problem, because the fluid flow is confined inside a closed chamber and therefore has a limited amount of energy to exchange with the

piston, and the piston is not excited by any other external and time-dependent force. However, the analysis framework presented in Section 3.2 is interesting because it also reveals a numerical property of the coupled model problem that turns out to be important for the design of an unconditionally stable partitioned procedure for solving Eqs. (24).

3.2. Analysis framework

Let X, M, and $Sp\,(M)$ denote respectively a real vector, a real matrix that is diagonalizable in the complex space $\mathcal{C}$, and its set of complex eigenvalues. We focus our attention on the linear system of ordinary differential equations (ODE):

$$\dot{X} \;=\; MX \tag{25}$$

First, we introduce two definitions.

DEFINITION 1. We will say that M is "stable" if and only if:
a) M is diagonalizable in $\mathcal{C}$
b) $\forall \lambda \in Sp\,(M)$, $\Re\,(\lambda) \;\leq\; 0$

In b), $\Re\,(\lambda)$ designates the real part of λ.

DEFINITION 2. We will say that the real symmetric positive definite (RSPD) matrix E_M is an "energy matrix" for M if and only if $E_M M$ is non-positive, that is:

$$\forall X, \quad X^t E_M M X \;\leq\; 0$$

Here, the superscript t designates the transpose operation.

Next, we state and prove four theorems.

THEOREM 1. An RSPD matrix E_M is an energy matrix for M if and only if:

$$\forall X \text{ solution of } \dot{X} = MX, \quad \frac{d}{dt}(\frac{1}{2} X^t E_M X) \;\leq\; 0 \tag{26}$$

Proof. From Eq. (25), it follows that $\frac{d}{dt}(\frac{1}{2} X^t E_M X) = X^t E_M M X$. Hence, $\frac{d}{dt}(\frac{1}{2} X^t E_M X) \leq 0$ if and only if E_M is an energy matrix for M.

THEOREM 2. If $M = P^{-1}\Omega P$ denotes the diagonalization of a stable matrix M, then an energy matrix for M is given by:

$$E_M \;=\; \overline{P}^t P \;+\; P^t \overline{P} \tag{27}$$

where $\overline{P}$ is the complex conjugate matrix of P.

Proof. Clearly, $E_M = \overline{P}^t P + P^t \overline{P}$ is real symmetric. For all real vectors X, $X^t E_M X = 2|PX|^2$ is positive and equal to zero only if $X = 0$ because P is non singular. Finally since $\overline{M} = M$, we have:

$$\begin{aligned} X^t E_M M X &= X^t \bar{P}^t \Omega P X + X^t P^t \bar{P} P^{-1} \Omega P X \\ &= 2\Re\left(X^t \bar{P}^t \Omega P X\right) = 2\sum_i |(PX)_i|^2 \Re(\Omega_i) \leq 0 \end{aligned}$$

which completes the proof of THEOREM 2.

THEOREM 3 (reciprocal of THEOREM 2). Let M be a real matrix that is diagonalizable in C. If there exists an energy matrix E_M for M, then M is stable.

Proof. Let $X = R + iI \neq 0$, where $i^2 = -1$, denote a complex eigenvector of M associated with an eigenvalue λ. If E_M is an energy matrix for M, we have:

$$\begin{aligned} 0 \geq R^t E_M M R &= R^t E_M \Re(MX) = R^t E_M \Re(\lambda X) = R^t E_M [\Re(\lambda) R - \Im(\lambda) I] \\ 0 \geq I^t E_M M I &= I^t E_M \Im(MX) = I^t E_M \Im(\lambda X) = I^t E_M [\Re(\lambda) I + \Im(\lambda) R] \end{aligned} \tag{28}$$

where $\Im(\lambda)$ designates the imaginary part of λ. Adding the two inequalities in (28) and exploiting the symmetry of E_M leads to $\Re(\lambda)[R^t E_M R + I^t E_M I] \leq 0$. Since E_M is RSPD, it follows that $\Re(\lambda) \leq 0$, which completes the proof of THEOREM 3.

THEOREM 4. If A and D are two real stable matrices with energy matrices E_A and E_D, then:

$$E_A B + (E_D C)^t = 0 \Longrightarrow M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \text{ is a stable matrix}$$

Proof. The matrix $E_M = \begin{pmatrix} E_A & 0 \\ 0 & E_D \end{pmatrix}$ is RSPD. If $X = \begin{pmatrix} \delta W \\ Q \end{pmatrix}$ is a real vector satisfying $\dot{X} = MX$, we have:

$$\begin{aligned} \frac{d}{dt}\left(\frac{1}{2} X^t E_M X\right) &= \delta W^t E_A A \delta W + \delta W^t E_A B Q + Q^t E_D C \delta W + Q^t E_D D Q \\ &= \delta W^t E_A A \delta W + \delta W^t \left[E_A B + (E_D C)^t\right] Q + Q^t E_D D Q \\ &= \delta W^t E_A A \delta W + Q^t E_D D Q \leq 0 \end{aligned}$$

which in view of THEOREM 1 implies that E_M is an energy matrix for M. From THEOREM 3, it follows that M is stable.

Theorems 1-3 set the stage to THEOREM 4 which has a nice physical interpretation. An uncoupled fluid system is physically stable: it does not produce energy. If D is non-positive, an uncoupled structural system is also stable. For a coupled fluid/structure system, $E_A B + (E_D C)^t = 0$ simply expresses that the energy extracted from one system is equal to that injected into the other one. Hence, THEOREM 4 merely states that a coupled system where the energy is exactly conserved is a physically stable system.

3.3. Physical stability of the model problem

Consider again the aeroelastic model problem introduced in Section 2. Assuming a constant mesh size Δx, the linearized energy $\mathcal{E}_{fluid}$ of the discretized fluid system can be written as:

$$\mathcal{E}_{fluid} = \Delta x \sum_{j=1}^{j=N} \left(\frac{c_0^2 \delta\rho_j^2}{2\rho_0} + \frac{\rho_0 \delta u_j^2}{2}\right) \tag{29}$$

and its perturbed state vector is $\delta W = (\delta\rho_1, \delta(\rho_0 u_1), \ldots, \delta\rho_N, \delta(\rho_0 u_N))^t$. Therefore, E_A can be constructed for this system as follows:

$$E_A = \begin{pmatrix} \frac{\Delta x c_0^2}{\rho_0} & 0 & \cdots & 0 \\ 0 & \frac{\Delta x}{\rho_0} & & \\ \vdots & & \ddots & \vdots \\ & & \frac{\Delta x c_0^2}{\rho_0} & 0 \\ 0 & \cdots & 0 & \frac{\Delta x}{\rho_0} \end{pmatrix} \tag{30}$$

Using Eqs. (19), the reader can verify through tedious but elementary calculations that E_A is an energy matrix for A induced by the spatial flux splitting.

For the piston, the state vector is $Q = \begin{pmatrix} q \\ \dot{q} \end{pmatrix}$ and the energy is $\mathcal{E}_{piston} = (kq^2 + m\dot{q}^2)/2$. Hence, for this structural system E_D can be written as:

$$E_D = \begin{pmatrix} k & 0 \\ 0 & m \end{pmatrix} \tag{31}$$

Proving that E_D is an energy matrix for D is straightforward. Using the second of Eqs. (22) we have:

$$E_D D = \begin{pmatrix} k & 0 \\ 0 & m \end{pmatrix} \begin{pmatrix} 0 & 1 \\ \frac{-k}{m} & \frac{-d}{m} \end{pmatrix} = \begin{pmatrix} 0 & k \\ -k & -d \end{pmatrix}$$

which shows that $E_D D$ is negative when the damping d is positive, and therefore proves that E_D is an energy matrix for D.

The transpiration term in the last grid cell and the pressure force on the piston generate respectively the matrices:

$$B = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & \frac{-\rho_0}{\Delta x} & 0 \end{pmatrix}^t \text{ and } C = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & \frac{c_0^2}{m} & 0 \end{pmatrix} \tag{32}$$

From Eqs. (30-32) and after some algebraic manipulations, it follows that $E_A B + (E_D C)^t = 0$, which shows that the semi-discrete aeroelastic model problem introduced in Section 2 admits a stable solution. Therefore, staggered algorithms for time-integrating Eqs. (24) can be analyzed for unconditional stability.

4. A family of implicit/implicit partitioned procedures

4.1. Unconditionally stable staggered time-integrators

Here, we present a family of unconditionally stable implicit/implicit staggered algorithms for solving the model Eqs. (24) whose "design" is based on the following 4-step methodology.

Step II1. Predict the structural field using the value computed at $t_n = n\Delta t$:

$$Q^p = Q^n$$

Step II2. Advance the fluid system using the trapezoidal rule:

$$\overline{\delta W^{n+1}} = \delta W^n + \Delta t A \overline{\delta W^{n+\frac{1}{2}}} + \Delta t B Q^p$$

where

$$\delta W^{n+\frac{1}{2}} = \frac{\delta W^n + \delta W^{n+1}}{2}$$

Step II3. Advance the structural system using an implicit time-integrator selected from the so-called *generalized trapezoidal family of methods* [19], and a midpoint value of the previously updated fluid state $\delta W^{n+\frac{1}{2}}$:

$$\overline{Q^{n+1}} = Q^n + \Delta t D \overline{Q^{n+\alpha}} + \Delta t C \overline{\delta W^{n+\frac{1}{2}}}$$

where

$$Q^{n+\alpha} = (1-\alpha) Q^n + \alpha Q^{n+1} \qquad \alpha \in]0,\ 1]$$

Step II4. Correct the equations giving $\overline{\delta W^{n+1}}$ and $\overline{Q^{n+1}}$ to enforce unconditional stability of the implicit/implicit staggered procedure:

$$\begin{aligned} \delta W^{n+1} &= \overline{\delta W^{n+1}} + [\delta W^{n+1^c}] \\ Q^{n+1} &= \overline{Q^{n+1}} + [Q^{n+1^c}] \end{aligned}$$

It should be emphasized that the above steps describe the design process of a solution methodology and not the computer implementation of a time-integration algorithm. In particular, neither the fluid nor the structural fields should be solved until the correction terms $[\delta W^{n+1^c}]$ and $[Q^{n+1^c}]$ are first specified.

The trapezoidal rule is unconditionally stable and second-order accurate when applied to the solution of the uncoupled linearized fluid system (Eqs. (17)). For $\alpha \geq 1/2$, the generalized trapezoidal family of methods is unconditionally stable when applied to time-integrate the uncoupled structural problem. For $\alpha \in]0,\ 1]$, these methods are first-order accurate, except for $\alpha = 1/2$, in which case the corresponding scheme is second-order accurate. The correction terms $[\delta W^{n+1^c}]$ and $[Q^{n+1^c}]$ should be computed to ensure the unconditional stability of the resulting implicit/implicit staggered solution procedure.

THEOREM 5. For $\alpha \geq 1/2$, $[\delta W^{n+1^c}] = \frac{1}{2}\Delta t^2 BC\delta W^{n+\frac{1}{2}}$ and $[Q^{n+1^c}] = (1-\alpha)\Delta t^2 DC\delta W^{n+\frac{1}{2}}$, the implicit/implicit staggered time-integrator defined by Steps II1-II4 above is unconditionally stable.

Proof. For $[\delta W^{n+1^c}] = \frac{1}{2}\Delta t^2 BC\delta W^{n+\frac{1}{2}}$ and $[Q^{n+1^c}] = (1-\alpha)\Delta t^2 DC\delta W^{n+\frac{1}{2}}$, the proposed implicit/implicit staggered solution algorithm for solving Eqs. (24) becomes:

$$\boxed{\begin{aligned} \delta W^{n+1} &= \delta W^n + \Delta t A\delta W^{n+\frac{1}{2}} + \Delta t BQ^n + \frac{1}{2}\Delta t^2 BC\delta W^{n+\frac{1}{2}} \\ Q^{n+1} &= Q^n + \Delta t DQ^{n+\alpha} + \Delta t C\delta W^{n+\frac{1}{2}} + (1-\alpha)\Delta t^2 DC\delta W^{n+\frac{1}{2}} \\ \alpha &\in [\frac{1}{2},\ 1] \end{aligned}} \qquad (33)$$

The above partitioned procedure can also be written as:

$$\begin{aligned} Q^* &= Q^n + \frac{\Delta t}{2} C\delta W^{n+\frac{1}{2}} \\ Q^{**} &= Q^n + \Delta t C\delta W^{n+\frac{1}{2}} \\ \frac{\delta W^{n+1} - \delta W^n}{\Delta t} &= A\delta W^{n+\frac{1}{2}} + BQ^* \\ \frac{Q^{n+1} - Q^{**}}{\Delta t} &= D((1-\alpha)Q^{**} + \alpha Q^{n+1}) \end{aligned} \qquad (34)$$

Using the energy matrices E_A and E_D, we define the system energy as:

$$\mathcal{E}_{W,Q} = \frac{1}{2}\delta W^t E_A \delta W + \frac{1}{2} Q^t E_D Q \tag{35}$$

From Eqs. (33-35) and DEFINITION 2 and after some algebraic manipulations, it follows that:

$$\begin{aligned}
\mathcal{E}_{\delta W^{n+1},Q^n} &= \mathcal{E}_{\delta W^n,Q^n} + \Delta t \delta W^{n+\frac{1}{2}^t} E_A A \delta W^{n+\frac{1}{2}} + \Delta t Q^{*^t} B^t E_A \delta W^{n+\frac{1}{2}} \\
&\Longrightarrow \mathcal{E}_{\delta W^{n+1},Q^n} \leq \mathcal{E}_{\delta W^n,Q^n} + \Delta t Q^{*^t} B^t E_A \delta W^{n+\frac{1}{2}} \\
\mathcal{E}_{\delta W^{n+1},Q^{**}} &= \mathcal{E}_{\delta W^{n+1},Q^n} + \Delta t \delta W^{n+\frac{1}{2}^t} C^t E_D Q^n + \frac{\Delta t^2}{2} \delta W^{n+\frac{1}{2}^t} C^t E_D C \delta W^{n+\frac{1}{2}} \\
&\Longrightarrow \mathcal{E}_{\delta W^{n+1},Q^{**}} = \mathcal{E}_{\delta W^{n+1},Q^n} + \Delta t \delta W^{n+\frac{1}{2}^t} C^t E_D Q^* \\
\mathcal{E}_{\delta W^{n+1},Q^{n+1}} &= \mathcal{E}_{\delta W^{n+1},Q^{**}} + \Delta t((1-\alpha)Q^{**} + \alpha Q^{n+1})^t E_D D((1-\alpha)Q^{**} + \alpha Q^{n+1}) \\
&+ (1-2\alpha)\frac{\Delta t^2}{2}((1-\alpha)Q^{**} + \alpha Q^{n+1})^t D^t E_D D((1-\alpha)Q^{**} + \alpha Q^{n+1}) \\
&\Longrightarrow \text{For } \alpha \geq \frac{1}{2},\ \mathcal{E}_{\delta W^{n+1},Q^{n+1}} \leq \mathcal{E}_{\delta W^{n+1},Q^{**}}
\end{aligned} \tag{36}$$

which also implies that:

$$\begin{aligned}
\mathcal{E}_{\delta W^{n+1},Q^{n+1}} &\leq \mathcal{E}_{\delta W^n,Q^n} + \Delta t Q^{*^t} B^t E_A \delta W^{n+\frac{1}{2}} + \Delta t \delta W^{n+\frac{1}{2}^t} C^t E_D Q^* \\
&\Longrightarrow \mathcal{E}_{\delta W^{n+1},Q^{n+1}} \leq \mathcal{E}_{\delta W^n,Q^n} + \Delta t \delta W^{n+\frac{1}{2}^t} [E_A B + (E_D C)^t] Q^*
\end{aligned} \tag{37}$$

Finally, since the aeroelastic model problem satisfies $E_A B + (E_D C)^t = 0$, it follows that:

$$\mathcal{E}_{\delta W^{n+1},Q^{n+1}} \leq \mathcal{E}_{\delta W^n,Q^n} \tag{38}$$

which shows that the numerical energy of the system does not increase in time, and therefore the partitioned solution procedure (33) is unconditionally stable.

THEOREM 6. The implicit/implicit unconditionally stable partitioned procedure defined by Eqs. (33) is first-order accurate.

Proof. Expanding the various terms in Eqs. (33) around the time $n\Delta t$ leads to:

$$\begin{aligned}
\delta \dot{W}^n &= A\delta W^n + BQ^n + O(\Delta t) \\
\dot{Q}^n &= C\delta W^n + DQ^n + O(\Delta t)
\end{aligned}$$

Comparing the above equations with Eqs. (24) completes the proof of this theorem. Clearly, second-order accuracy would require a more sophisticated predictor in Step II1.

It is interesting to note that with the partitioned solution methodology described in Steps II1-II4, we are able to achieve unconditional stability without resorting to an augmentation technique [8,20]. Augmentation based schemes are often expensive and cumbersome to implement because they require forming and factoring the product of the independent field and coupling matrices. The staggered time-integrator described in Eqs. (33) requires only one additional sparse matrix-matrix product to form BC.

REMARK 3. Under some mild assumptions, the condition $E_A B + (E_D C)^t = 0$ can be shown to hold for two-dimensional and three-dimensional linearized aeroelastic problems. In that case, the staggered time-integrator described in Eqs. (33) is also unconditionally stable for these two-dimensional and three-dimensional problems.

4.2. Subcycling

The fluid and structure fields have often different time scales. For problems in aeroelasticity, the fluid flow usually requires a smaller temporal resolution than the structural vibration. Therefore, if the unconditionally stable staggered algorithm (33) is used to solve a coupled fluid/structure problem, the coupling time-step Δt_c will be typically dictated by the time-step Δt_F that guarantees a certain accuracy in the flow solution, rather than the time-step $\Delta t_S > \Delta t_F$ that meets the accuracy requirements of the structural field.

Using the same time-step Δt_c 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:

- savings in the overall simulation CPU time, because in that case the structural field will be advanced fewer times.
- 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 that are small enough to offset these

advantages. For example, consider the following fluid-subcycled version of the unconditionally stable staggered time-integrator given in Eqs. (33):

$$
\begin{aligned}
&\delta W^{n+1^{(0)}} = \delta W^n \\
&\{ \\
&\quad For\ k = 0,\ ...,\ n_{S/F} - 1 \\
&\quad \delta W^{n+1^{(k+1)}} = \delta W^{n+1^{(k)}} + (\Delta t A + \frac{\Delta t^2}{2} BC)\delta W^{n+1^{(k+\frac{1}{2})}} + \Delta t B Q^n \qquad (39) \\
&\} \\
&\delta W^{n+1} = \delta W^{n+1^{(n_{S/F})}} \\
&Q^{n+1} = Q^n + \Delta t D Q^{n+\alpha} + \Delta t C \delta W^{n+\frac{1}{2}} + (1-\alpha)\Delta t^2 DC \delta W^{n+\frac{1}{2}}
\end{aligned}
$$

This algorithm implements the simplest possible subcycling scheme and is often used in many applications. Unfortunately, the reader can easily check that the above fluid-subcycled partitioned procedure (39) is no longer unconditionally stable. Next, we present an improved subcycling approach that preserves the unconditional stability of the partitioned procedure (33).

THEOREM 7. For $\alpha \geq 1/2$, the following fluid-subcycled version of the staggered time-integrator given in Eqs. (33) is unconditionally stable:

$$
\begin{aligned}
&\delta W^{n+1^{(0)}} = \delta W^n \\
&X^{(0)} = Q^n \\
&\{ \\
&\quad For\ k = 0,\ ...,\ n_{S/F} - 1 \\
&\quad \delta W^{n+1^{(k+1)}} = \delta W^{n+1^{(k)}} + (\Delta t A + \frac{\Delta t^2}{2} BC)\delta W^{n+1^{(k+\frac{1}{2})}} + \Delta t B X^{(k)} \\
&\quad X^{(k+1)} = X^{(k)} + \Delta t C \delta W^{n+1^{(k+\frac{1}{2})}} \\
&\} \\
&\delta W^{n+1} = \delta W^{n+1^{(n_{S/F})}} \\
&Q^{n+1} = X^{(n_{S/F})} + n_{S/F}\Delta t D((1-\alpha)X^{(n_{S/F})} + \alpha Q^{n+1}) \\
&\alpha \in [\frac{1}{2},\ 1]
\end{aligned}
\qquad (40)
$$

Proof. The proof of this theorem is similar to that of THEOREM 5 and uses the system energy defined in Eq. (35). Using Eqs. (35,40) and DEFINITION 2

one obtains:

$$
\begin{aligned}
&\mathcal{E}_{\delta W^{n+1^{(0)}},X^{(0)}} = \mathcal{E}_{\delta W^n,Q^n} \\
&\mathcal{E}_{\delta W^{n+1^{(k+1)}},X^{(k)}} \leq \mathcal{E}_{\delta W^{n+1^{(k)}},X^{(k)}} + \Delta t \delta W^{n+1^{(k+\frac{1}{2})^t}} E_A B(X^{(k)} + \frac{\Delta t}{2} C \delta W^{n+1^{(k+\frac{1}{2})}}) \\
&\quad \Longrightarrow \mathcal{E}_{\delta W^{n+1^{(k+1)}},X^{(k)}} \leq \mathcal{E}_{\delta W^{n+1^{(k)}},X^{(k)}} + \Delta t \delta W^{n+1^{(k+\frac{1}{2})^t}} E_A B X^{(k+\frac{1}{2})} \\
&\mathcal{E}_{\delta W^{n+1^{(k+1)}},X^{(k+1)}} = \mathcal{E}_{\delta W^{n+1^{(k+1)}},X^{(k)}} + \Delta t \delta W^{n+1^{(k+\frac{1}{2})^t}} C^t E_D(X^{(k)} + \frac{\Delta t}{2} C \delta W^{n+1^{(k+\frac{1}{2})}}) \\
&\quad \Longrightarrow \mathcal{E}_{\delta W^{n+1^{(k+1)}},X^{(k+1)}} = \mathcal{E}_{\delta W^{n+1^{(k+1)}},X^{(k)}} + \Delta t \delta W^{n+1^{(k+\frac{1}{2})^t}} C^t E_D X^{(k+\frac{1}{2})} \\
&\mathcal{E}_{\delta W^{n+1},Q^{n+1}} = \mathcal{E}_{\delta W^{n+1^{(n_{S/F})}},X^{n+1^{(n_{S/F})}}} \\
&\qquad + n_{S/F} \Delta t((1-\alpha)X^{(n_{S/F})} + \alpha Q^{n+1})^t E_D D((1-\alpha)X^{(n_{S/F})} + \alpha Q^{n+1}) \\
&\qquad + (1-2\alpha)\frac{\Delta t^2}{2}((1-\alpha)X^{(n_{S/F})} + \alpha Q^{n+1})^t D^t E_D D((1-\alpha)X^{(n_{S/F})} + \alpha Q^{n+1}) \\
&\quad \Longrightarrow \text{For } \alpha \geq \frac{1}{2},\ \mathcal{E}_{\delta W^{n+1},Q^{n+1}} \leq \mathcal{E}_{\delta W^{(n_{S/F})},X^{(n_{S/F})}}
\end{aligned}
\tag{41}
$$

The above inequalities also imply that:

$$
\mathcal{E}_{\delta W^{n+1},Q^{n+1}} \leq \mathcal{E}_{\delta W^n,Q^n} + \Delta t \sum_{k=0}^{k=n_{S/F}-1} \delta W^{n+1^{(k+\frac{1}{2})^t}} [E_A B + (E_D C)^t] X^{(k+\frac{1}{2})} \tag{42}
$$

Finally, since the aeroelastic model problem satisfies $E_A B + (E_D C)^t = 0$, it follows that:

$$
\mathcal{E}_{\delta W^{n+1},Q^{n+1}} \leq \mathcal{E}_{\delta W^n,Q^n} \tag{43}
$$

which shows that the numerical energy of the system does not increase in time, and therefore the partitioned and fluid-subcycled solution procedure (40) is unconditionally stable.

THEOREM 8. The implicit/implicit unconditionally stable partitioned and fluid-subcycled procedure defined by Eqs. (40) is first-order accurate.

Proof. The proof of this theorem is similar to that of THEOREM 6.

The subcycling approach advocated in Eqs. (40) preserves the computational advantages of subcycling. At each stage, the evaluation by the fluid solver of the correction term $X^{(k+1)}$ does not require neither advancing the structural state vector, nor exchanging information with the structural solver. Moreover,

updating $X^{(k+1)}$ and $BX^{(k)}$ requires only two sparse matrix-vector products and therefore is relatively inexpensive.

The interpretation of the role of the correction term $X^{(k+1)}$ goes as follows. In order to solve $\dot{Q} = DQ + C\delta W$ between t_n and t_{n+1}, the structure kernel must receive from the fluid module the best possible approximation of the coupling quantity $\int_{t_n}^{t_{n+1}} C\delta W dt$. In the fluid-subcycled partitioned procedure (39), this integral is approximated by $\Delta t C\delta W^{n+\frac{1}{2}}$, which guarantees a certain accuracy but does not warrant unconditional stability. On the other hand, the strategy consisting in approximating $\int_{t_n}^{t_{n+1}} C\delta W dt$ via updating $X^{(k+1)} = X^{(k)} + \Delta t C\delta W^{n+1^{(k+\frac{1}{2})}}$ and replacing in the procedure (39) the "frozen" $\Delta t BQ^n$ by the updated quantity $\Delta t BX^{(k)}$ not only provides a better coupling accuracy, but also preserves the unconditional stability of the original non subcycled partitioned procedure (33).

The implications of the above results and discussion on the staggered and subcycled solution of more complex aeroelastic problems can be formulated as follows. When the fluid field is subcycled and updated ahead of the structural field, then:

- the motion of the moving fluid boundary induced by the structural deformation should not be completely absorbed during the first fluid subcycle and "frozen" during the remaining ones. Rather, this induced motion should be distributed among all subcycle stages via a careful interpolation scheme.
- after all fluid subcycles are completed, the mean value rather than the final value of the pressure field must be transmitted to the structure.

4.3. Examples

Here, we illustrate the numerical properties of the family of implicit/implicit staggered procedures presented in Sections 4.1-4.2 with the solution of the aeroelastic model problem (24). We consider the case where $\alpha = 1/2$ and the structure is undamped ($d = 0$). First, we introduce the non-dimensional variables:

$$\begin{aligned} \bar{t} &= \frac{c_0}{l_0} t, \quad \bar{x} = \frac{x}{l_0}, \quad \bar{q} = \frac{q}{l_0} \\ \bar{\rho} &= \frac{\rho}{\rho_0}, \quad \overline{\rho u} = \frac{\rho u}{\rho_0 c_0}, \quad \delta\overline{W} = \begin{pmatrix} \delta\bar{\rho} \\ \delta\overline{\rho u} \end{pmatrix} \end{aligned} \tag{44}$$

and rewrite Eqs. (19) and (4) in non-dimensional form as follows:

$$\Delta\bar{x}_j\ \delta W'_j - \overline{J_0^+}(0)\delta\overline{W}_{j-1} + (\overline{J_0^+}(0) - \overline{J_0^-}(0))\delta\overline{W}_j + \overline{J_0^-}(0)\delta\overline{W}_{j+1} = 0 \quad (j \neq N)$$

$$\Delta\bar{x}_N\ \delta W'_N - \overline{J_0^+}(0)\delta\overline{W}_{N-1} + (\overline{J_0^+}(0) - \overline{J_0^-}(0))\delta\overline{W}_N - \begin{pmatrix} \bar{q}' \\ 0 \end{pmatrix} = 0$$

$$\bar{q}'' + \bar{\omega}_s^2\bar{q} = \frac{1}{\bar{m}}(\delta\bar{p}_N - 1)$$

where

$$' = \frac{d}{d\bar{t}}$$

$$\overline{J_0^+}(0) = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad \overline{J_0^-}(0) = \frac{1}{2}\begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$\bar{\omega}_s^2 = \frac{l_0^2}{c_0^2}\frac{k}{m}$$

$$\bar{m} = \frac{m}{\rho_0 l_0} \tag{45}$$

Next, we discretize the piston chamber into 21 grid points and $N = 20$ finite volume cells, and set the non-dimensional parameters to $\bar{\omega}_s^2 = 3.03 \times 10^{-4}$ and $\bar{m} = 30.77$. We consider the following initial conditions:

$$\delta\bar{p}(\bar{t}=0) = \delta\overline{\rho u}(\bar{t}=0) = 0$$
$$\bar{q}(\bar{t}=0) = 0 \quad \dot{\bar{q}}(\bar{t}=0) = 1$$

and solve the coupled equations (45) using eight time-steps varying between $\Delta t = 1\times\text{CFL}$ and $\Delta t = 128\times\text{CFL}$, where $\text{CFL} = l_0/(Nc_0)$ is computed with respect to the uncoupled fluid problem. The obtained non-dimensional piston displacement $\bar{q}/l_0$ and fluid pressure in the cell in contact with the piston $(p/\rho_0 c_0^2)_{20}$ are depicted in Fig. 3-4 for the case without subcycling.

Clearly, the results reported in Fig. 3-4 highlight the unconditional stability of the family of implicit/implicit staggered procedures presented in Sections 4.1-4.2. Stable responses are observed for all time-steps, and accurate results are obtained for both the piston displacement and fluid pressure for time-steps as large as $\Delta t = 32\times\text{CFL}$.

The previous computations are repeated using the fluid-subcycled staggered time-integrator (40), $2\times\text{CFL} \leq \Delta t_F \leq 4\times\text{CFL}$, and several subcycling factors $1 \leq n_{S/F} \leq 32$. The corresponding results (Fig. 5-8) confirm numerically the unconditional stability proved mathematically in THEOREM 7. Note however

that large subcycling factors $n_{S/F}$ introduce a spurious phase shift in the initial stages of the coupled computations that can ruin the accuracy of the response history. Both amplitude and phase errors can also be observed in that case. This suggests that an adaptive time-stepping strategy is needed in order to resolve better the initial response of the coupled system.

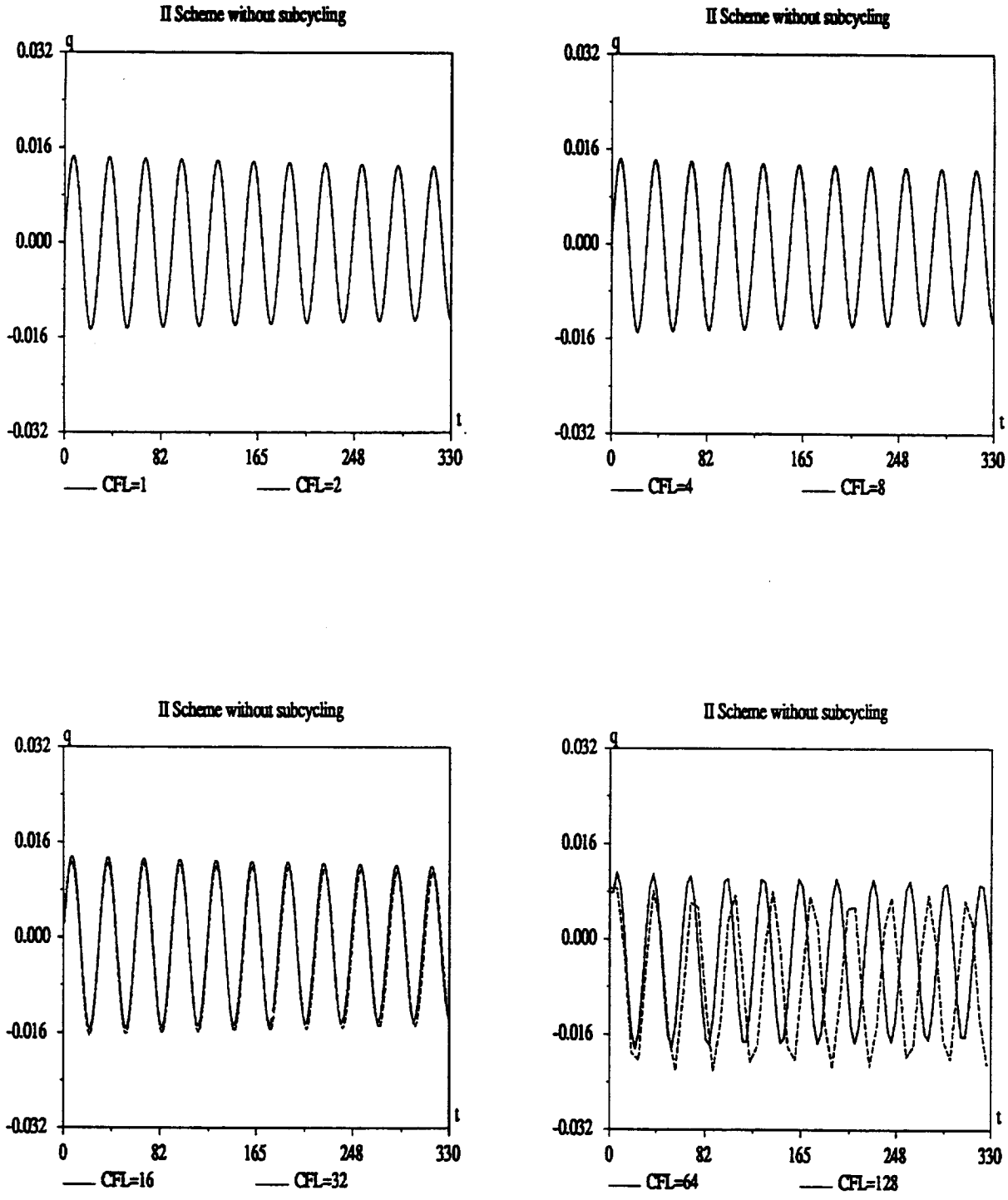

Fig. 3. Computed non-dimensional piston displacement (without subcycling)

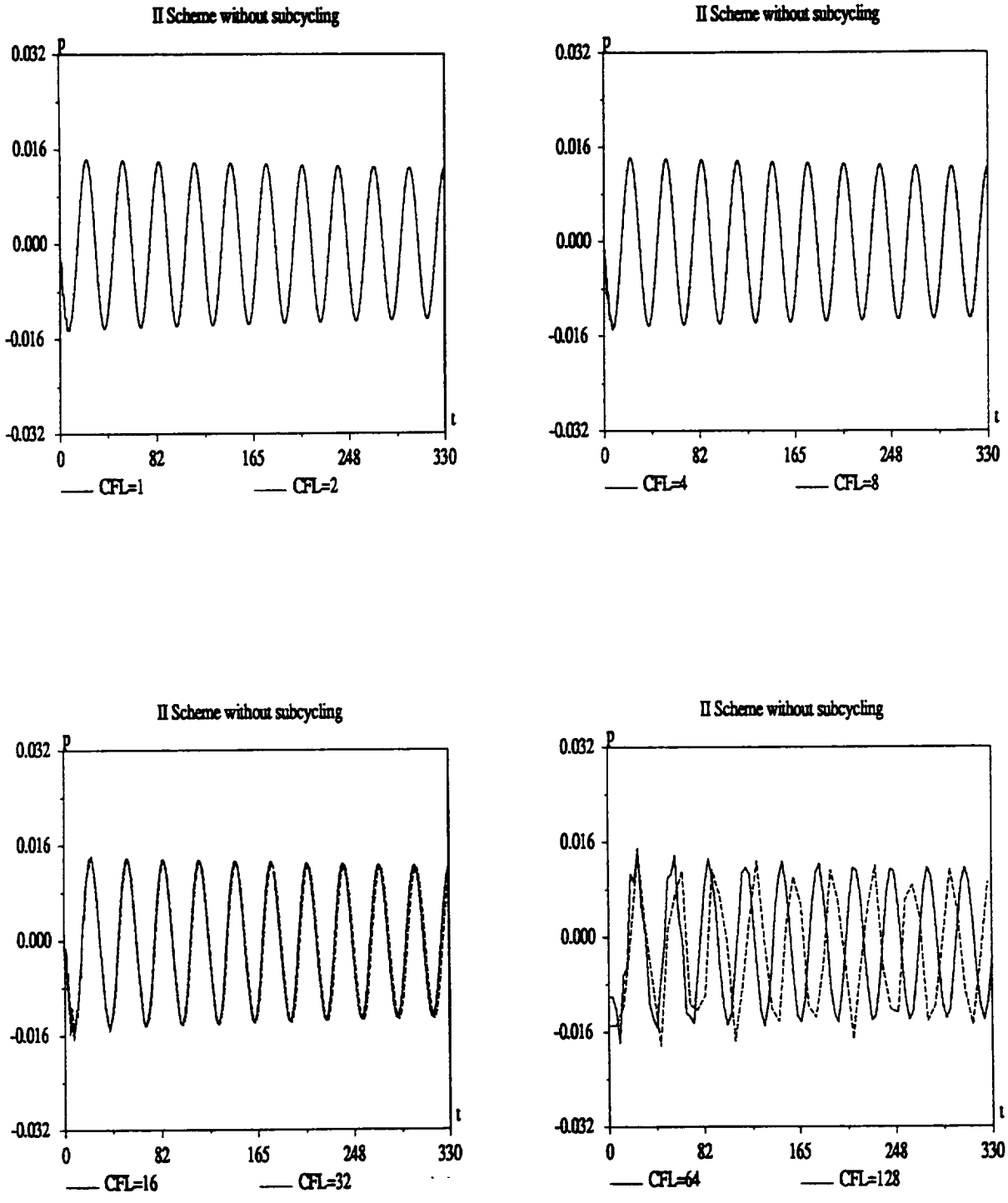

Fig. 4. Computed non-dimensional fluid pressure (without subcycling)

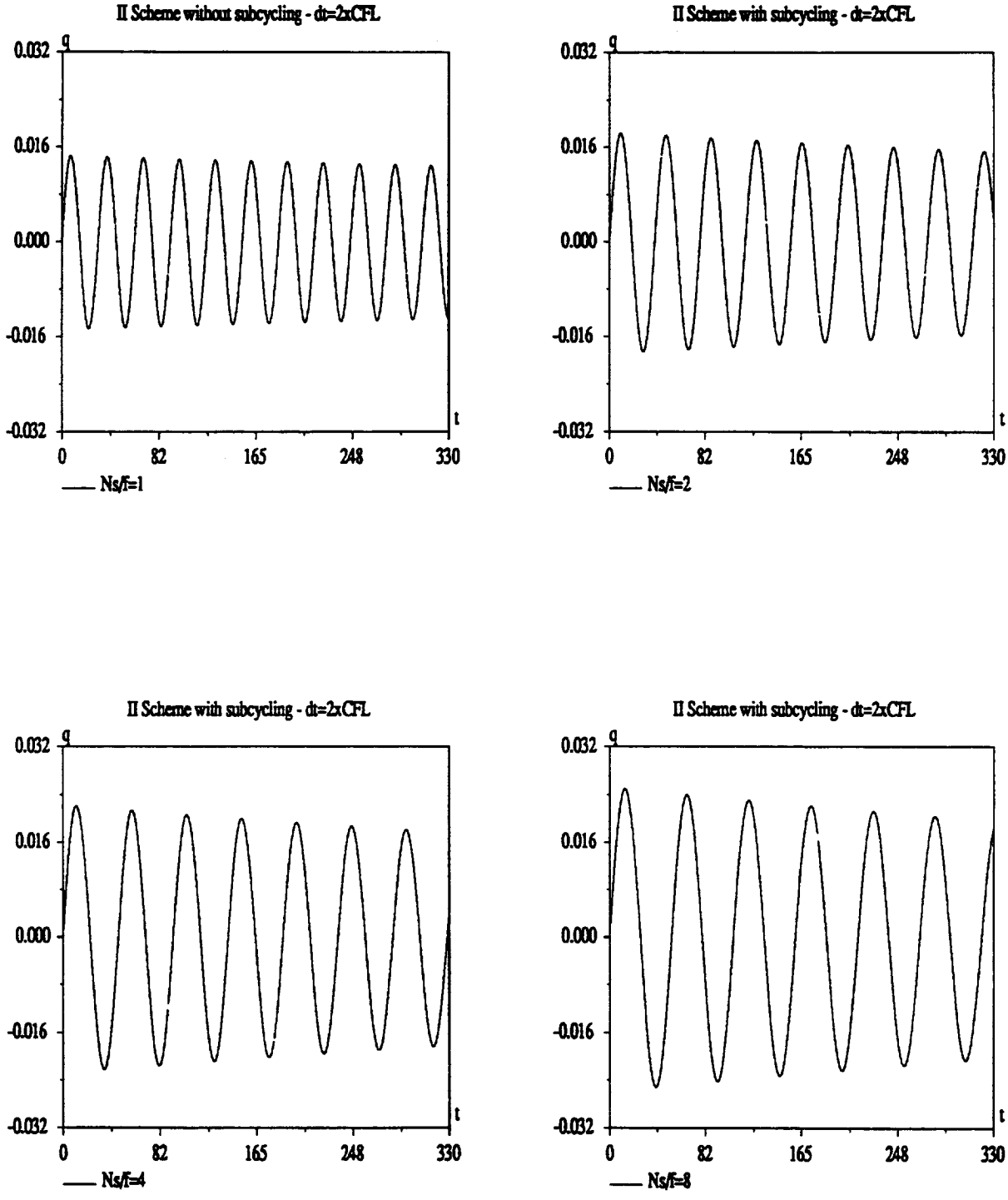

Fig. 5. Computed non-dimensional piston displacement ($\Delta t = 2\times\text{CFL}$ and with subcycling)

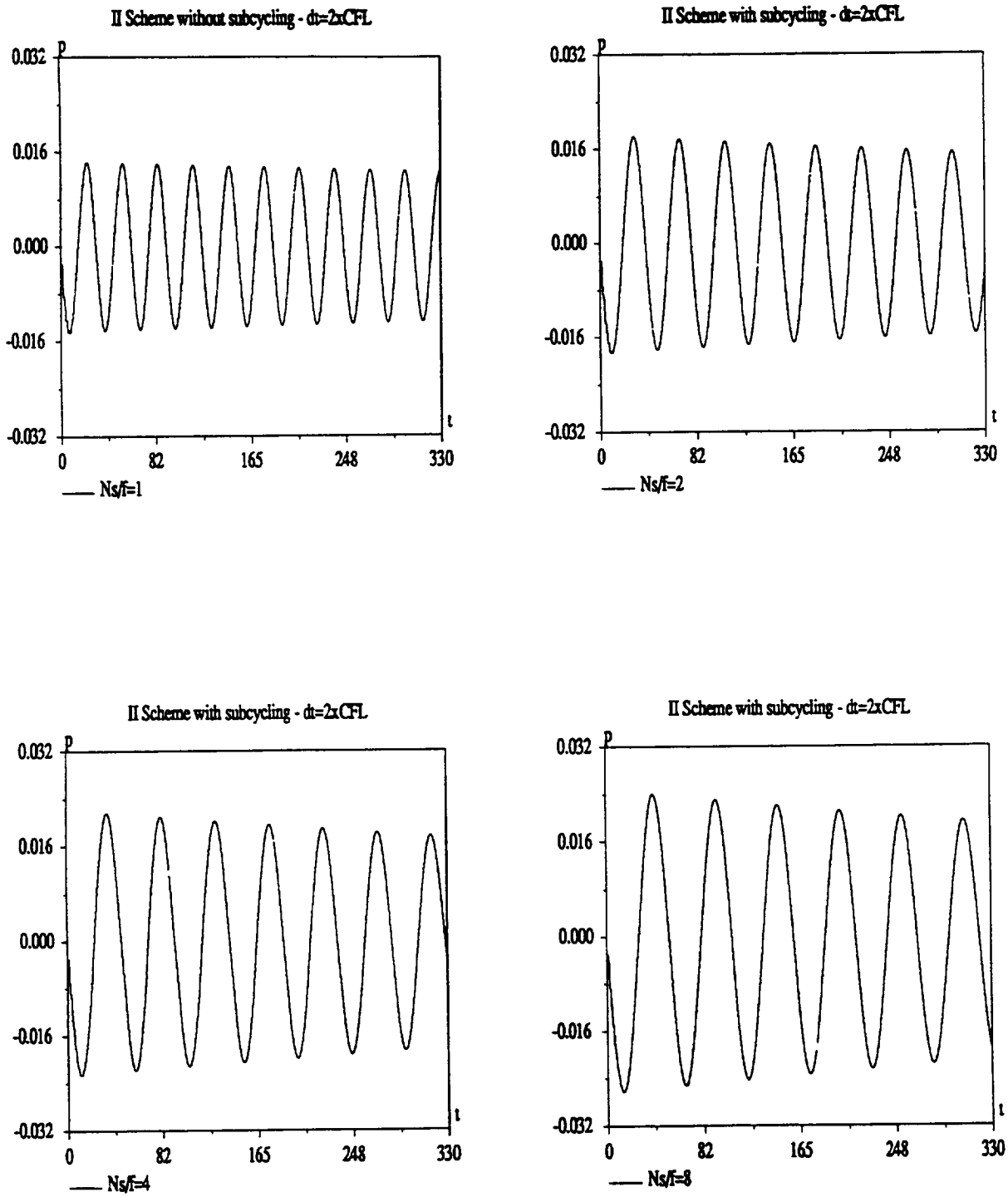

Fig. 6. Computed non-dimensional fluid pressure ($\Delta t = 2\times\text{CFL}$ and with subcycling)

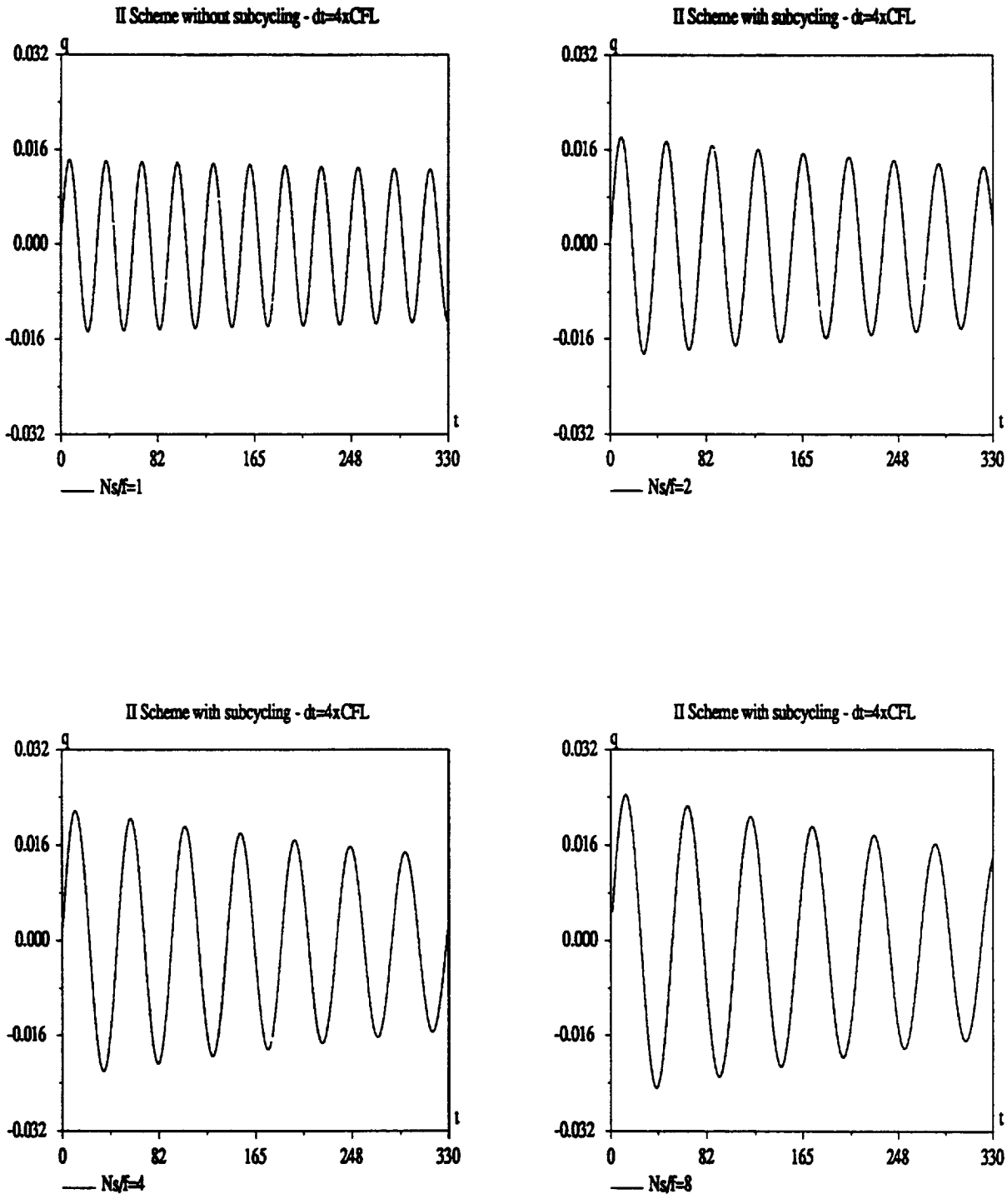

Fig. 7. Computed non-dimensional piston displacement ($\Delta t = 4 \times \mathrm{CFL}$ and with subcycling)

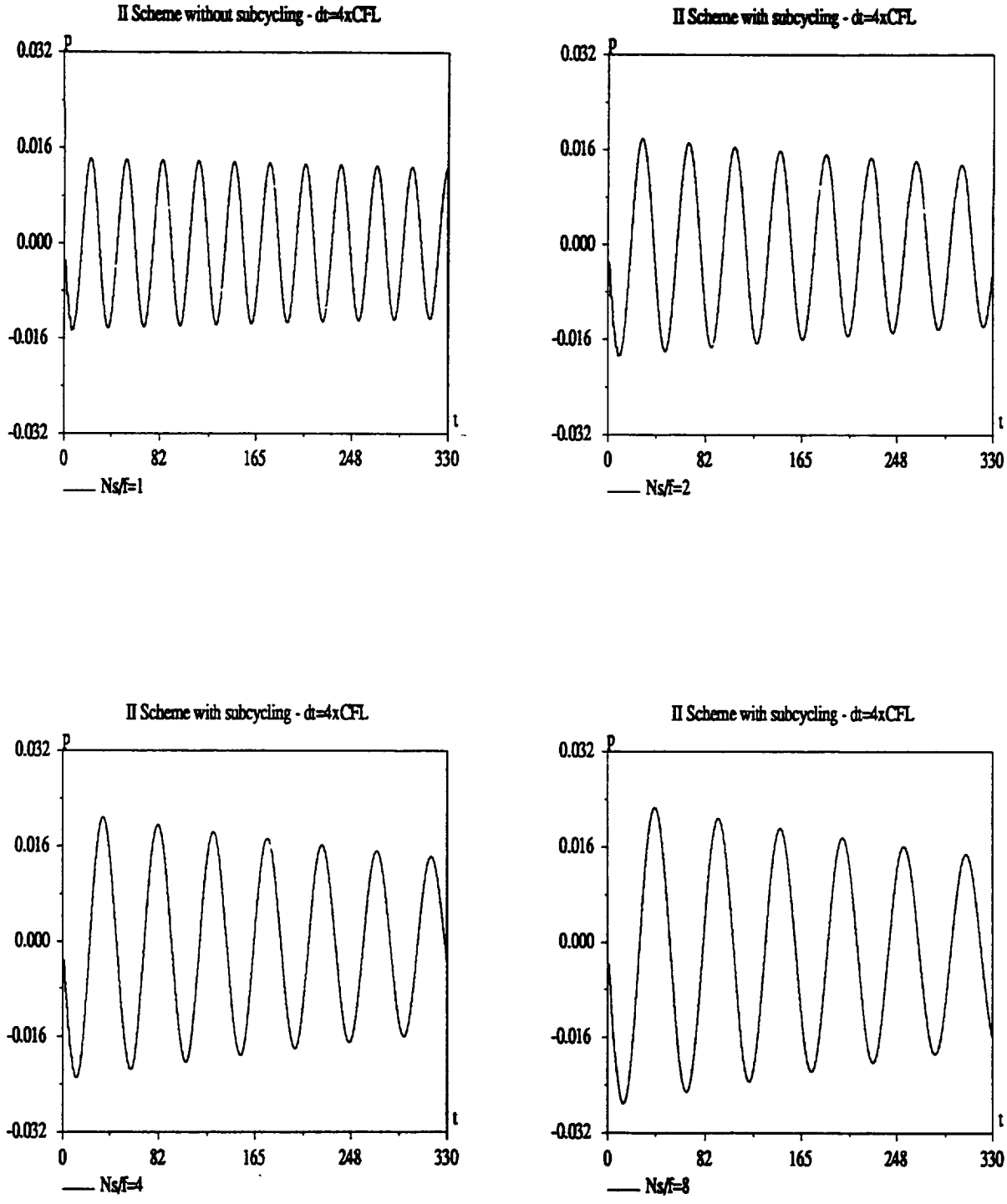

Fig. 8. Computed non-dimensional fluid pressure ($\Delta t = 4 \times \mathrm{CFL}$ and with subcycling)

5. A family of explicit/implicit fluid/structure partitioned procedures

Next, we consider the case where an explicit scheme is desired for advancing the fluid field. Consequently, we cannot design an unconditionally stable staggered algorithm for solving the coupled fluid/structure problem, because in that case it would be only conditionally consistent [21]. Rather, we focus on developing a family of partitioned procedures whose stability limit is governed by the critical time-step of the fluid solver. In other words, we wish to design a staggered solution algorithm for the coupled problem whose stability limit is not worse than that of the underlying fluid explicit time-integrator. This is not necessarily a trivial task because coupling effects can restrict the stability limits of the independent field time-integrators.

5.1. A predictor-corrector approach

Here, we present a family of explicit/implicit staggered algorithms for solving the fluid/structure Eqs. (24) that is based on similar ideas to those presented in Section 4:

Step EI1. Predict the structural field using the value computed at $t_n = n\Delta t$:

$$Q^p = Q^n$$

Step EI2. Predict the fluid system using the forward Euler explicit scheme:

$$\delta W^p = \delta W^n + \Delta t A \delta W^n + \Delta t B Q^p$$

Step EI3. Improve the structural field using an implicit generalized trapezoidal method and the predicted fluid state δW^p:

$$\overline{Q^{n+1}} = Q^n + \Delta t D \overline{Q^{n+\alpha}} + \Delta t C \delta W^p \qquad \alpha \in]0,\ 1]$$

Step EI4. Correct the expressions yielding the fluid and structural fields to enhance the stability of the explicit/implicit staggered procedure:

$$\begin{aligned} \delta W^{n+1} &= \delta W^p + [\delta W^{n+1^c}] \\ Q^{n+1} &= \overline{Q^{n+1}} + [Q^{n+1^c}] \end{aligned}$$

When applied to the solution of the uncoupled linearized fluid system (Eqs. (17)), the forward Euler algorithm is first-order accurate and stable for CFL ≤ 1.

For $\alpha \geq 1/2$, the generalized trapezoidal family of methods is unconditionally stable when applied to time-integrate the uncoupled structural problem. For $\alpha \in]0,\ 1]$, these methods are first-order accurate, except for $\alpha = 1/2$, in which case the corresponding scheme is second-order accurate. The correction terms $[\delta W^{n+1^c}]$ and $[Q^{n+1^c}]$ should be computed to enhance the stability of the resulting explicit/implicit staggered solution procedure.

THEOREM 9. For $\alpha \geq 1/2$, $[\delta W^{n+1^c}] = \alpha \Delta t B(Q^{n+1} - Q^n)$ and $[Q^{n+1^c}] = -\Delta t C(\Delta t B Q^n + \delta W^{n+1^c})/2$, the stability of the explicit/implicit staggered time-integrator defined by Steps EI1-EI4 above is governed by the stability of the explicit time-integrator of the uncoupled fluid problem.

Proof. For $[\delta W^{n+1^c}] = \alpha \Delta t B(Q^{n+1} - Q^n)$ and $[Q^{n+1^c}] = -\Delta t C(\Delta t B Q^n + \delta W^{n+1^c})/2$, the proposed explicit/implicit staggered solution algorithm for solving Eqs. (24) becomes:

$$\boxed{\begin{aligned} Q^{n+1} &= Q^n + \Delta t D Q^{n+\alpha} + \Delta t C(\delta W^n + \Delta t A \delta W^n + \frac{\Delta t}{2} B Q^{n+\alpha}) \\ \delta W^{n+1} &= \delta W^n + \Delta t A \delta W^n + \Delta t B Q^{n+\alpha} \end{aligned}} \tag{46}$$

Let E^n be defined as follows:

$$E^n = \frac{1}{2} Q^{n\,t} E_D Q^n$$

If the second of Eqs. (46) is re-written as:

$$\begin{aligned} Q^{n+1} &= Q^n + \Delta t D Q^{n+\alpha} + \Delta t Z \\ &\text{where} \\ Z &= C(\delta W^n + \Delta t A \delta W^n + \frac{\Delta t}{2} B Q^n) \end{aligned}$$

it follows that for $\alpha \geq \frac{1}{2}$ we have:

$$E^{n+1} \leq E^n + \Delta t Q^{n+\alpha\,t} E_D Z$$

Using the above inequality and calculations similar to those in (36), one can show that if the energy of the uncoupled fluid problem decreases in time, then the energy of the coupled system also decreases in time. Hence, the stability of the explicit/implicit staggered procedure defined by Eqs. (46) is governed by the CFL condition (CFL ≤ 1) of the explicit scheme applied to the uncoupled fluid problem.

THEOREM 10. The explicit/implicit partitioned procedure defined by Eqs. (46) is first-order accurate.

Proof. The proof of this theorem is similar to that of THEOREM 6.

REMARK 2. Note that while the family of implicit/implicit partitioned procedures (33) and its fluid-subcycled version (40) require updating the fluid flow ahead of the structural field, the family of explicit/implicit staggered algorithms (46) require updating the structural system first.

5.2. Examples

Here, we illustrate the numerical properties of the family of explicit/implicit staggered procedures presented in Section 5.1 with the solution of the non-dimensional coupled equations (45). We consider the case $\alpha = 1/2$, and use the same non-dimensional parameters and finite volume mesh as in Section 4.3.

First, we solve the coupled problem for $0.5\times\text{CFL} \leq \Delta t \leq 1.0\times\text{CFL}$, where the CFL is with respect to the uncoupled fluid problem. The obtained non-dimensional piston displacement $\bar{q}/l_0$ and fluid pressure in the cell in contact with the piston $(p/\rho_0 c_0^2)_{20}$ are reported in Fig. 9-10 for the case without subcycling. Clearly, these results demonstrate numerical stability for $\Delta t \leq 1.0\times\text{CFL}$. However, they also show that at time-steps close to the uncoupled fluid CFL condition, the errors introduced in the initial stages of the computations propagate throughout the entire history response of the fluid pressure, but do not affect significantly the evaluation of the structural displacement.

Next, we repeat the previous explicit/implicit computations and subcycle the fluid system. We use a subcycling scheme similar to that introduced in Eqs. (40) in order to maximize the coupled stability time-step. In that case, the numerical results reported in Fig. 11 for the non-dimensional piston displacement $\bar{q}/l_0$ indicate that there exists a maximum subcycling factor beyond which the explicit/implicit time-integrator (46) with subcycling becomes numerically unstable.

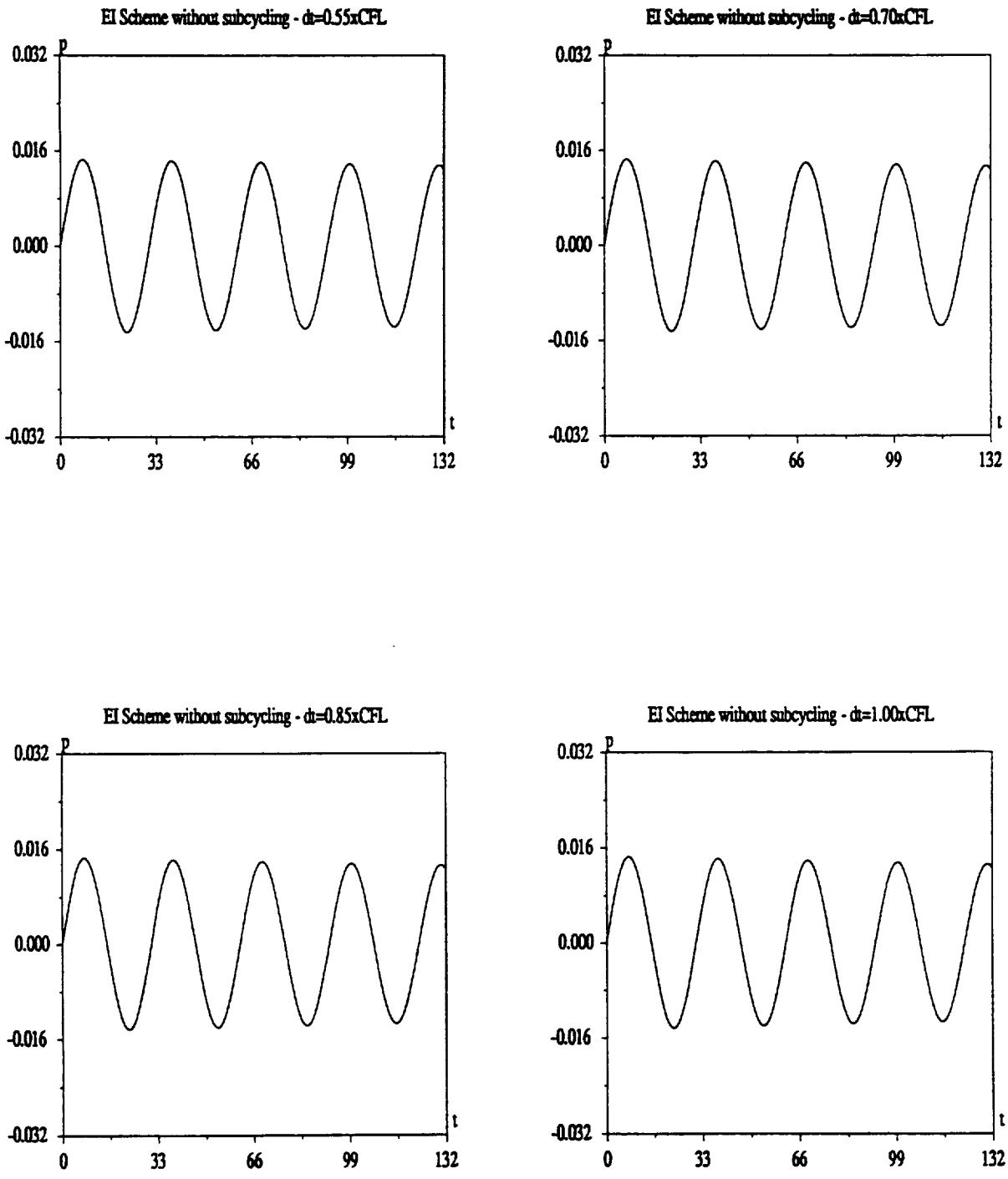

Fig. 9. Computed non-dimensional piston displacement (without subcycling)

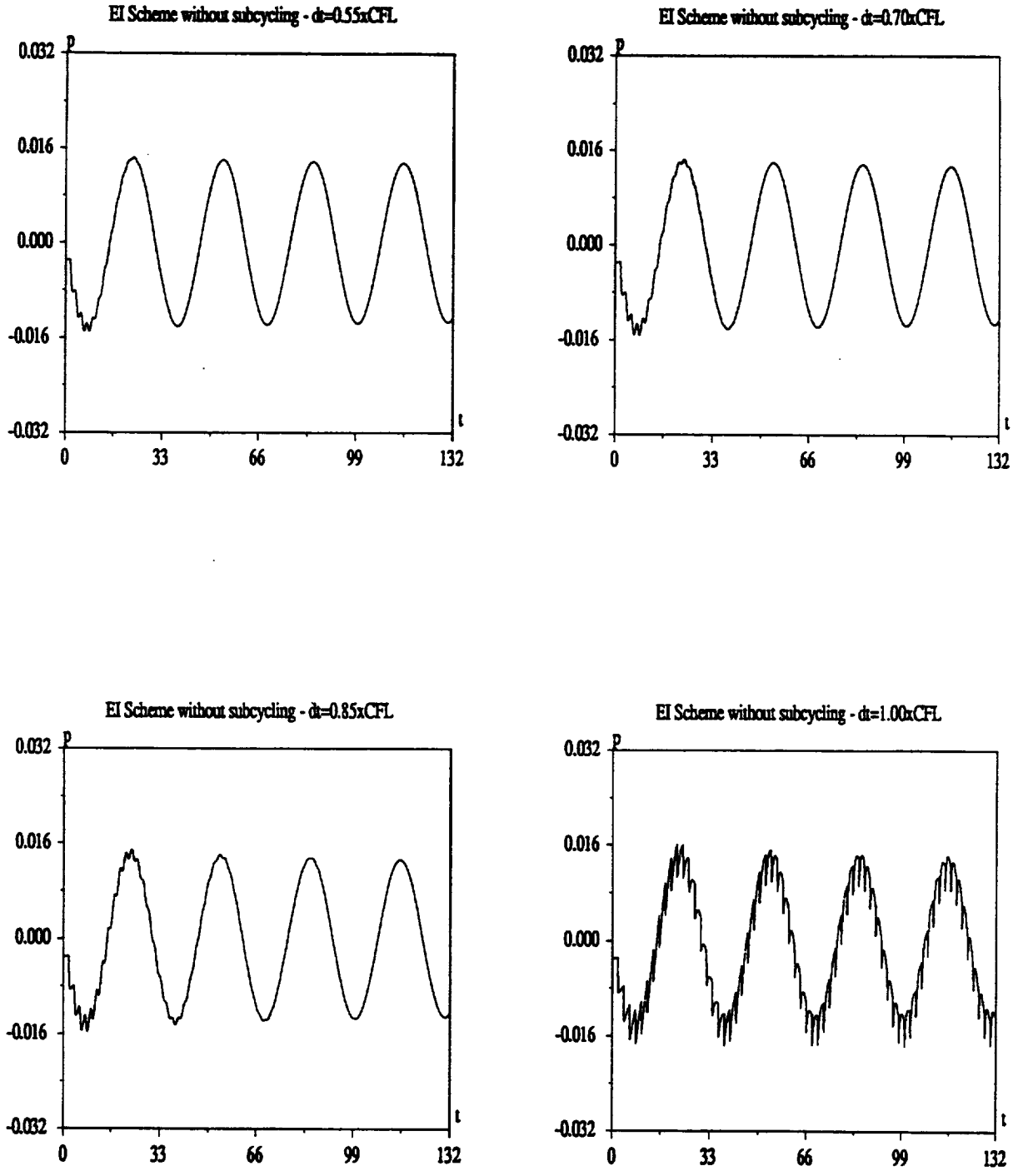

Fig. 10. Computed non-dimensional fluid pressure (without subcycling)

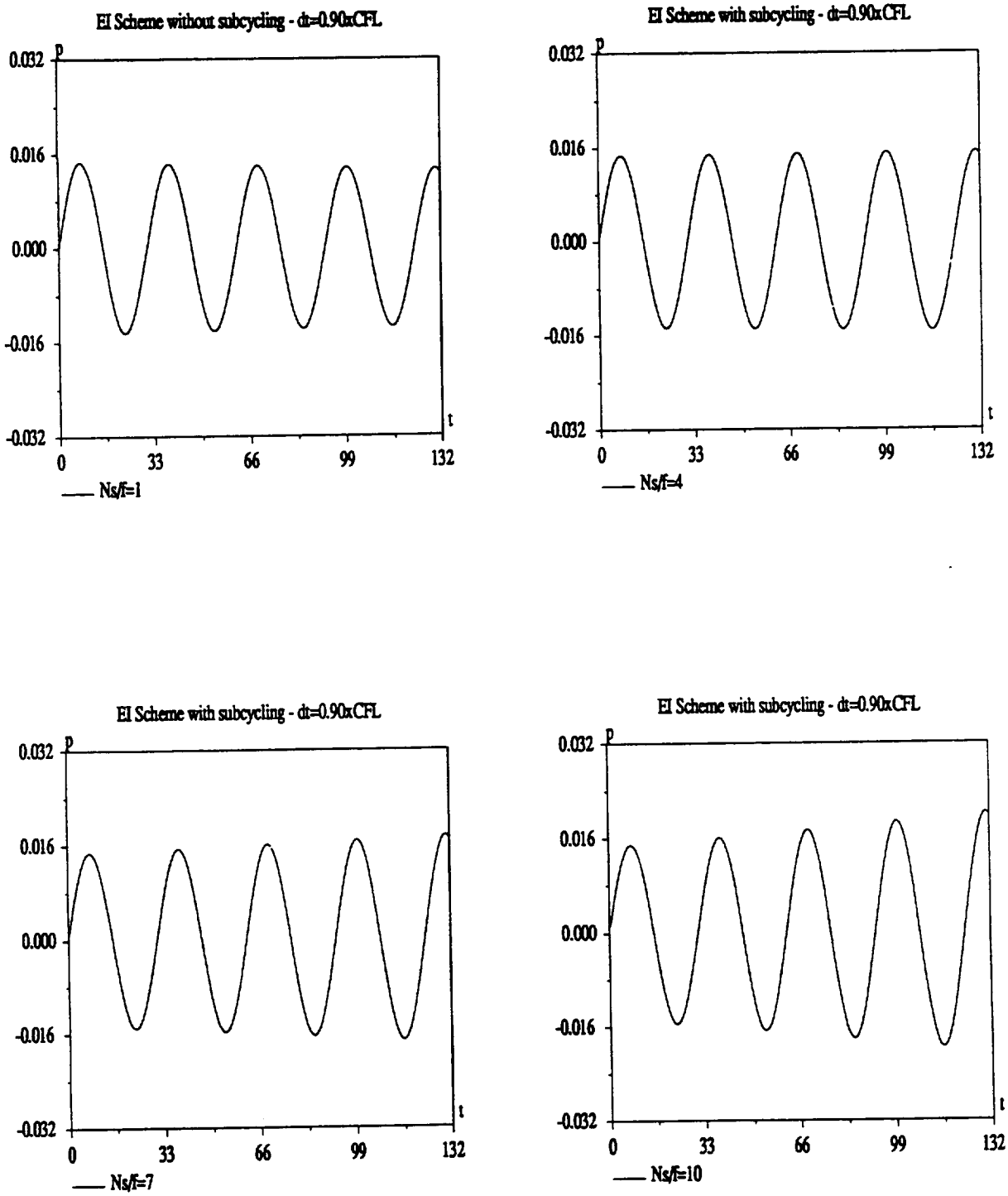

Fig. 11. Computed non-dimensional piston displacement ($\Delta t = 0.9 \times \text{CFL}$ and with subcycling)

6. Parallel staggering strategies, error analysis, and CPU distribution

The family of partitioned procedures presented in Sections 4 and 5 are inherently sequential: in the implicit/implicit case, the fluid state vector must be updated before the structural system can be advanced, and in the explicit/implicit case, the new structural displacements and velocities must be computed before the new fluid state vector can be evaluated. With the advent of parallel processors and distributed computing platforms, it becomes interesting not only to parallelize each field computations, but also to design staggered time-integration algorithms that promote inter-field parallelism — that is, that allow advancing simultaneously the fluid and structural systems. In this Section, we present such partitioning procedures and discuss their accuracy v.s. speed trade-offs. We consider only explicit/implicit algorithms. More specifically, we focus on the case where the fluid is advanced using the first-order accurate forward Euler scheme and the structure is advanced using the second-order accurate midpoint rule, because these algorithms are already available in our large-scale simulation parallel software. We use the family of time-integrators (46) as reference, and therefore begin with their error analysis.

We introduce the following nomenclature:

O_F : number of floating-point operations in one fluid time-step

O_S : number of floating-point operations in one structural time-step

T_F : fluid-to-structure single pass transfer time

T_S : structure-to-fluid single pass transfer time

CPU_F : CPU resource allocated to the fluid kernel

CPU_S : CPU resource allocated to the structure kernel

For every partitioned procedure, we give the resource distribution between the fluid and structure kernels for simulations on heterogeneous platforms.

6.1. Algorithm ALG0: *the basic staggered scheme*

In the sequel, we refer to the explicit/implicit time-integrator (46) as the basic staggered algorithm ALG0. Let Z be defined as follows:

$$Z = \begin{pmatrix} Q \\ \delta W \end{pmatrix} \tag{47}$$

Considering the time-interval $[t_n = n\Delta t,\ t_{n+n_{S/F}} = (n + n_{S/F})\Delta t]$, it follows from Eqs. (46-47) that:

$$Z^{n+n_{S/F}} = [I + n_{S/F}\Delta t \begin{pmatrix} D & C \\ B & A \end{pmatrix} + n_{S/F}^2 \frac{\Delta t^2}{2} \begin{pmatrix} D^2 + BC & DC + CA \\ BD + AB & BC + A^2 \end{pmatrix} + n_{S/F} \frac{\Delta t^2}{2} \begin{pmatrix} 0 & CA \\ -AB & A^2 \end{pmatrix} + O\left(\Delta t^3\right)] Z^n \tag{48}$$

If $ER_{\delta W}$ and ER_Q denote respectively the errors in the fluid and structural responses after $n_{S/F}$ time-steps, it follows from (48) that:

$$\boxed{\begin{aligned} ER_{\delta W}^{ALG0} &= -n_{S/F} \frac{\Delta t^2}{2} A \delta \dot{W}^n + O(\Delta t^3) \\ ER_Q^{ALG0} &= n_{S/F} \frac{\Delta t^2}{2} CAQ^n + O\left(\Delta t^3\right) \end{aligned}} \tag{49}$$

which shows that ALG0 is first-order accurate. Hence, the accuracy of the structural computation is first-order even though the midpoint rule applied to the uncoupled structural problem is second-order accurate.

The basic steps of ALG0 are graphically depicted in Fig. 12. The CPU time needed to advanced the coupled solution $n_{S/F}\Delta t$ is equal to:

$$T_{coupled} = n_{S/F}(T_F + T_S + \frac{O_F}{CPU_F} + \frac{O_S}{CPU_S}) \tag{50}$$

If the total amount of CPU resources $CPU = CPU_F + CPU_S$ is assumed to be fixed, $T_{coupled}$ is minimum for:

$$\frac{CPU_F}{CPU_S} = \sqrt{\frac{O_F}{O_S}}$$

which gives:

$$\boxed{T_{coupled}^{ALG0} = n_{S/F}(T_F + T_S + \frac{O_F + O_S + 2\sqrt{O_F O_S}}{CPU})} \tag{51}$$

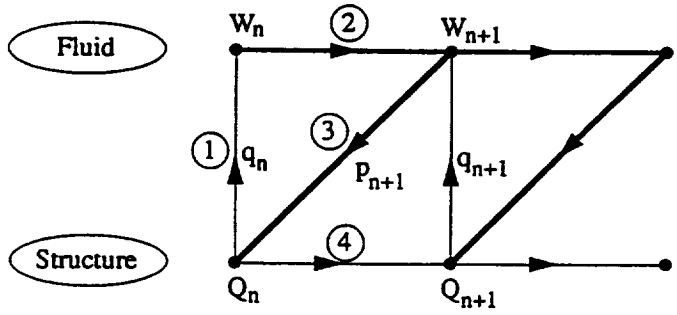

Fig. 12. ALG0: the basic staggered algorithm

6.2. Algorithm ALG1: *subcycling the fluid system*

A fluid-subcycled version of a slightly modified ALG0 where the subcycling scheme follows the guidelines of THEOREM 7 is referred to as ALG1 in the sequel and is given by:

$$
\begin{aligned}
&\delta W^{n+1^{(0)}} = \delta W^n \\
&X^{(0)} = Q^n \\
&\{ \\
&\quad For\ k = 0,\ ...,\ n_{S/F} - 1 \\
&\quad \delta W^{n+1^{(k+1)}} = \delta W^{n+1^{(k)}} + \Delta t A \delta W^{n+1^{(k)}} + \Delta t B Q^n \\
&\quad X^{(k+1)} = X^{(k)} + \Delta t C \delta W^{n+1^{(k+\frac{1}{2})}} \\
&\} \\
&\delta W^{n+1} = \delta W^{n+1^{(n_{S/F})}} \\
&Q^{n+1} = Q^{n+n_{S/F}} = X^{(n_{S/F})} + n_{S/F} \Delta t D(\frac{Q^n + Q^{n+n_{S/F}}}{2})
\end{aligned}
\tag{52}
$$

Essentially, the fluid system is subcycled during $n_{S/F}$ time-steps Δt, and the structural field is advanced in one shot a large time-step equal to $n_{S/F}\Delta t$ using the average fluid pressure between t_n and $t_{n+n_{S/F}}$.

Expanding the various terms in Eqs. (52) around $t_n = n\Delta t$ leads to:

$$\boxed{\begin{aligned} ER_{\delta W}^{ALG1} &= -n_{S/F}^2 \frac{\Delta t^2}{2} B\delta\dot{W}^n + O\left(\Delta t^3,\ n_{S/F}\Delta t^2\right) \\ ER_{Q}^{ALG1} &= n_{S/F}^3 \frac{\Delta t^3}{12}(D^3Q^n + 2C\delta\ddot{W}^n - 2CB\dot{Q}^n) + O\left(\Delta t^4,\ n_{S/F}^2\Delta t^3\right) \end{aligned}} \tag{53}$$

which shows that ALG1 is also first-order accurate. However, from (49) and (53), it follows that subcycling amplifies the fluid errors by the subcycling factor $n_{S/F}$. Measuring the effect of subcycling on the structural errors of ALG0 is less trivial: in order to keep its computer implementation simple, we have designed ALG1 as a fluid-subcycled version of a "slightly modified" rather than the "original" ALG0. Consequently, the structural errors grow as $O\left(n_{S/F}\Delta t^2\right)$ in ALG0, and as $O\left(n_{S/F}^3\Delta t^3\right)$ in ALG1.

The basic steps of ALG1 are graphically depicted in Fig. 13. The CPU time needed to advanced the coupled solution $n_{S/F}\Delta t$ is equal to:

$$T_{coupled} = T_F + T_S + \frac{n_{S/F}O_F}{CPU_F} + \frac{O_S}{CPU_S} \tag{54}$$

For a fixed total amount of CPU resources $CPU = CPU_F + CPU_S$, $T_{coupled}$ is minimum for:

$$\frac{CPU_F}{CPU_S} = \sqrt{\frac{n_{S/F}O_F}{O_S}}$$

which gives:

$$\boxed{T_{coupled}^{ALG1} = T_F + T_S + \frac{n_{S/F}O_F + O_S + 2\sqrt{n_{S/F}O_F O_S}}{CPU}} \tag{55}$$

The comparison of $T_{coupled}^{ALG0}$ and $T_{coupled}^{ALG1}$ highlights the computational advantages of subcycling.

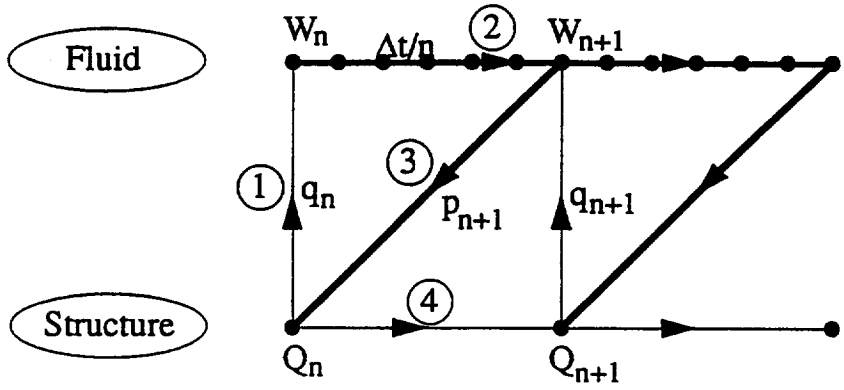

Fig. 13. The fluid-subcycled ALG1 staggered algorithm

6.3. Algorithm ALG2: *improving the accuracy of* ALG1

In order to improve the accuracy of the fluid solution in ALG1, we introduce a computational phase shift between the fluid and structure kernels equal to $n_{S/F}\Delta t/2$. Assuming that δW^n and $Q^{n+\frac{1}{2}}$ are available, the improved subcycled explicit/implicit algorithm ALG2 computes δW^{n+1} and $Q^{n+\frac{3}{2}}$ as follows:

$$
\begin{aligned}
&\delta W^{n+1^{(0)}} = \delta W^n \\
&X^{(0)} = Q^{n+\frac{1}{2}} = Q^{n+\frac{n_{S/F}}{2}} \\
&\{ \\
&\quad For\ k = 0, \ldots, n_{S/F} - 1 \\
&\quad \delta W^{n+1^{(k+1)}} = \delta W^{n+1^{(k)}} + \Delta t A \delta W^{n+1^{(k)}} + \Delta t B Q^{n+\frac{n_{S/F}}{2}} \\
&\quad X^{(k+1)} = X^{(k)} + \Delta t C \delta W^{n+1^{(k+\frac{1}{2})}} \\
&\} \\
&\delta W^{n+1} = \delta W^{n+1^{(n_{S/F})}} \\
&Q^{n+\frac{3}{2}} = Q^{n+\frac{1}{2}+n_{S/F}} = X^{(n_{S/F})} + n_{S/F}\Delta t D\left(\frac{Q^{n+\frac{1}{2}} + Q^{n+\frac{1}{2}+n_{S/F}}}{2}\right)
\end{aligned}
\tag{56}
$$

Algorithm ALG2 has the same computational and I/O transfer requirements as ALG1. However, its error analysis leads to:

$$\boxed{\begin{aligned} ER_{\delta W}^{ALG2} &= -n_{S/F}\frac{\Delta t^2}{2}A\delta\dot{W}^n + O\left(\Delta t^3\right) \\ ER_{Q}^{ALG2} &= O\left(\Delta t^4,\ n_{S/F}^3\Delta t^3\right) \end{aligned}} \tag{57}$$

A direct comparison of (49), (53) and (57) shows that ALG2 offers the computational advantages of ALG1, and the higher accuracy of ALG0.

6.4. Algorithm ALG3: *introducing inter-field parallelism*

ALG0, ALG1 and ALG2 are inherently sequential. In all three algorithms, the fluid system must be updated before the structural system can be advanced. The following explicit/implicit fluid-subcycled time-integrator ALG3 introduces inter-field parallelism in the solution of Eqs. (24):

$$\boxed{\begin{aligned} &\delta W^{n+1^{(0)}} = \delta W^n \\ &\{ \\ &\quad For\ k = 0,\ ...,\ n_{S/F} - 1 \\ &\quad \delta W^{n+1^{(k+1)}} = \delta W^{n+1^{(k)}} + \Delta t A\delta W^{n+1^{(k)}} + \Delta t B Q^n \\ &\} \\ &\delta W^{n+1} = \delta W^{n+1^{(n_{S/F})}} \\ &Q^{n+1} = Q^{n+n_{S/F}} = Q^n + n_{S/F}\Delta t C\delta W^n + n_{S/F}\Delta t D(\frac{Q^n + Q^{n+n_{S/F}}}{2}) \end{aligned}} \tag{58}$$

Clearly, the fluid and structure kernels can run in parallel during the time-interval $[t_n,\ t_{n+n_{S/F}}]$. Inter-field communication or I/O transfer is needed only at the beginning of a time-interval. Expanding the various terms in Eqs. (58) around $t_n = n\Delta t$ leads to:

$$\boxed{\begin{aligned} ER_{\delta W}^{ALG3} &= O\left(n_{S/F}^2\Delta t^2\right) \\ ER_{Q}^{ALG3} &= O\left(n_{S/F}^2\Delta t^2\right) \end{aligned}} \tag{59}$$

which demonstrates that ALG3 is first-order accurate. However, the above error analysis also shows that parallelism in ALG3 is achieved at the expense of amplified errors in both the fluid (a factor equal to $n_{S/F}$ with respect to ALG2) and structural (a factor equal to $1/(n_{S/F}\Delta t)$ with respect to ALG2) systems.

The basic steps of ALG3 are graphically depicted in Fig. 14. The parallel CPU time needed to advanced the coupled solution $n_{S/F}\Delta t$ is equal to:

$$T_{coupled} = T_F + T_S + max(\frac{n_{S/F}O_F}{CPU_F}, \frac{O_S}{CPU_S}) \tag{60}$$

For a fixed total amount of CPU resources — for example, a fixed number of processors in a parallel machine — $T_{coupled}$ is minimum for:

$$\frac{CPU_F}{CPU_S} = \frac{n_{S/F}O_F}{O_S}$$

Hence, the parallel CPU time associated with ALG3 is:

$$\boxed{T^{ALG3}_{coupled} = T_F + T_S + \frac{n_{S/F}O_F + O_s}{CPU}} \tag{61}$$

which demonstrates the computational advantages of this parallel scheme.

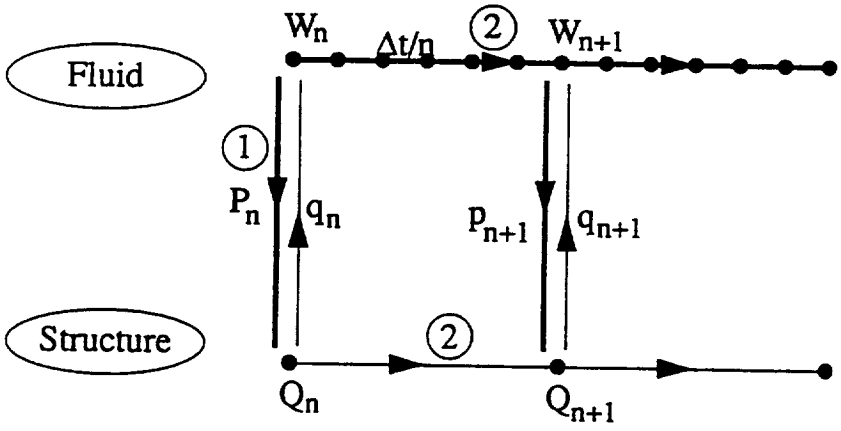

Fig. 14. The basic parallel subcycled ALG3 algorithm

6.5. Algorithm ALG4: improving the accuracy of ALG3

In order to improve the accuracy of the basic parallel time-integrator ALG3, we propose to exchange information between the fluid and structure kernels at half-step in the following specific manner (ALG4):

$$
\begin{aligned}
&\delta W^{n+1^{(0)}} = \delta W^n \\
&\{ \\
&\quad For\ k = 0,\ ...,\ \frac{n_{S/F}}{2} - 1 \\
&\quad \delta W^{n+1^{(k+1)}} = \delta W^{n+1^{(k)}} + \Delta t A \delta W^{n+1^{(k)}} + \Delta t B Q^n \\
&\} \\
&\delta W^{n+\frac{1}{2}} = \delta W^{n+1^{(\frac{n_{S/F}}{2})}} \\
&\overline{Q^{n+1}} = Q^n + n_{S/F}\Delta t C \delta W^n + n_{S/F}\Delta t D(\frac{Q^n + \overline{Q^{n+1}}}{2}) \\
&\} \\
&\quad For\ k = \frac{n_{S/F}}{2},\ ...,\ n_{S/F} - 1 \\
&\quad \delta W^{n+1^{(k+1)}} = \delta W^{n+1^{(k)}} + \Delta t A \delta W^{n+1^{(k)}} + \Delta t B \overline{Q^{n+1}} \\
&\} \\
&\delta W^{n+1} = \delta W^{n+1^{(n_{S/F})}} \\
&Q^{n+1} = Q^n + n_{S/F}\Delta t C \delta W^{n+\frac{1}{2}} + n_{S/F}\Delta t D(\frac{Q^n + Q^{n+1}}{2})
\end{aligned}
\tag{62}
$$

The above algorithm ALG4 is illustrated in Fig. 15. The first-half of the computations is identical to that of ALG3, except that the fluid system is subcycled only up to $t_{n+\frac{n_{S/F}}{2}}$, while the structure is advanced in one shot up to $t_{n+n_{S/F}}$. At $t_{n+\frac{n_{S/F}}{2}}$, the fluid and structure kernels exchange pressure, displacement and velocity information. In the second-half of the computations, the fluid system is subcycled from $t_{n+\frac{n_{S/F}}{2}}$ to $t_{n+n_{S/F}}$ using the new structural information, and the structural behavior is re-computed in parallel using the newly received pressure distribution. Note that the first evaluation of the structural state vector $\overline{Q^{n+1}}$ can be interpreted as a predictor.

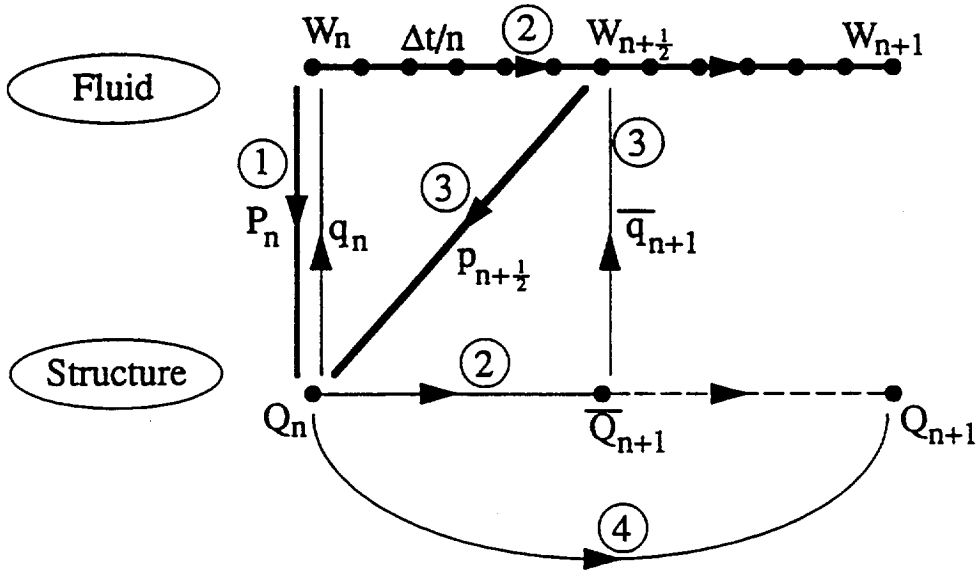

Fig. 15. The improved parallel subcycled ALG4 algorithm

An error analysis of ALG4 reveals that:

$$\boxed{\begin{array}{rcl} ER_{\delta W}^{ALG4} & = & O\left(n_{S/F}\Delta t^2\right) \\ ER_{Q}^{ALG4} & = & O\left(n_{S/F}^3\Delta t^3\right) \end{array}} \tag{63}$$

which shows that this parallel algorithm has the same accuracy as the improved ALG2.

The parallel CPU time needed to advanced the coupled solution $n_{S/F}\Delta t$ using ALG4 is equal to:

$$T_{coupled} = 2\left(T_F + T_S + max\left(\frac{\frac{n_{S/F}}{2}O_F}{CPU_F}, \frac{O_S}{CPU_S}\right)\right) \tag{64}$$

For a fixed total amount of CPU resources, this parallel time is minimum for:

$$\frac{CPU_F}{CPU_S} = \frac{n_{S/F}O_F}{2O_S}$$

Hence, the parallel CPU time corresponding to ALG4 is:

$$\boxed{T_{coupled}^{ALG4} = 2\left(T_F + T_S\right) + \frac{n_{S/F}O_F + 2O_s}{CPU}} \tag{65}$$

In summary, the accuracy of the basic parallel algorithm ALG3 is improved at the expense of an additional communication step or I/O transfer during each coupled cycle.

6.6. Applications

The advantages and limitations of ALG0—ALG4 are summarized in Fig. 16 which contrasts the various computed solutions $\bar{q}/l_0$ of the non-dimensional Eqs. (45), using the same non-dimensional parameters and finite volume mesh as in Section 4.3, and $\Delta = 0.9\times\text{CFL}$.

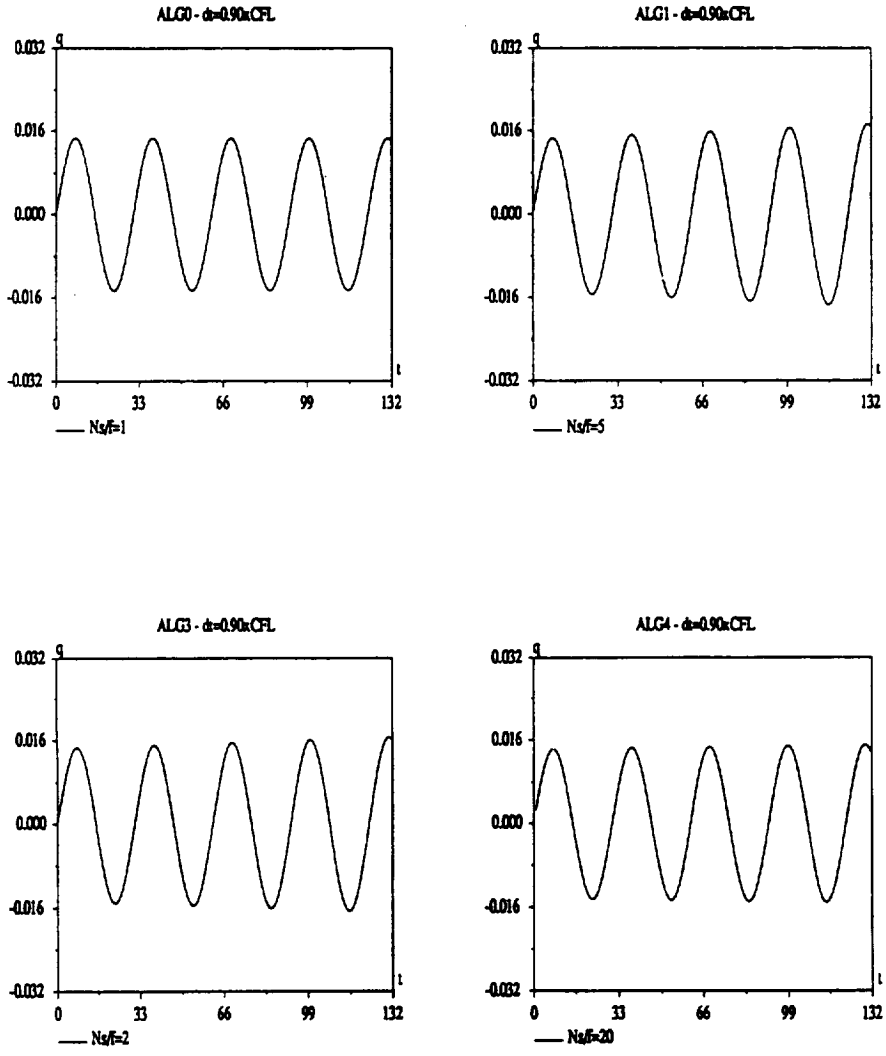

Fig. 16. Computed non-dimensional piston displacement (ALG0—ALG4)

If the ALG0 solution is used as reference, the reader can observe that ALG1 is at the stability limit when $n_{S/F} = 5$. The accuracy of ALG3 is comparable to that of ALG1, but its stability is restricted to $n_{S/F} = 2$. On the other hand, the parallel algorithm ALG4 is shown to have the same accuracy as ALG0 even for $n_{S/F} = 20$, which highlights the merits of this improved parallel algorithm.

7. Aeroelastic response of a flexible panel in transonic nonlinear regime

Based on the insight gained from the analysis and solution of the coupled piston problem, we have extended the algorithms presented in this paper to the solution of the three-field coupled formulations summarized in Eqs. (1), and complex aeroelastic problems. The generalization of ALG0—ALG4, their implementation on heterogeneous parallel processors, and the analysis of their performance results are discussed in details in a companion paper [22]. Here, we focus on validating qualitatively the conclusions drawn from the mathematical and numerical investigations of the model problem with the simulation of the two-dimensional transient aeroelastic response of a flexible panel in transonic nonlinear regime.

For a two-dimensional simulation, the panel is represented by its cross section that is assumed to have a unit length $L = 1$, a uniform thickness $h = 10^{-2} \times L$, and to be clamped at both ends. This rectangular cross section is discretized into 300×3 plane strain 4-node elements with perfect aspect ratio to avoid mesh locking. This fine discretization — which generates 1204 nodes — is not needed for accuracy; we have designed this structural mesh only because we were also interested in assessing some computational and I/O performance issues. The two-dimensional flow domain around the panel is discretized into 2880 triangles and 1504 vertices. The free stream Mach number is set to $M_\infty = 0.8$, and a slip condition is imposed at the fluid/structure boundary. Because the fluid and structural meshes are not compatible at their interfaces (Fig. 17), the "Matcher" software [23] is used to transfer the pressure load to the structure, and to transmit the structural deformations at the surfaces of the panel to the fluid.

Initially, a steady-state flow is computed around the panel at $M_\infty = 0.8$ (Fig. 18). Next, this flow is perturbed via an initial displacement of the panel that is proportional to its second fundamental mode (Fig. 19), and the subsequent panel motion and flow evolution are computed using the ALG0—ALG4 explicit/implicit fluid/structure time-integrators (Fig. 20-21).

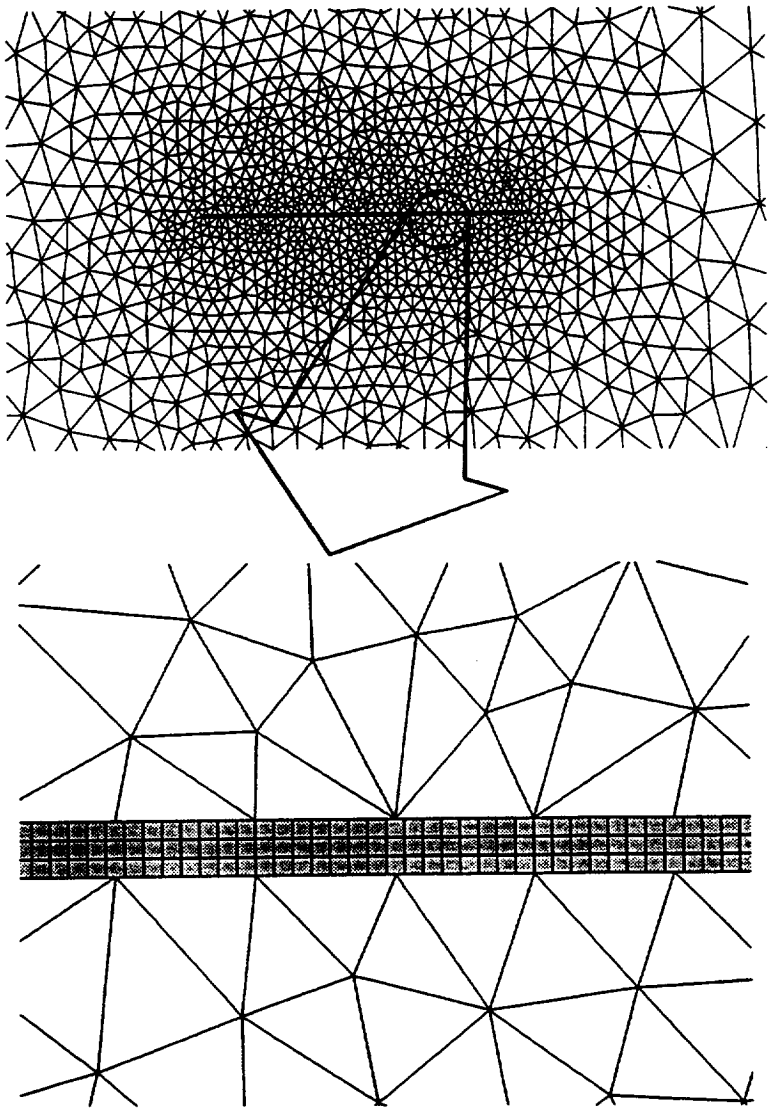

Fig. 17. A partial view of the structure and fluid discretizations

More specifically, the dynamic equations of equilibrium of the structure are solved via the parallel implicit transient FETI method [24], with the improvements proposed in [25] for the efficient iterative solution of systems with repeated right hand sides. The Euler flow equations are solved with a parallel algorithm

that combines a second-order accurate Monotonic Upwinding Scheme for Conservation Laws for spatial approximation, and a second-order low-storage explicit Runge-Kutta scheme for time-integration [26]. All computations are carried out on an iPSC-860 parallel processor. Four processors are allocated to the fluid code, and two processors to the structural program. The fluid and structural computations are implemented in a heterogeneous manner using the intercube communication procedures described in [27].

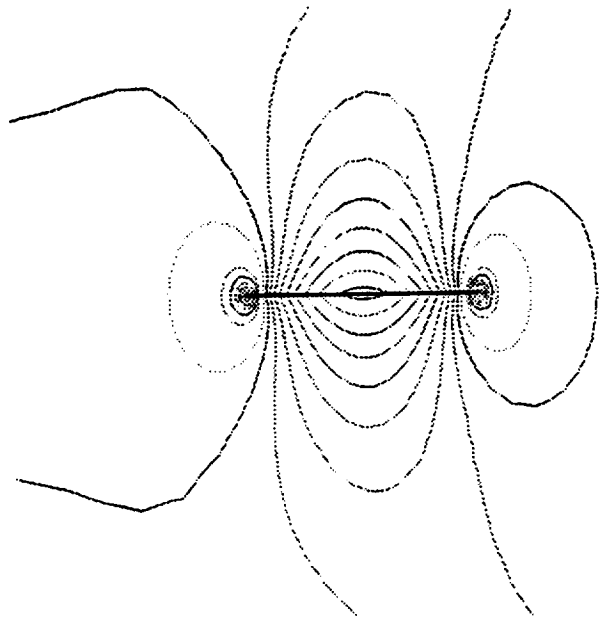

Fig. 18. Pressure isovalues of the steady-state flow solution

Fig. 19. Initial perturbation of the panel displacement field

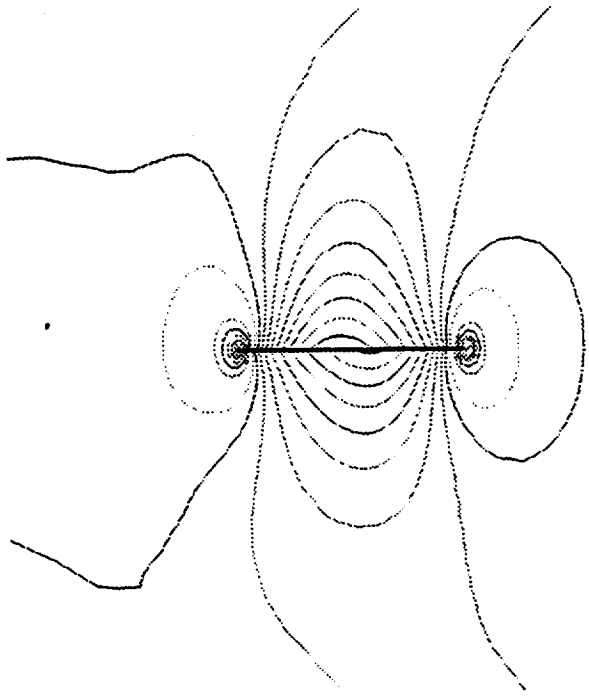

Fig. 20. Pressure isovalues at $t = 8.75 \times 10^{-5}$

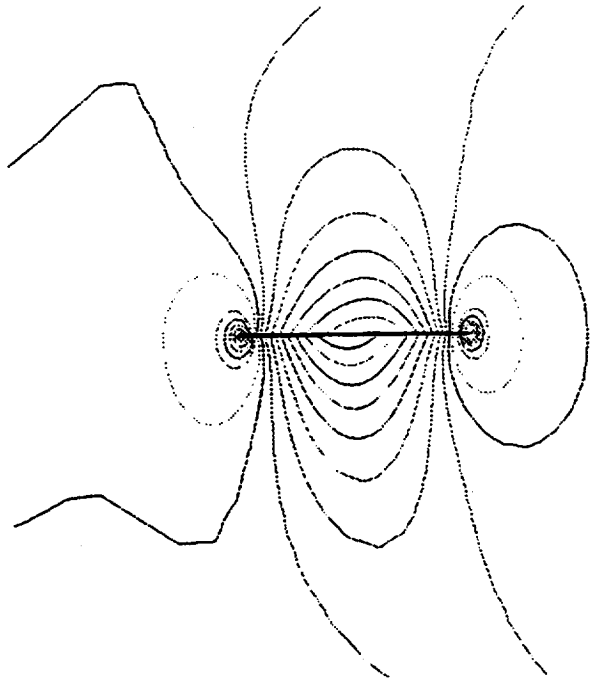

Fig. 21. Pressure isovalues at $t = 1.25 \times 10^{-4}$

For the uncoupled fluid problem, the CFL stability time-step is $\Delta t = 1\times\text{CFL} = 6.25\times 10^{-6}$. The lift solutions computed by ALG0 ($\Delta t_c = \Delta t_F = \Delta t_S = 6.25\times 10^{-6}$), ALG1 ($\Delta t_F = 6.25\times 10^{-6}$, $\Delta t_S = 5\times 10^{-5}$, $n_{S/F} = 8$), ALG3 ($\Delta t_F = 6.25\times 10^{-6}$, $\Delta t_S = 5\times 10^{-5}$, $n_{S/F} = 8$), and ALG4 ($\Delta t_F = 6.25\times 10^{-6}$, $\Delta t_S = 5\times 10^{-5}$, $n_{S/F} = 8$) are depicted in Fig. 22. Clearly, as predicted by the theory presented in this paper, all proposed explicit/implicit time-integrators are shown to be numerically stable at $\Delta t_F = 1\times\text{CFL}$ and $n_{S/F} = 8$, and ALG4 is reported to improve the accuracy of ALG3. Note also that the parallel algorithm ALG4 is shown to achieve in practice a better accuracy than the sequential algorithm ALG1 for the same time-steps $\Delta t_F = 1\times\text{CFL}$ and $\Delta t_S = 8\times$ CFL.

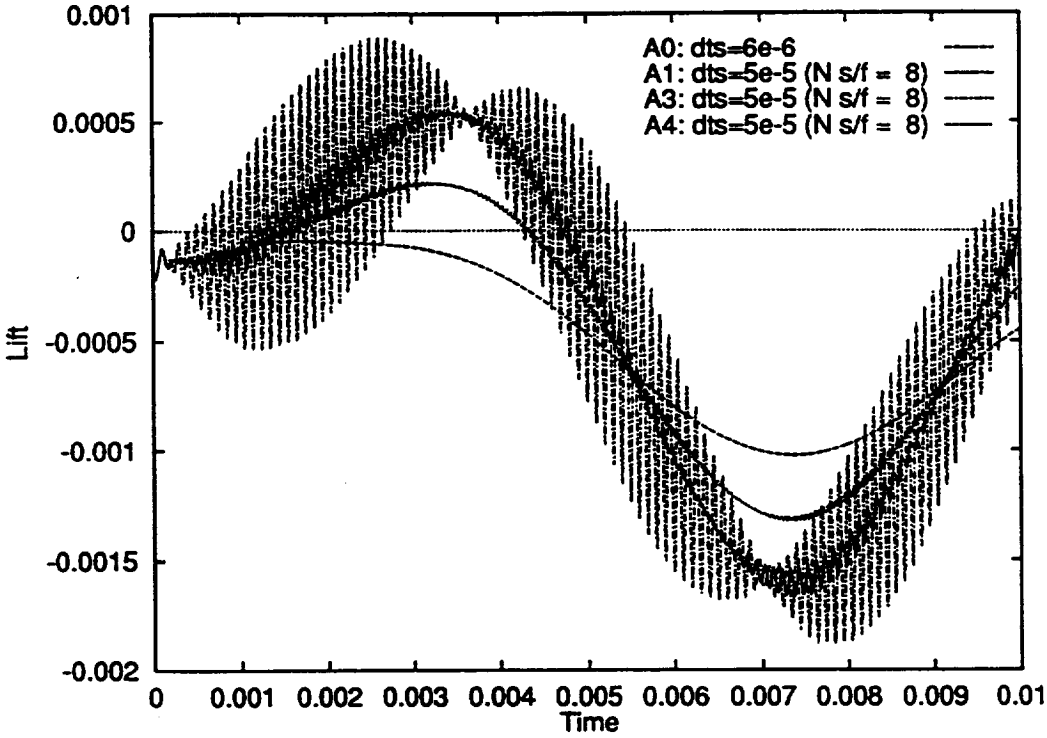

Fig. 22. Computed lifts (ALG0, ALG1, ALG3, ALG4)

The superiority of ALG4 over ALG3 is also illustrated in Fig. 23 which shows that the lift solution computed by ALG4 with $\Delta t_S = 16\times$ CFL is less oscillating than that computed by ALG3 with a smaller $\Delta t_S = 8\times$ CFL. While both computed lift solutions may not be accurate enough for structural analysis purposes, ALG3 and ALG4 quickly (faster than ALG0) and correctly reproduce the overall aeroelastic behavior of the system — for example, they show that flutter is not occuring — which is what a designer is mostly interested in verifying initially.

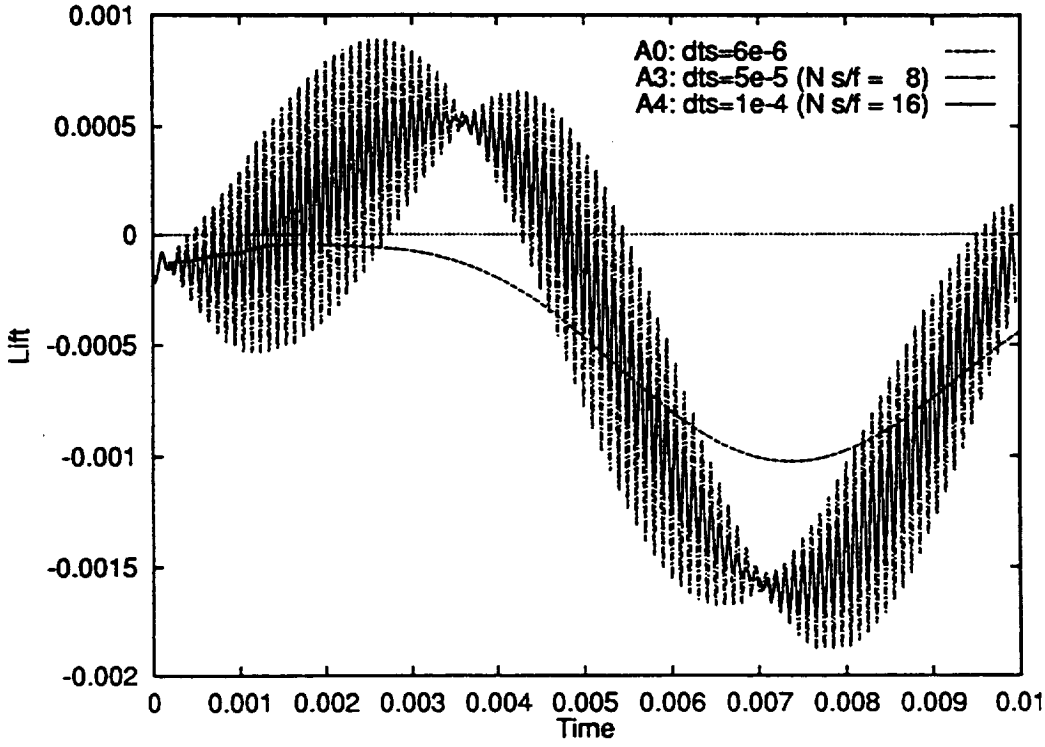

Fig. 23. Computed lifts (ALG0, ALG3, ALG4)

8. Closure

In this paper, we have presented several partitioned procedures for time-integrating the transient coupled aeroelastic problem, and have discussed their merits in terms of accuracy, stability, heterogeneous computing, I/O transfers, subcycling, and parallel processing. All theoretical results have been derived for a one-dimensional piston model problem with a compressible flow, because the complete three-dimensional aeroelastic problem is difficult to analyze mathematically. However, the insight gained from the analysis of the coupled piston problem and the conclusions drawn from its numerical investigation have been confirmed with the numerical simulation of the two-dimensional transient aeroelastic response of a flexible panel in a transonic nonlinear Euler flow regime. In particular, we hope that with the unconditionally stable implicit-implicit staggered procedure and the parallel coupling strategy with superior accuracy properties presented in this paper, large-scale transient aeroelastic computations will be finally feasible.

Acknowledgments

The second author acknowledges partial support by NASA Lewis Research Center under Grant NAG3-1425, and partial support by the National Science Foundation under Grant ASC-9217394. The authors would like to thank P.-S. Chen, L. A. Crivelli, S. Lanteri, and N. Maman for the parallel implementations on the iPSC-860.

References

[1] J. Donea, An arbitrary Lagrangian-Eulerian finite element method for transient fluid-structure interactions, Comput. Meths. Appl. Mech. Engrg. 33 (1982) 689–723.

[2] T. J. R. Hughes, W. K. Liu and T. K. Zimmermann, Lagrangian-Eulerian finite element formulation for incompressible viscous flows, U.S.-Japan Seminar on Interdisciplinary Finite Element Analysis, Cornell Univ., Ithaca, NY, Aug. 7–11 (1978).

[3] T. Belytschko and J. M. Kennedy, Computer models for subassembly simulation, Nucl. Eng. Design 49 (1978) 17–38.

[4] O. A. Kandil and H. A. Chuang, Unsteady vortex-dominated flows around maneuvering wings over a wide range of mach numbers, AIAA Paper No. 88-0317, AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, January 11–14, 1988.

[5] C. Farhat and T. Y. Lin, Transient aeroelastic computations using multiple moving frames of reference, AIAA Paper No. 90-3053, AIAA 8th Applied Aerodynamics Conference, Portland, Oregon, August 20–22, 1990.

[6] J. T. Batina, Unsteady Euler airfoil solutions using unstructured dynamic meshes, AIAA Paper No. 89-0115, AIAA 27th Aerospace Sciences Meeting, Reno, Nevada, January 9–12, 1989.

[7] M. Lesoinne and C. Farhat, Stability analysis of dynamic meshes for transient aeroelastic computations, AIAA Paper No. 93-3325, 11th AIAA Computational Fluid Dynamics Conference, Orlando, Florida, July 6–9, 1993.

[8] K. C. Park and C. A. Felippa, Partitioned analysis of coupled systems, in: Computational Methods for Transient Analysis, T. Belytschko and T. J. R. Hughes, Eds., North-Holland Pub. Co. (1983) 157–219.

[9] C. J. Borland and D. P. Rizzetta, Nonlinear transonic flutter analysis, AIAA Journal (1982) 1606–1615.

[10] V. Shankar and H. Ide, Aeroelastic computations of flexible configurations, Computers & Structures 30 (1988) 15–28.

[11] R. D. Rausch, J. T. Batina and T. Y. Yang, Euler flutter analysis of airfoils using unstructured dynamic meshes, AIAA Paper No. 89-13834, 30th Structures, Structural Dynamics and Materials Conference, Mobile, Alabama, April 3–5, 1989.

[12] M. Blair, M. H. Williams and T. A. Weisshaar, Time domain simulations of a flexible wing in subsonic compressible flow, AIAA Paper No. 90-1153, AIAA 8th Applied Aerodynamics Conference, Portland, Oregon, August 20–22, 1990.

[13] T. W. Strganac and D. T. Mook, Numerical model of unsteady subsonic aeroelastic behavior, AIAA Journal 28 (1990) 903–909.

[14] E. Pramono and S. K. Weeratunga, Aeroelastic computations for wings through direct coupling on distributed-memory MIMD parallel computers, AIAA Paper No. 94-0095, 32nd Aerospace Sciences Meeting & Exhibit, Reno, January 10-13, 1994.

[15] T. J. R. Hughes and W. K. Liu, Implicit-explicit finite elements in transient analysis: stability theory, ASME Journal of Applied Mechanics 45 (1978) 371–374.

[16] T. Belytschko, P. Smolenski and W. K. Liu, Stability of multi-time step partitioned integrators for first-order finite element systems, Comput. Meths. Appl. Mech. Engrg. 49 (1985) 281–297.

[17] Y. C. Fung, An introduction to the theory of aeroelasticity, Dover Publications, Inc., New York (1969) 160–185.

[18] H. Schlichting, Boundary layer theory, Fourth edition, McGraw-Hill, New York, 1960.

[19] T. J. R. Hughes, Analysis of transient algorithms with particular reference to stability behavior, T. Belytschko and T. J. R. Hughes, Eds., North-Holland Pub. Co. (1983) 67–155.

[20] C. Farhat, K. C. Park and Y. D. Pelerin, An unconditionally stable staggered algorithm for transient finite element analysis of coupled thermoelastic problems, Comput. Meths. Appl. Mech. Engrg. 85 (1991) 349–365.

[21] C. Farhat and N. Sobh, A consistency analysis of a class of concurrent transient implicit/explicit algorithms, Comput. Meths. Appl. Mech. Engrg. 84 (1990) 147–162.

[22] C. Farhat, S. Lanteri, N. Maman, and P.S. Chen, Parallel and heterogeneous solution of three-dimensional transient coupled aeroelastic problems," *(submitted for publication)*

[23] N. Maman and C. Farhat, Matching fluid and structure meshes for aeroelastic computations: a parallel approach, Report CU-CSSC-93-12, The University of Colorado at Boulder, June 1993.

[24] L. Crivelli and C. Farhat, Implicit transient finite element structural computations on MIMD systems: FETI v.s. direct solvers, AIAA Paper No. 93-1310, AIAA 34th Structural Dynamics Meeting, La Jolla, California, April 19-21, 1993.

[25] C. Farhat, L. Crivelli and F. X. Roux, Extending substructure based iterative solvers to multiple load and repeated analyses, Comput. Meths. Appl. Mech. Engrg. *(in press)*

[26] 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. Meths. Appl. Mech. Engrg. 102 (1991) 61–88.

[27] E. Barszcz, Intercube communication on the iPSC/860, Scalable High Performance Computing Conference, Williamsburg, April 26-29, 1992.

[28] S. Piperno, Numerical methods used in aeroelasticity simulations, Rapport CERMICS 92-5, Oct. 1992.

APPENDIX II

Simulation of Compressible Viscous Flows on a Variety of MPPs: Computational Algorithms for Unstructured Dynamic Meshes and Performance Results

by

C. FARHAT AND S. LANTERI

Department of Aerospace Engineering Sciences and
Center for Space Structures and Controls
University of Colorado
Boulder, Colorado 80309-0429

January 1994

Abstract

Here we report on our effort in simulating unsteady viscous flows on the iPSC-860, the CM-5, and the KSR-1 MPPs (Massively Parallel Processors), using a Monotonic Upwind Scheme for Conservation Laws finite volume/finite element method on fully unstructured fixed and moving grids. We advocate mesh partitioning with message passing as a portable paradigm for parallel processing. We present and discuss several performance results obtained on all three MPP systems in terms of interprocessor communication costs, I/O, scalability, and sheer performance.

Contents

1 Introduction

In this paper, we detail our approach to the simulation of large scale, steady and unsteady, compressible viscous flows on massively parallel processors. We consider the numerical solution of the two-dimensional Navier-Stokes equations using a mixed finite element/finite volume formulation based on unstructured triangular meshes. The spatial approximation method combines a Galerkin centered approximation for the viscous terms, and a Roe upwind scheme for the computation of the convective fluxes. Higher order accuracy is achieved through the use of a piecewise linear interpolation method that follows the principle of the MUSCL (Monotonic Upwind Scheme for Conservative Laws) procedure. The temporal solution is carried out via a 3-step variant of the explicit Runge-Kutta method which lends itself to parallel processing. An ALE (Arbitrary Lagrangian Eulerian) formulation is incorporated in the fluid solver to allow the grid points to displace in Lagrangian fashion, or be held fixed in Eulerian manner, or be moved in some specified way to give a continuous and automatic rezoning capability, depending on the needs of the physical problem to be solved.

Explicit solvers are naturally amenable to parallel processing because they essentially involve local computations on vertices, and/or edges, and/or triangles of a mesh. However, unstructured meshes induce indirect addressing memory operations that are costly on many hardware architectures. In particular, the present mixed finite element/finite volume solver incurs multiple gather/scatter operations between vertex and triangle based arrays. Therefore, in this paper we highlight the impact of irregular data access patterns on the computational scalability of a parallel unstructured solver, and emphasize the importance of data locality in achieving high level performances. These concerns and our drive for developing a portable code have led us to adopt mesh partitioning with message-passing as a paradigm for parallel processing.

The remainder of this paper is organized as follows. Section 2 describes the mathematical model of the problem and the approximation methods involved in the numerical solution algorithm. Section 3 identifies the main computational kernels, and motivates the selected

parallelization strategy. Overlapping and non-overlapping mesh partitions are presented, discussed, and contrasted. Finally, Sections 4 and 5 report and analyze the performance results obtained on the iPSC-860, the KSR-1, and the CM-5 parallel processors for various external and internal viscous flow simulations, with fixed and moving meshes.

2 Simulation of compressible viscous flows

We are interested in the numerical simulation of two-dimensional compressible viscous flows around or within, fixed, or moving and deforming bodies. Here, we overview the spatial and temporal discretization methods that have been previously detailed in Farhat, Fezoui and Lanteri [3] for fixed meshes, and outline the mesh updating procedure adopted for aeroelastic computations.

2.1 Governing equations

Let $\Omega \subset \mathbb{R}^2$ be the flow domain of interest and Γ be its boundary. The conservative law form of the equations describing two-dimensional Navier-Stokes flows is given by :

$$\frac{\partial}{\partial t} W(\vec{x},t) + \vec{\nabla}.\vec{\mathcal{F}}(W(\vec{x},t)) = \frac{1}{Re}\vec{\nabla}.\vec{\mathcal{R}}(W(\vec{x},t)) \qquad (1)$$

where $\vec{x}$ and t denote the spatial and temporal variables, and

$$W = (\rho\ ,\ \rho u\ ,\ \rho v\ ,\ E)^T \quad , \quad \vec{\nabla} = \left(\frac{\partial}{\partial x}\ ,\ \frac{\partial}{\partial y}\right)^T$$

and

$$\vec{\mathcal{F}}(W) = \begin{pmatrix} F(W) \\ G(W) \end{pmatrix} \quad , \quad \vec{\mathcal{R}}(W) = \begin{pmatrix} R(W) \\ S(W) \end{pmatrix}$$

$F(W)$ and $G(W)$ denote the convective fluxes and are given by:

$$F(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix} \quad , \quad G(W) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}$$

while $R(W)$ and $S(W)$ denote the diffusive fluxes and are given by:

$$R(W) = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} + \frac{\gamma k}{Pr}\frac{\partial \varepsilon}{\partial x} \end{pmatrix} \quad , \quad S(W) = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} + \frac{\gamma k}{Pr}\frac{\partial \varepsilon}{\partial y} \end{pmatrix}$$

In the above expressions, ρ is the density, $\vec{U} = (u, v)$ is the velocity vector, E is the total energy per unit of volume, p is the pressure, ε is the specific internal energy, τ_{xx}, τ_{xy}, and τ_{yy} are the components of the two-dimensional Cauchy stress tensor, k is the normalized thermal conductivity, $Re = \frac{\rho_0 U_0 L_0}{\mu_0}$ where ρ_0, U_0, L_0 and μ_0 denote the characteristic density, velocity, length, and diffusivity is the Reynolds number, and $Pr = \frac{\mu_0 C_p}{k_0}$ is the Prandtl number.

The velocity, energy, and pressure are related by the equation of state for a perfect gas:

$$p = (\gamma - 1)(E - \frac{1}{2}\rho \parallel \vec{U} \parallel^2)$$

where γ is the ratio of specific heats ($\gamma = 1.4$ for air), and the specific internal energy is related to the temperature via:

$$\varepsilon = C_v T = \frac{E}{\rho} - \frac{1}{2} \parallel \vec{U} \parallel^2$$

The components of the Cauchy stress tensor are related to the velocities via:

$$\tau_{xx} = \frac{2}{3}\mu\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right) \ , \ \tau_{yy} = \frac{2}{3}\mu\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right) \ , \ \tau_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)$$

where μ denotes the normalized viscosity.

In order to account for a potential motion or deformation of the computational grid, the following coordinate transformation is introduced:

$$\begin{cases} \vec{\xi} &= \vec{\xi}(\vec{x}, t) \\ \tau &= t \end{cases} \tag{2}$$

Throughout this paper, it is assumed that the Jacobian of the above transformation defined as:

$$J = \det\left(\frac{\partial \vec{x}}{\partial \vec{\xi}}|_t\right)$$

does not vanish at any point and any time. Introducing the grid velocity:

$$\vec{w} = \frac{\partial \vec{x}}{\partial t}|_{\vec{\xi}}$$

and using Eqs. (2), Eq. (1) can be transformed into the following ALE (Arbitrary Lagrangian Eulerian) formulation (see, for example, Donea [2]):

$$\frac{\partial (JW)}{\partial t}|_{\vec{\xi}} + J\vec{\nabla}.\vec{\mathcal{F}}_c(W) = \frac{1}{Re}\vec{\nabla}.\vec{\mathcal{R}}(W) \tag{3}$$

where :

$$\vec{\mathcal{F}}_c(W) = \begin{pmatrix} F_c(W) \\ G_c(W) \end{pmatrix}$$

and $F_c(W)$ and $G_c(W)$ are the modified convective fluxes given by:

$$F_c(W) = \begin{pmatrix} \rho\bar{u} \\ \rho u\bar{u} + p \\ \rho\bar{u}v \\ E\bar{u} + up \end{pmatrix} \quad , \quad G_c(W) = \begin{pmatrix} \rho\bar{v} \\ \rho u\bar{v} \\ \rho v\bar{v} + p \\ E\bar{v} + pv \end{pmatrix}$$

and:

$$\begin{cases} \bar{u} &= u - w_x \\ \bar{v} &= v - w_y \end{cases} \tag{4}$$

2.2 Boundary conditions

The boundary $\Gamma(t)$ of the flow domain is partitioned into a wall boundary $\Gamma_w(t)$ and an infinity boundary $\Gamma_\infty(t)$: $\Gamma(t) = \Gamma_w(t) \cup \Gamma_\infty(t)$. Let $\vec{n}(t)$ denote the outward unit normal at any point of $\Gamma(t)$, and $\vec{U}_w$ and T_w denote the wall velocity and temperature.

On the wall boundary $\Gamma_w(t)$, a no-slip condition and a Dirichlet condition on the temperature are imposed:

$$\vec{U} = \vec{U}_w \quad , \quad T = T_w \tag{5}$$

No boundary condition is specified for the density. Hence, the total energy per unit of volume and the pressure on the wall are given by :

$$p = (\gamma - 1)\rho C_v T_w \ , \quad E = \rho C_v T_w + \frac{1}{2}\rho \parallel \vec{U}_w \parallel^2 \tag{6}$$

For external flows around airfoils, the viscous effects are assumed to be negligible at infinity, so that a uniform free-stream state vector W_∞ is imposed on $\Gamma_\infty(t)$:

$$\rho_\infty = 1 \ , \quad \vec{U}_\infty = \begin{pmatrix} cos\alpha \\ sin\alpha \end{pmatrix} \ , \quad p_\infty = \frac{1}{\gamma M_\infty^2} \tag{7}$$

where α is the angle of attack, and M_∞ is the free-stream Mach number.

For internal flows, $\Gamma_\infty(t)$ is partitioned into upstream and downstream boundaries which are in general in contact with the wall boundary: $\Gamma_\infty(t) = \Gamma_\infty^{in}(t) \cup \Gamma_\infty^{out}(t)$. In that case, the previous definition of W_∞ is improved using a parabolic profile for the free-stream velocity. For example, for the horizontal flow between two plates shown in Figure 1, one can specify:

$$\vec{U}_\infty = (0 \ , \ v_\infty(y))^T \tag{8}$$

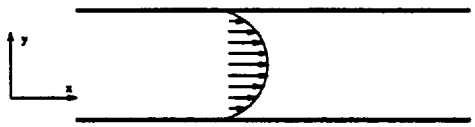

Figure 1: Horizontal velocity profile for internal flows

2.3 Spatial discretization

The flow domain $\Omega(t)$ is assumed to be a polygonal bounded region of $\mathbb{R}^2$. Let $\mathcal{T}_h$ be a standard triangulation of $\Omega(t)$, and h the maximal length of the edges of $\mathcal{T}_h$. A vertex of a triangle T is denoted by S_i, and the set of its neighboring vertices by $K(i)$. At each vertex S_i, a cell $C_i(t)$ 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 (see Figure 2). The boundary of $C_i(t)$ is denoted by $\partial C_i(t)$, and the unit vector of the outward normal to $\partial C_i(t)$ by $\vec{\nu}_i(t) = (\nu_{ix}(t), \nu_{iy}(t))$. The union of all these control volumes constitutes a discretization of domain $\Omega(t)$:

$$\Omega_h(t) = \bigcup_{i=1}^{ns} C_i(t)$$

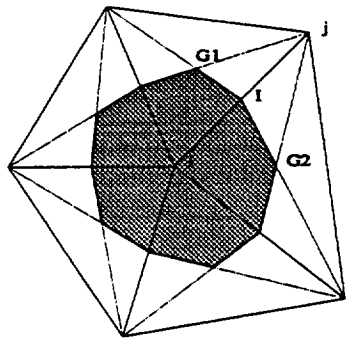

Figure 2: Control volume in an unstructured grid

The spatial discretization method adopted here combines the following features (see [3] for details):

- a finite volume upwind approximation method for the convective fluxes. Second order spatial accuracy is achieved using an extension of Van Leer's MUSCL technique [13] to unstructured meshes;
- a classical Galerkin finite element centered approximation for the diffusive fluxes.

Let $C_i^{\vec{x}}$ and $C_i^{\vec{\xi}}$ denote the two representations of $C_i(t)$ in the coordinate systems defined by $\vec{x}$ and $\vec{\xi}$, respectively. By definition, $C_i^{\vec{\xi}}$ is a fixed reference representation of $C_i(t)$. Integrating Eq. (3) over $C_i^{\vec{\xi}}$ yields:

$$\int\int_{C_i^{\vec{\xi}}} \frac{\partial(JW)}{\partial t}|_{\vec{\xi}}\, d\vec{\xi} + \int\int_{C_i^{\vec{\xi}}} J\vec{\nabla}.\vec{\mathcal{F}}_c(W)d\vec{\xi} = \int\int_{C_i^{\vec{\xi}}} \frac{1}{Re}\vec{\nabla}.\vec{\mathcal{R}}(W) \quad (9)$$

Given that the time derivative is computed for a constant $\vec{\xi}$ and that $C_i^{\vec{\xi}}$ does not depend on the time t, the mapping $\vec{\xi} = \vec{\xi}(\vec{x}, t)$ and the identity $d\vec{x} = Jd\vec{\xi}$ can be used to transform the left-hand side of Eq. (9) into:

$$\frac{d}{dt}\int\int_{C_i^{\vec{x}}} W d\vec{x} + \int\int_{C_i^{\vec{x}}} \vec{\nabla}.\vec{\mathcal{F}}_c(W)d\vec{x} = \int\int_{C_i^{\vec{x}}} \frac{1}{Re}\vec{\nabla}.\vec{\mathcal{R}}(W) \quad (10)$$

Finally, integrating Eq. (10) by parts leads to:

$$\begin{aligned} \frac{d}{dt}\int\int_{C_i(t)} W d\vec{x} \;+\; & \sum_{j\in K(i)} \int_{\partial C_{ij}^{\vec{x}}(t)} \vec{\mathcal{F}}_c(W).\vec{\nu}_i d\sigma && < 1 > \\ + & \int_{\partial C_i(t)\cap\Gamma_w(t)} \vec{\mathcal{F}}_c(W).\vec{n}_i d\sigma && < 2 > \\ + & \int_{\partial C_i(t)\cap\Gamma_\infty(t)} \vec{\mathcal{F}}_c(W).\vec{n}_i d\sigma && < 3 > \\ = \; & -\frac{1}{Re}\sum_{T, S_i\in T}\int\int_T \vec{\mathcal{R}}(W).\vec{\nabla}N_i^T d\vec{x} && < 4 > \end{aligned} \quad (11)$$

where $\partial C_{ij}^{\vec{x}}(t) = \partial C_i^{\vec{x}}(t)\cap\partial C_j^{\vec{x}}(t)$, and $N_i^T = N_i^T(x, y)$ is the P1 shape function defined at the vertex S_i and associated with the triangle T.

A first order finite volume discretization of $< 1 >$ goes as follows:

$$< 1 > = A_i^{n+1}W_i^{n+1} - A_i^n W_i^n + \Delta t \sum_{j\in K(i)} \Phi_{\mathcal{F}_c}(W_i^n, W_j^n, \tilde{\vec{\nu}}_{ij}) \quad (12)$$

where A_i^n denotes the area of the control volume $C_i^{\vec{x}}$ measured at time t^n, $\vec{\nu}_{ij}(t)$ denotes a spatial mean value the normal to $\partial C_{ij}(t)$, the tilde superscript designates a temporal mean value between t^n and

t^{n+1}, and $\Phi_{\mathcal{F}_c}$ denotes a numerical flux function that approximates the following quantity:

$$\Delta t \Phi_{\mathcal{F}_c}(W_i, W_j, \vec{\nu}_{ij}) \approx \int_{t^n}^{t^{n+1}} \int_{\partial C_{ij}^{\sharp}} \vec{\mathcal{F}}_c(W).\vec{\nu}_i d\sigma \tag{13}$$

Upwinding is introduced by extending Roe's approximate Riemann solver [11] to dynamic meshes and computing $\Phi_{\mathcal{F}_c}$ as follows:

$$\begin{aligned} \Phi_{\mathcal{F}_c}(W_i, W_j, \vec{\nu}_{ij}) &= \frac{\vec{\mathcal{F}}_c(W_i) + \vec{\mathcal{F}}_c(W_j)}{2}.\vec{\nu}_{ij} \\ &- \mid \mathcal{A}(W_i, W_j, \vec{\nu}_{ij}) - (\vec{w}.\vec{\nu}_{ij})I \mid \frac{(W_j - W_i)}{2} \end{aligned} \tag{14}$$

where $\mathcal{A}$ is Roe's mean value of the flux Jacobian matrix $\dfrac{\partial \vec{F}(W)}{\partial W}$, I is the identity matrix, and the dot product $\vec{w}.\vec{\nu}_{ij}$ is computed as suggested recently by N'Konga and Guillard [10]:

$$\vec{w}.\vec{\nu}_{ij} = \frac{1}{2}\left(\vec{w}(P_{1ij}) + \vec{w}(P_{2ij})\right).\vec{\nu}_{ij} \tag{15}$$

where P_{1ij} and P_{2ij} are the end points of the bi-segment ∂C_{ij} (see 3).

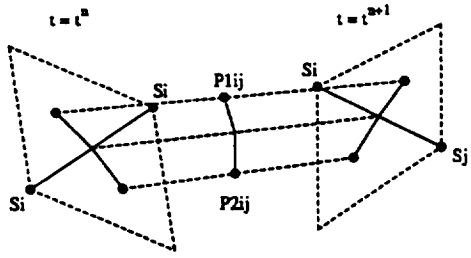

Figure 3: Computation of $\vec{w}.\vec{\nu}_{ij}$

The control volume area A_i^{n+1} is updated using the following finite volume scheme:

$$A_i^{n+1} = A_i^n + \Delta t . \sum_{j \in K(i)} \vec{w}.\vec{\nu}_{ij} \tag{16}$$

Following the MUSCL technique, second order accuracy is achieved in expression (13) via a piecewise linear interpolation of the states W_{ij} and W_{ji} at the interface between cells C_i and C_j. This requires the evaluation of the gradient of the solution at each vertex as follows:

$$\begin{cases} W_{ij}^{\star} &= W_i^{\star} + \frac{1}{2}(\vec{\nabla}W)_i^{\beta}.S_i\vec{S}_j \\ W_{ji}^{\star} &= W_j^{\star} - \frac{1}{2}(\vec{\nabla}W)_j^{\beta}.S_i\vec{S}_j \end{cases} \tag{17}$$

where $W^{\star} = (\rho\ ,\ u\ ,\ v\ ,\ p)^T$ — in other words, the interpolation is performed on the physical variables instead of the conservative variables. The approximate nodal gradients $(\vec{\nabla}W)_{i,j}^{\beta}$ are obtained using a β-combination of centered and fully upwind gradients :

$$(\nabla\vec{W})_i^{\beta} = (1-\beta)(\nabla\vec{W})_i^{Cent} + \beta(\nabla\vec{W})_i^{Upw} \tag{18}$$

The half-upwind scheme ($\beta = \frac{1}{2}$) is simply obtained by means of a linear interpolation of the Galerkin gradients computed on each triangle of C_i:

$$\begin{aligned} \left(\nabla\vec{W}\right)_i^{\beta=\frac{1}{2}} &= \frac{\iint_{C_i} \nabla\vec{W}|_T d\vec{x}}{\iint_{C_i} d\vec{x}} \\ &= \frac{1}{area(C_i)} \sum_{T\epsilon C_i} \frac{area(T)}{3} \sum_{k=1,k\epsilon T}^{3} W_k^{\star}\vec{\nabla}N_k^T \end{aligned} \tag{19}$$

and the centered gradient $(\nabla\vec{W})_i^{Cent}$ ($\beta = 0$) is given by any vector that verifies:

$$(\nabla\vec{W})_i^{Cent}.S_i\vec{S}_j = W_j - W_i \tag{20}$$

The second term $< 2 >$ and the third term $< 3 >$ of Eq. (11) include the contributions of the boundary conditions and are evaluated as follows:

Wall boundary : the no-slip condition is enforced with a strong formulation and therefore the corresponding boundary integral in $< 2 >$ is not explicitly computed.

Inflow and outflow boundaries : at these boundaries, a precise set of compatible exterior data that depend on the flow regime and the velocity direction must be specified. Here, a *plus-minus* flux splitting is applied between exterior data and interior values. More specifically, the boundary integral $< 3 >$ is evaluated using a non-reflective version of the flux-splitting of Steger and Warming [12]:

$$\int_{\partial C_i(t) \cap \Gamma_\infty} \mathcal{F}(W).\vec{n}_i d\sigma = \mathcal{A}^+(W_i, \vec{v}_{i\infty}).W_i + \mathcal{A}^-(W_i, \vec{v}_{i\infty}).W_\infty \quad (21)$$

Finally, the viscous integral $< 4 >$ is evaluated using a classical Galerkin finite element P1 method. The components of the stress tensor and those of ∇N_i^T are constant in each triangle. The velocity vector in a triangle is computed as follows:

$$U_T = \frac{1}{3} \sum_{k=1, k \epsilon T}^{3} U^k$$

Consequently, the viscous fluxes are approximated as follows:

$$\sum_{T, S_i \in T} \int_T \vec{\mathcal{R}}(W).\nabla N_i^T d\vec{x} = \sum_{T, S_i \in T} area(T) \left(R_T \frac{\partial N_i^T}{\partial x} + S_T \frac{\partial N_i^T}{\partial y} \right)$$

where R_T and S_T are the constant values of $R(W)$ and $S(W)$ in the triangle T.

2.4 Time integration

The resulting semi-discrete fluid flow equations can be written as:

$$\frac{dW}{dt} + \psi(W) = 0$$

Because it lends itself to massive parallelism, the following 3-step variant of the explicit Runge-Kutta algorithm is selected for time integrating the above equations:

$$\begin{cases} W^{(0)} &= W^n = W(t = n\Delta t) \\ W^{(k)} &= \frac{A_i^n}{A_i^{n+1}} W^{(0)} + \frac{1}{A_i^{n+1}} \alpha_k \Delta t \Psi(W^{(k-1)}) \qquad k = 1,2,3 \\ W^{(3)} &= W^{n+1} \end{cases} \tag{22}$$

This scheme is often referred to as a low-storage Runge-Kutta algorithm because only the solution at substep $k-1$ is needed to compute the one at substep k. The coefficients α_k are the standard Runge-Kutta coefficients and are given by:

$$\alpha_k = \frac{1}{4-k}$$

The above time integration algorithm is third order accurate in the linear case, and second order accurate in the general non-linear case.

2.5 Dynamic meshes

In this work, an unstructured dynamic fluid mesh is represented by a pseudo structural model (see, for example, Batina [1]) where a fictitious linear spring is associated with each edge connecting two fluid grid points S_i and S_j and is attributed the following stiffness:

$$k_{ij} = \frac{1}{\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}} \tag{23}$$

The grid points located on the downstream and upstream boundaries are held fixed. The motion of those points located on the wall boundary is determined from the wall 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 a displacement driven pseudo structural problem 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:

$$\begin{cases} \bar{\delta} x_i = 2\delta^n x_i - \delta^{n-1} x_i \\ \bar{\delta} y_i = 2\delta^n y_i - \delta^{n-1} y_i \end{cases} \tag{24}$$

with $\delta^n x = x^{n+1} - x^n$. Next, the above predictions are corrected with a few explicit Jacobi relaxations on the static equilibrium equations as follows:

$$\begin{cases} \delta^{n+1} x_i &= \dfrac{\sum\limits_{j \in K(i)} k_{ij} \tilde{\delta} x_j}{\sum\limits_{j \in K(i)} k_{ij}} \\ \delta^{n+1} y_i &= \dfrac{\sum\limits_{j \in K(i)} k_{ij} \tilde{\delta} y_j}{\sum\limits_{j \in K(i)} k_{ij}} \end{cases} \tag{25}$$

Finally, the new positions are computed as:

$$\begin{cases} x_i^{n+1} &= x_i^n + \delta^{n+1} x_i \\ y_i^{n+1} &= y_i^n + \delta^{n+1} y_i \end{cases} \tag{26}$$

3 Computational and parallel implementation issues

3.1 Identification of the computational kernels

From Eqs. (11-22), it follows that our fluid solver contains essentially two kernels of elementary computations, one for the convective fluxes, and the other for the diffusive ones. Both type of computations can be described as three-step sequences of the form `Gather-Compute-Scatter`.

3.1.1 The convective flux kernel

The evaluation of the second term of $< 1 >$ in Eq. (11) using the numerical flux function $\Phi_{\mathcal{F}_c}$ (14) with the second order approximation outlined in Eq. (17) can be summarized as follows:

$$\begin{cases} H_{ij} = \int\limits_{\partial C_{ij}(t)} \vec{\mathcal{F}}_c(W).\vec{\nu}_i d\sigma = \Phi_{\mathcal{F}_{c_{ij}}}(W_i, W_j, \vec{\nu}_{ij}) \\ H_{ji} = -H_{ij} \end{cases} \tag{27}$$

where:

$$\vec{\nu}_{ij}(t) = \int\limits_{\partial C_{ij}(t)} \vec{\nu}_i d\sigma = \vec{\nu}_1(t) + \vec{\nu}_2(t) \tag{28}$$

Essentially, one-dimensional elementary convective fluxes are computed at the intersection between the control volumes $C_i(t)$ and $C_j(t)$ (see Figure 4 below). Each elementary flux contributes to a flux balance at the boundary of the control volume $C_i(t)$. This balance involves the accumulation over the set of neighboring vertices $K(i)$ of all computed fluxes. From the second of Eqs. (27), it follows that only H_{ij} needs to be computed in order to update the flux balances at the two end-point nodal values of edge $E_{ij} = \{S_i, S_j\}$. Therefore, the most efficient way for evaluating the convective fluxes is to loop over the list of the mesh edges and compute as follows:

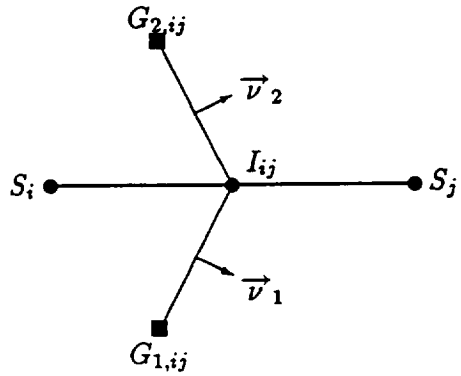

Figure 4: Evaluation of a convective flux along an edge $\{S_i, S_j\}$

For each edge $E_{ij} = \{S_i, S_j\}$ **of** $\mathcal{T}_h$ **Do**

Gather $W_i = W(S_i)$, $W_i = W(S_i)$
Gather $\vec{\nabla} W_i = \vec{\nabla} W(S_i)$, $\vec{\nabla} W_i = \vec{\nabla} W(S_i)$
Compute H_{ij}
Scatter $\Phi_i = \Phi_i + H_{ij}$
Scatter $\Phi_j = \Phi_j - H_{ij}$

End Do

3.1.2 The diffusive flux and nodal gradient kernel

In the last term $< 4 >$ of Eq. (11), the elementary diffusive flux $\vec{\mathcal{R}}_i(T)$ is constant in each triangle T. Its evaluation requires accessing the values of the physical state W at the three vertices S_i , S_j and S_k :

$$\left\{ \begin{array}{rl} \vec{\mathcal{R}}_i(T) & = \displaystyle\iint_T \vec{\mathcal{R}}(W).\nabla \vec{N}_i^T \, dxdy \\ & = area(T)(R_T \dfrac{\partial N_i^T}{\partial x} + S_T \dfrac{\partial N_i^T}{\partial y}) \end{array} \right. \tag{29}$$

The values of R_T and S_T contribute tọ the diffusive fluxes at all three vertices of triangle T. The sum symbol in $< 4 >$ is a clear indication

of a gather operation. Clearly, the most efficient way for evaluating the convective fluxes is to loop over the list of the mesh triangles and compute as follows:

For each element $T_{ijk} = \{S_i, S_j, S_k\}$ of $\mathcal{T}_h$ **Do**

Gather $W_i = W(S_i)$, $W_j = W(S_j)$, $W_k = W(S_k)$
Compute R_T, S_T
Scatter $\mathcal{V}_i = \mathcal{V}_i + \vec{\mathcal{R}}_i(T)$
Scatter $\mathcal{V}_j = \mathcal{V}_j + \vec{\mathcal{R}}_j(T)$
Scatter $\mathcal{V}_k = \mathcal{V}_k + \vec{\mathcal{R}}_k(T)$

End Do

The evaluation of the half-upwind nodal gradient (19) follows the same computational pattern described above.

3.2 The mesh partitioning with message-passing parallel paradigm

In addition to efficiency and parallel scalability, portability should be a major concern. With the proliferation of computer architectures, it is essential to adopt a programming model that does not require rewriting thousands of lines of code — or even worse, altering the architectural foundations of a code — every time a new parallel processor emerges. Here, we are neither referring to differences between programming languages, nor to differences between the multitude of parallel extensions to a specific programming language. We are more concerned about the impact of a given parallel hardware architecture on the software design, and sometimes, on the solution algorithm itself. For example, a data parallel code written for the CM-2 or CM-5 machines could require major rehauling before it can be adapted to an iPSC computer. A parallel-do-loop based code can be easily ported across different true shared memory multiprocessors, but may require substantial modifications before it can run successfully on some distributed memory systems.

Based on our "hands on" experience with a dozen of different parallel processors, we believe that the mesh partitioning and message-passing lead to portable software designs for parallel computational

mechanics. Essentially, the underlying mesh is assumed to be partitioned into several submeshes, each defining a subdomain. The same "old" serial code can be executed within every subdomain. The assembly of the subdomain results can be implemented in a separate software module and optimized for a given machine. This approach enforces data locality, and therefore is suitable for all parallel hardware architectures. For example, we have shown in [8] that for unstructured meshes, this approach produces substantially better performance results on the KSR-1 than the acclaimed virtual shared memory programming model. Note that in this context, message-passing refers to the assembly phase of the subdomain results. However, it does not imply that messages have to be explicitly exchanged between the 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 one.

In this work, we use essentially the same code on the iPSC-860, the KSR-1, and the CM-5 parallel processors. This code also runs on a workstation. We consider mesh partitions with and without overlapping for reasons that are discussed next.

3.2.1 Overlapping mesh partitions

The reader can verify that for the computations described herein, mesh partitions with overlapping simplify the programming of the subdomain interfacing module. Only one communication step is required, after the local physical states have been updated. Depending on the order of the spatial approximation, the overlapping region can be one or three triangles wide (see Figure 5 below). For a first order spatial approximation, we have by definition:

$$\begin{cases} W_{ij}^{\star} = W_i^{\star} \\ W_{ji}^{\star} = W_j^{\star} \end{cases} \tag{30}$$

which shows that the overlapping region needs in that case to be only one triangle wide.

However, mesh partitions with overlapping also have a drawback: they incur redundant floating-point operations.

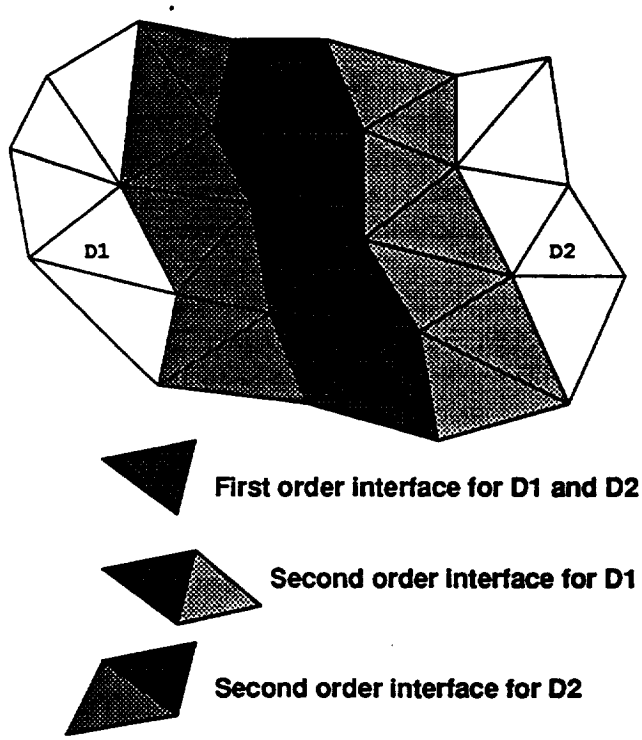

Figure 5: Overlapping mesh partition

For fixed meshes and overlapping partitions, the main loop of the parallel fluid solver described herein goes as follows:

Repeat $step = step + 1$

```
Compute the local time steps
```
For $srk = 1$ **to** $nsrk$ **Do**

```
Compute the nodal gradients
Compute the diffusive fluxes
Compute the convective fluxes
Update the physical states
Exchange the conservatives variables
```
End Do

Until $step = step_{max}$

In the above pseudo code, $step_{max}$ denotes the maximum number of time steps, and $nsrk$ denotes the number of steps in the Runge-Kutta integration algorithm.

All overlapping mesh partitions used in this investigation were generated by the decomposer described in [9].

3.2.2 Non-overlapping mesh partitions

Non-overlapping mesh partitions (see Figure 6 below) incur little redundant floating-point operations but induce one additional communication step. While physical state variables are exchanged between the subdomains in overlapping mesh partitions, partially gathered nodal gradients and partially gathered fluxes are exchanged between subdomains in non-overlapping ones.

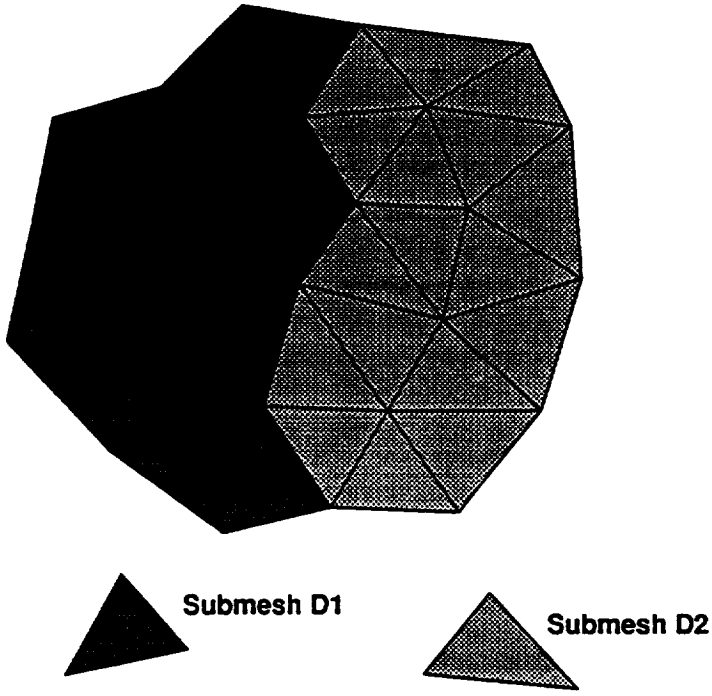

Figure 6: Non-overlapping mesh partition

For fixed meshes and non-overlapping partitions, the main loop of the parallel fluid solver described herein goes as follows:

Repeat $step = step + 1$

```
Compute the local time steps
```
For $srk = 1$ to $nsrk$ **Do**

```
Compute the nodal gradients
Compute the diffusive fluxes
Exchange the nodal gradients
Compute the convective fluxes
Exchange the convective fluxes
Update the physical states
```
End Do

Until $step = step_{max}$

All non-overlapping mesh partitions discussed in this investigation were generated by the TOP/DOMDEC software described in [4].

3.2.3 To overlap or not to overlap?

To answer this question, we analyze the communication requirements of both families of mesh partitions, and the amount of redundant computations they incur for two-dimensional fixed problems. Because we are interested in a comparative study, it suffices to consider the case of a single interface between two subdomains with uniform triangulations.

Let n_I^{nov} denote the number of interface vertices in a non overlapping mesh partition (see Figure 7). In the first communication step, $8 \times n_I^{nov}$ words related to the nodal gradients are exchanged between the two subdomains. In the second communication step, $4 \times n_I^{nov}$ fluxes are exchanged. Hence, the total communication cost per subdomain is given by:

$$T_{com}^{nov} = 2 \times T_s + 12 \times n_I^{nov} \times T_r \tag{31}$$

where T_s denotes the startup time of a message, and T_r denotes the transmit time for a 64-bit word.

In a non-overlapping mesh partition, the only redundant computations are those associated with the evaluation of the convective fluxes along the interface edges. Since an elementary convective flux requires about 200 floating-point operations, the total time per subdomain associated with redundant computations can be estimated as:

$$T_{red}^{nov} = 200 \times (n_I^{nov} - 1) \times T_a \tag{32}$$

where T_a denotes the times it takes to perform a single floating-point operation.

From Figure 8, it follows that the total number of overlapping vertices in an overlapping mesh partition is $4 \times n_I^{nov}$. In this case, only one communication step is required to exchange 4 components of the physical state at half of the overlapping vertices. Hence, the total amount of communication per subdomain is given by:

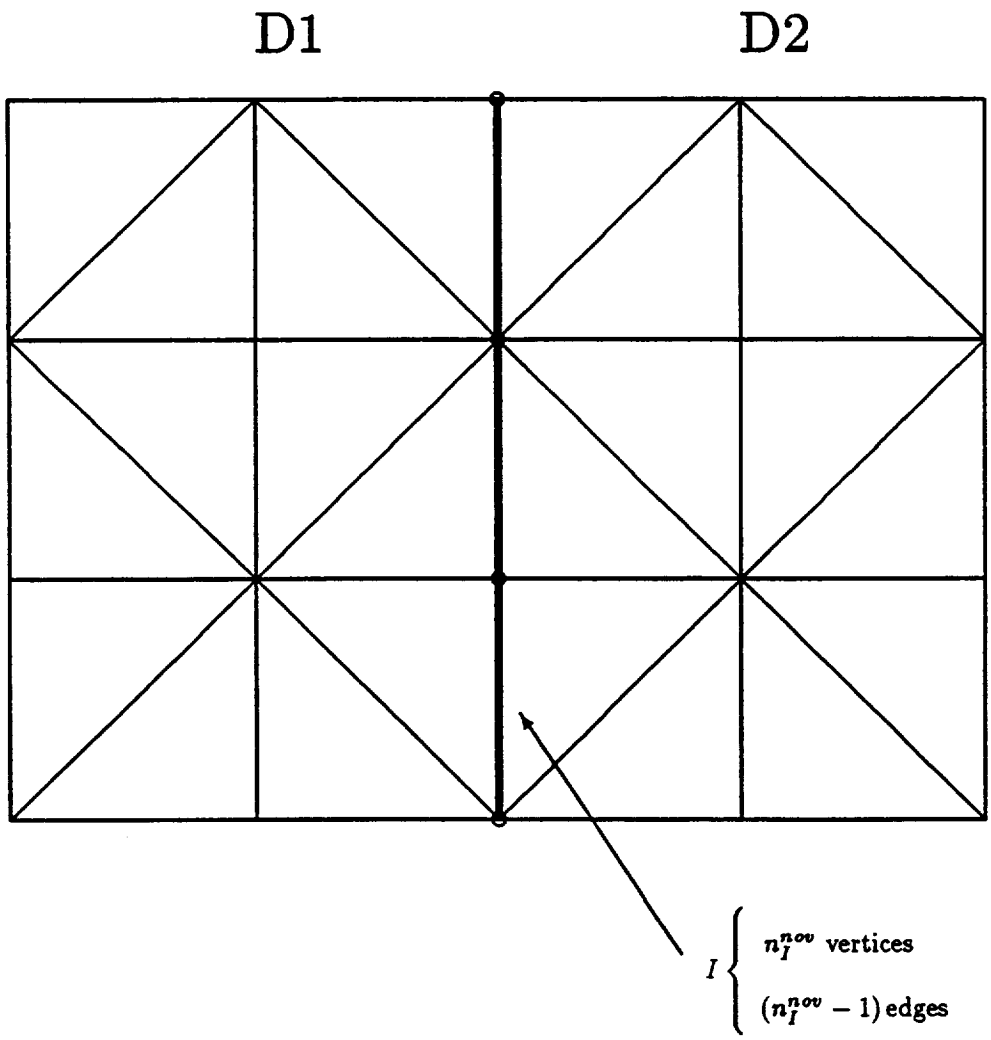

Figure 7: Analysis of a non-overlapping mesh partition

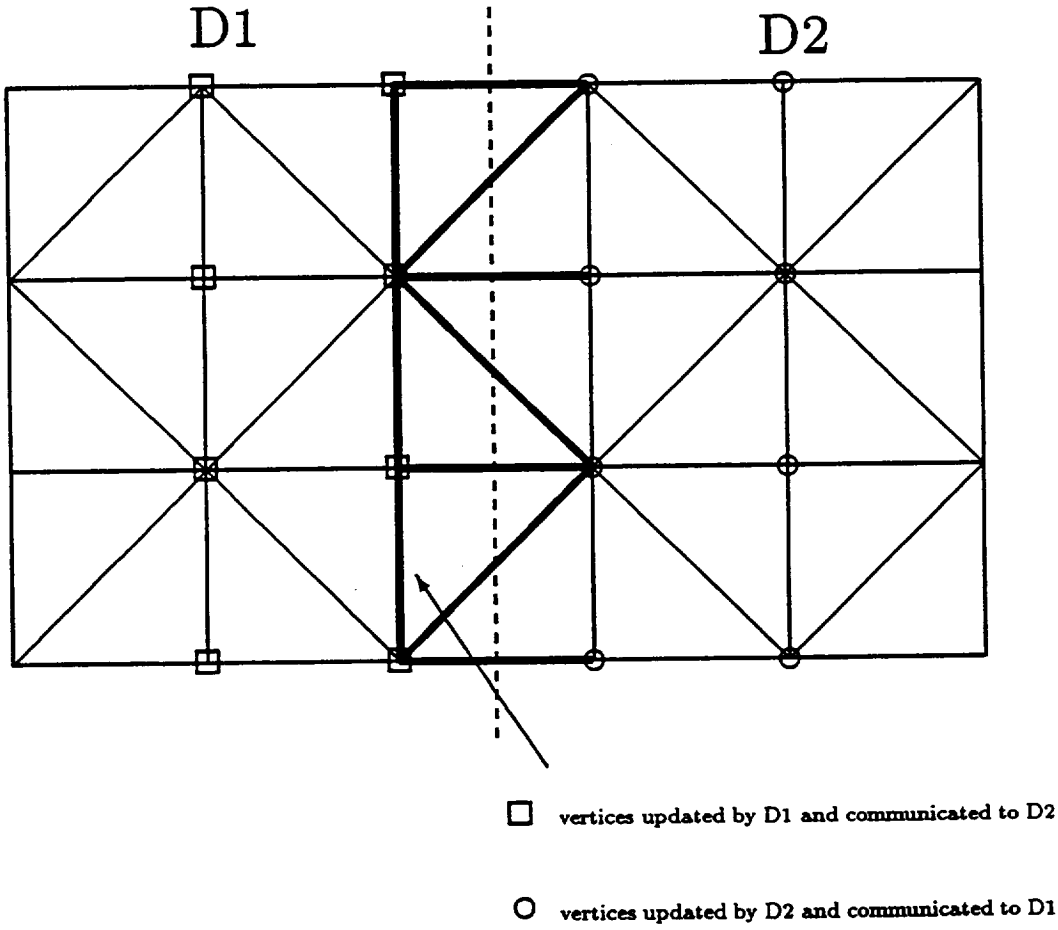

Figure 8: Analysis of an overlapping mesh partition

$$T_{com}^{ov} = T_s + 8 \times n_I^{nov} \times T_r \tag{33}$$

In an overlapping mesh partition, redundant computations are performed during the evaluation of both the convective and diffusive fluxes. Given that an elementary diffusive flux requires about 100 floating-point operations, the total time per subdomain associated with redudant computations is given in that case by:

$$T_{red}^{ov} = (200 \times (4 \times n_I^{nov}) + 100 \times (4 \times (n_I^{nov} - 1)) \times T_a \tag{34}$$

where $4 \times n_I^{nov}$ is the number of overlapping edges that generate redundant convective flux computations, and $4 \times (n_I^{nov} - 1)$ is the number of overlapping triangles that generate redundant diffusive flux computations.

From Eqs. (31,33), it follows that for sufficiently large messages we have:

$$\frac{T_{com}^{ov}}{T_{com}^{nov}} = \frac{T_s + 8 \times n_I^{nov} \times T_r}{2 \times T_s + 12 \times n_I^{nov} \times T_r} \approx \frac{2}{3} \tag{35}$$

which shows that overlapping the mesh partitions reduces the communication costs by 33%.

However from Eqs. (31-34), it follows that for a sufficiently large n_I^{nov} we have:

$$(T_{com}^{ov} + T_{red}^{ov}) - (T_{com}^{nov} + T_{red}^{nov}) \approx 1000 \times n_I^{nov} \times T_a (1 - \frac{1}{250} \times \frac{T_r}{T_a}) \tag{36}$$

which suggests that overlapping the mesh partitions will incur a greater total parallel overhead than not overlapping them, because of the resulting redundant computations. For example on the iPSC-860, $T_a = 0.2 \times 10^{-6}$ seconds (sustained 5 Mflops per processor), $T_r = 3.184 \times 10^{-6}$ seconds (sustained 2.5 Mbytes/second), and $(T_{com}^{ov} + T_{red}^{ov}) - (T_{com}^{nov} + T_{red}^{nov}) \approx 0.18 \times 10^{-3} \times n_I^{ov} > 0$. Nevertheless, this specific result also suggest that for two-dimensional problems, similar performance results will be obtained for mesh partitions with or without overlapping.

Finally, we caution the reader that different conclusions may be drawn for three-dimensional problems where overlapping can also require significantly more storage.

4 Performance results on a variety of MPPs

In this section, we discuss the parallel performance results obtained on various configurations of the iPSC-860, the KSR-1, and the CM-5 parallel processors.

4.1 Focus problem

We consider the numerical simulation of the unsteady viscous flow around a fixed NACA0012 airfoil, starting impulsively from a uniform flow. The angle of attack is set to 30°, and the free stream Mach number to 0.1. Several physical solutions of this problem were previously reported in [3] for different Reynolds numbers. All performance results reported herein are for 100 iterations and 64-bit arithmetic. More importantly, **the redundant floating-point operations are not counted** when evaluating the mflop rate, which is a strict approach to benchmarking.

A partial view of an unstructured triangulation of the computational domain is given in Figure 9. Seven meshes with increasing sizes have been generated. Their characteristics are summarized in Table 1 below where N_V denotes the number of vertices, N_T the number of triangles, and N_E the number of edges.

MESH	N_V	N_T	N_E
M1	8119	15998	24117
M2	16116	30936	47052
M3	32236	63992	96228
M4	63974	127276	191250
M5	131035	261126	392161
M6	262717	523914	786631
M7	523196	1044504	1567700

Table 1 : seven meshes and their characteristics

Throughout the remainder of this paper, the following nomenclature is used for the investigated mesh partitioning algorithms:

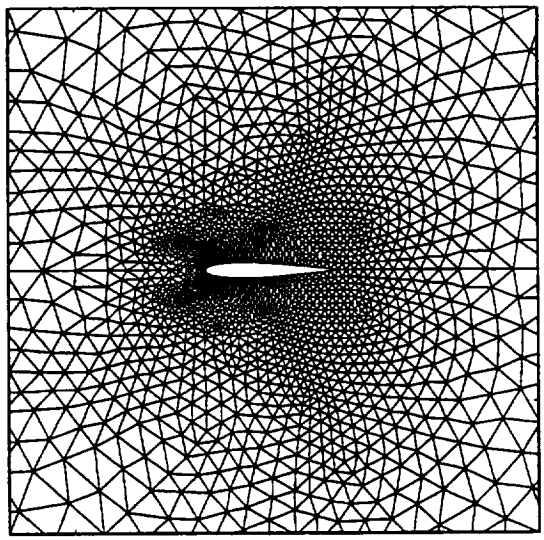

Figure 9: Partial view of a NACA0012 mesh

- **SCT** : Sector [9]
- **RIB** : Recursive inertial bisection [5]
- **GRD** : Greedy [5]
- **RGB** : Recursive graph bisection [16]
- **RSB** : Recursive spectral bisection [16]

4.2 Parallel scalability for increasing size problems

Parallel scalability is evaluated here for problems where the subdomain size is fixed, and the total size is increased with the number of processors. Note that because we are dealing with unstructured meshes, some slight deviations are inevitable. Overlapping mesh partitions are generated using the **RIB** heuristic.

Tables 2-3 summarize the performance results obtained on the iPSC-860 and KSR-1 parallel systems. The number of processors is denoted as N_p. The parallel CPU time and the Mflop rate are shown

to remain almost constant when the problem size is increased with the number of processors, which demonstrates the scalability of the parallel solver. The slight degradations in efficiency are mainly attributed to overlapping since redundant operations are not accounted for in the evaluation of the mflop rate. The KSR-1 processor cell is a RISC-style superscalar 64-bit unit operating at a peak of 40 Mflops. Clearly, despite a rather large number of gather/scatter operations, 25% of this peak performance is attained.

N_V	N_p	CPU Time	Mflop/s	Comm Time	% Comm
8119	1	491.2 s	6	0 s	0
16116	2	491.7 s	11	28.6 s	5.81
32236	4	514.7 s	22	31.3 s	6.09
63974	8	519.5 s	43	40.0 s	7.71
131035	16	536.6 s	85	40.3 s	7.52
262717	32	548.6 s	159	44.8 s	8.17

Table 2 : Parallel scalability for increasing size problems
Computations with *overlapping* mesh partitions on the iPSC-860

N_V	N_p	CPU Time	Mflop/s	Comm Time	% Comm
8119	1	272.3 s	10	0 s	0
16116	2	272.3 s	20	1.8 s	0.67
32236	4	286.0 s	39	3.2 s	1.18
63974	8	289.3 s	77	4.3 s	1.52
131035	16	306.7 s	149	5.9 s	1.95
262717	32	316.1 s	276	8.4 s	2.66

Table 3 : Parallel scalability for increasing size problems
Computations with *overlapped* mesh partitions on the KSR-1

4.3 Influence of the mesh partitioning algorithm

Next, we focus on mesh *M6* with 32 processors and non-overlapping partitions, and investigate the influence of the partitioning algorithm on parallel performance. Tables 4 and 5 report the measured CPU time and Mflop rates. "Conv" and "Diff" designate respectively the

convective and diffusive fluxes. In all cases, the **RSB** algorithm yields the fastest solution time, even when it does not produce the smallest communication time. The reason is that, for mesh *M6*, the **RSB** algorithm does a better job than the others at generating subdomains that are well balanced vertex-wise, element-wise, and edge-wise, simultaneously.

Decomp	CPU Time	Conv Time	Diff Time	Comm Time	Mflop/s
GRD	550.0 s	330.4 s	163.4 s	47.4 s	158
RGB	556.3 s	335.1 s	157.9 s	62.7 s	158
RSB	538.0 s	326.9 s	153.7 s	53.8 s	162

Table 4 : Influence of the mesh partitioning algorithm
Computations with *non-overlapping* mesh partitions on an iPSC-860/32

Decomp	CPU Time	Conv Time	Diff Time	Comm Time	Mflop/s
GRD	343.7 s	165.9 s	112.8 s	18.9 s	254
RGB	340.1 s	168.3 s	105.1 s	18.5 s	256
RSB	322.5 s	160.4 s	101.4 s	14.2 s	270

Table 5 : Influence of the mesh partitioning algorithm
Computations with *non-overlapping* mesh partitions on a KSR-1/32

The reader can verify that the numbers reported in columns 3 to 5 of Tables 4-5 do not add up to the total CPU time reported in column 2. The difference corresponds to various parallel and sequential overheads. On the iPSC-860, these overheads represent less than 2% of the total solution time. However on the KSR-1, they represent about 14% of the total CPU time. Indeed, the computing mode on the iPSC-860 is parallel by default, while on the KSR-1 it is sequential by default. Hence, many fork-join type of procedures are necessary on the KSR-1, which explains the relatively large amount of overhead.

Also, note that using the same number of processors, the KSR-1 is twice as fast as the iPSC-860 at computing the convective fluxes, but only 1.5 times faster at computing the diffusive ones. This is because the evaluation of the convective fluxes requires less indirect addressing than the evaluation of the diffusive ones.

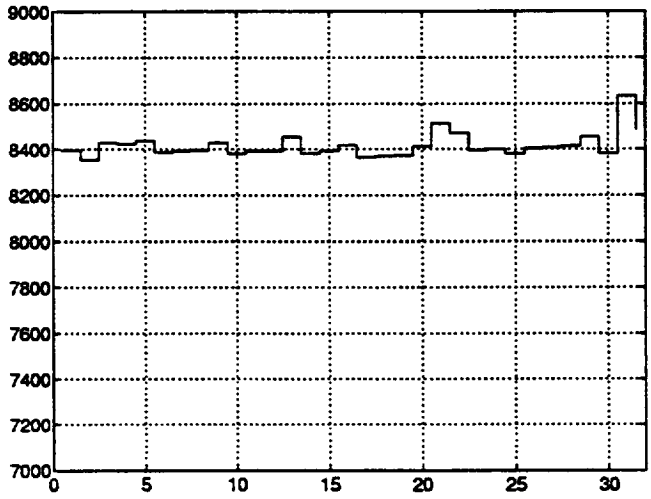

Figure 10: Vertex-wise load bal. - 32 *non-overlapping* sub. **GRD** algorithm

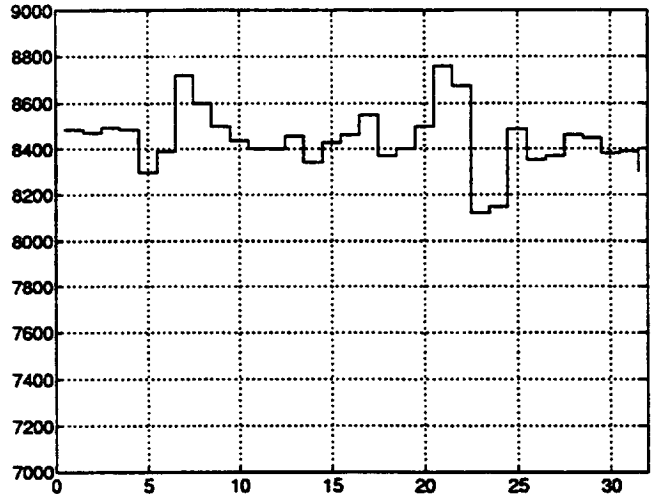

Figure 11: Vertex-wise load bal. - 32 *non-overlapping* sub. **RGB** algorithm

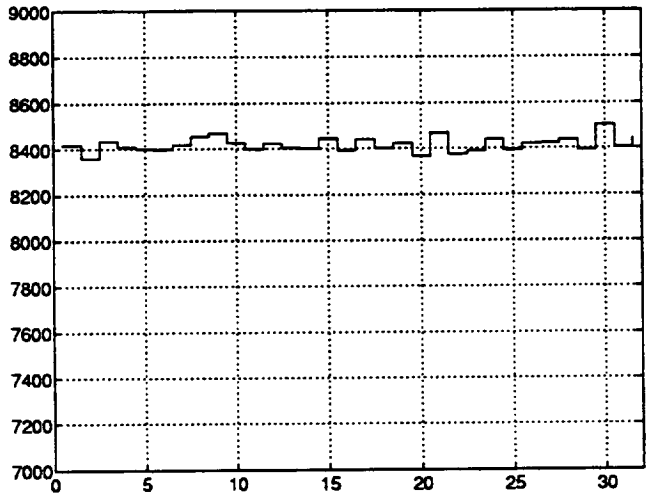

Figure 12: Vertex-wise load bal. - 32 *non-overlapping* sub.
RSB algorithm

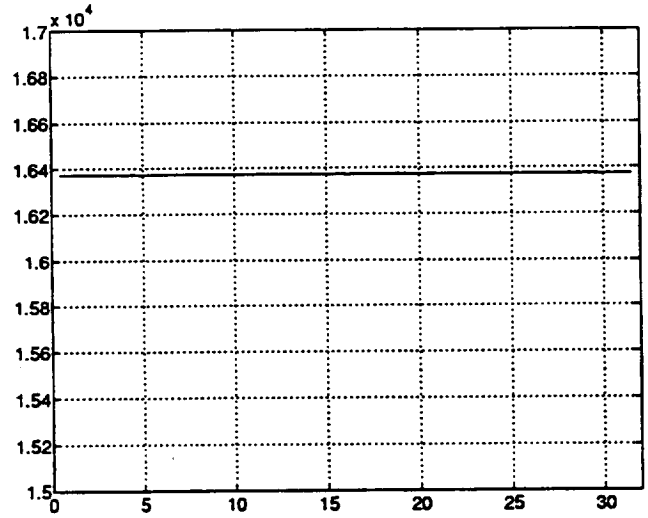

Figure 13: Element-wise load bal. - 32 *non-overlapping* sub.
GRD algorithm

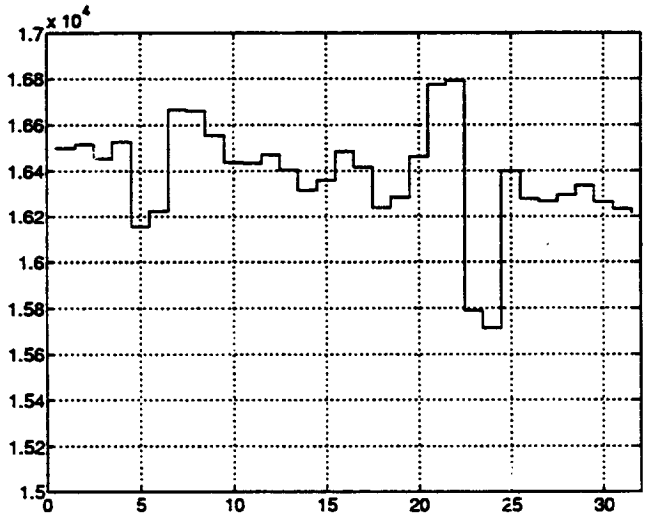

Figure 14: Element-wise load bal. - 32 *non-overlapping* sub. **RGB** algorithm

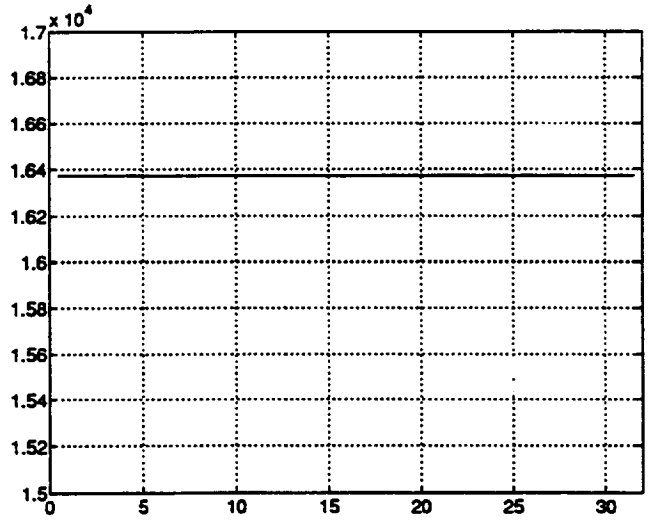

Figure 15: Element-wise load bal. - 32 *non-overlapping* sub. **RSB** algorithm

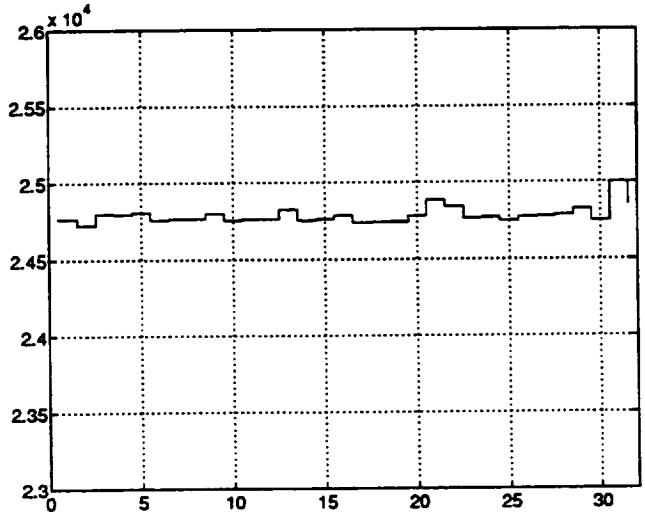

Figure 16: Edge-wise load bal. - 32 *non-overlapping* sub.
GRD algorithm

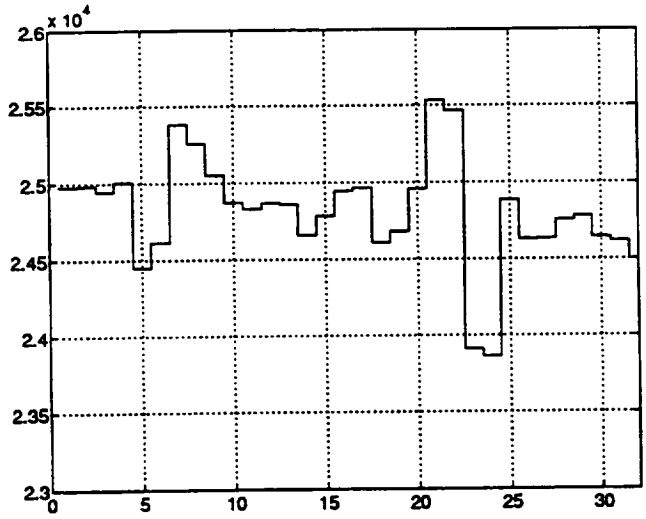

Figure 17: Edge-wise load bal. - 32 *non-overlapping* sub.
RGB algorithm

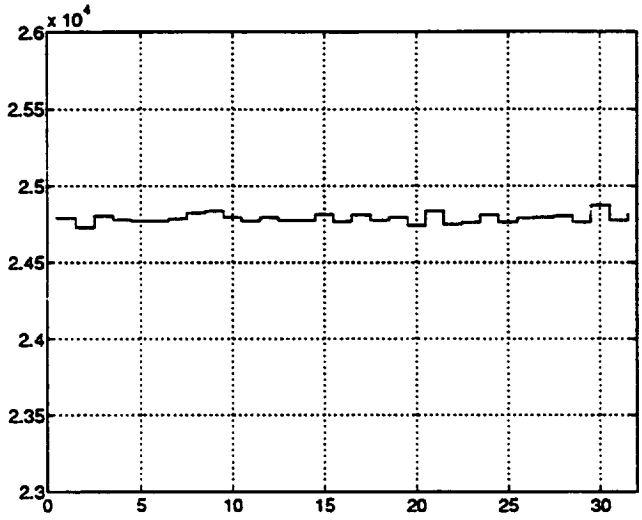

Figure 18: Edge-wise load bal. - 32 *non-overlapping* sub. **RSB** algorithm

4.4 Parallel scalability for fixed size problems

Tables 6-8 summarize the performance results obtained on the iPSC-860 and KSR-1 parallel processors for overlapping mesh partitions generated by the **RIB** algorithm. For the largest mesh *M7*, a Gigaflop performance level is attained using 128 processors of the KSR-1 system. Good scalability is observed on both machines.

N_p	CPU Time	Conv Time	Diff Time	Comm Time	Mflop/s
32	548.7 s	323.1 s	164.6 s	44.9 s	159
64	282.7 s	163.6 s	82.4 s	24.4 s	309
128	144.5 s	82.2 s	42.7 s	18.9 s	617

Table 6 : Performance results for mesh *M6*
Computations with *overlapping* mesh partitions on the iPSC-860

N_p	CPU Time	Conv Time	Diff Time	Comm Time	Mflop/s
32	316.1 s	155.4 s	111.3 s	8.4 s	276
64	169.7 s	79.0 s	58.1 s	9.8 s	514
128	93.0 s	39.7 s	31.2 s	6.4 s	938

Table 7 : Performance results for mesh *M6*
Computations with *overlapping* mesh partitions on the KSR-1

N_p	CPU Time	Conv Time	Diff Time	Comm Time	Mflop/s
128	174.0 s	77.7 s	60.2 s	14.0 s	1024

Table 8 : Performance results for mesh *M7*
Computations with *overlapping* mesh partitions on the KSR-1

Finally, Tables 9-10 compare the performances of the iPSC-860 and KSR-1 parallel systems for the case of mesh *M6* and non-overlapping mesh partitions generated by the **GRD** algorithm. For the same number of processors, the KSR-1 machine is reported to be 1.67 times faster than the iPSC-860, even though its basic processor is supposed to be 1.5 times slower than that of the iPSC-860.

N_p	CPU Time	Conv Time	Diff Time	Comm Time	Mflop/s
64	287.7 s	164.0 s	79.8 s	42.3 s	303

Table 9 : Performance results for mesh *M6*
Computations with *non-overlapping* mesh partitions on the iPSC-860

N_p	CPU Time	Conv Time	Diff Time	Comm Time	Mflop/s
64	171.8 s	83.1 s	51.1 s	14.1 s	508
128	94.8 s	40.1 s	24.3 s	15.2 s	921

Table 10 : Performance results for mesh *M6*
Computations with *non-overlapping* mesh partitions on the KSR-1

4.5 Performance results on the CM-5

Recently, we have implemented our fluid solver on a 32 processor CM-5 system using Fortran 77 on a node and the CMMD message passing library (this corresponds to the Sparc model of computation on

the CM-5). For mesh *M6* and a 32 subdomain decomposition with overlapping using the **RIB** algorithm, Table 11 reports the measured performance results and compares them to those obtained on an iPSC-860/32 and a KSR-1/32 computers.

MPP	CPU Time	Conv Time	Diff Time	Comm Time	Mflop/s
CM-5	855.4 s	541.3 s	241.7 s	44.0 s	102
iPSC-860	548.6 s	323.1 s	164.6 s	44.8 s	159
KSR-1	316.1 s	155.4 s	111.3 s	8.4 s	276

Table 11 : Performance results for mesh *M6*
Computations with *overlapping* mesh partitions

Clearly, the results reported in Table 11 for the CM-5 are not as impressive as those reported, for example, in [15, ?]. This can be attributed to several factors including the use of the message-passing model for portability reasons, the variety of gather/scatter operations required by our specific fluid solver, and more importantly our strict approach to performance benchmarking.

Next, we report performance results on the CM-5 using a global CM Fortran approach (this corresponds to the Vector Units model of computation on the CM-5), and the parallel version of our solver that was previously developed for the CM-2/200 and described in [3]. In this so called data parallel approach, all local computations are carried out on a control volume. Therefore, all data structures are vertex based. Note that this approach generates a substantial amount of redundant computations. The reported communication timings correspond to the inter and intra vector units gather/scatter operations. In Table 12, MFU is a Mflop rate that does not account for redundant arithmetic operations, while MFR is a Mflop rate that does.

CPU Time	Conv Time	Diff Time	Comm Time	MFU	MFR
342.0 s	116.9 s	49.0 s	162.0 s	107	238

Table 12 : Performance results for mesh *M5*
Computations with *overlapping* mesh partitions on the CM-5

The results reported in Table 12 show that about 50% of the elapsed time is spent in gather/scatter operations, which is consistent with the results obtained by other investigators for two-dimensional finite element fluid problems [7]. Note that this percentage is higher than those observed on the iPSC-860 and KSR-1 systems.

5 Applications

5.1 Steady viscous flow inside a model jet engine

First, we consider the numerical simulation of a steady viscous flow inside a model jet engine. The free stream Mach number and the Reynolds number are set respectively to 0.2 and 2000. The computational grid is illustrated in Figure 19. Its characteristics are $N_V = 12233$, $N_T = 22936$, and $N_E = 35170$.

This simulation is carried out on the KSR-1 using overlapping mesh partitions generated by the **RIB** algorithm. Here, the pseudo time integration is carried out at CFL=1.9 via a four step Runge-Kutta method with $\alpha_1 = 0.11$, $\alpha_2 = 0.2766$, $\alpha_3 = 0.5$ and $\alpha_4 = 1.0$. A local time step strategy is introduced in order to accelerate convergence. After 1527 iterations, the initial residual is reduced by a factor of 10^4. The resulting steady mach lines are depicted in Figure 20.

Table 13 reports the performance results obtained on the KSR-1, and Table 14 summarizes the characteristics of the generated mesh partitions. $S(p)$ denotes the speed-up using p processors. N_I denotes the total number of interface vertices, and Max N_I denotes the maximum number of interface vertices per subdomain. The jump of communication costs between the case with $N_p = 2$ and that with $N_p = 4$ can be attributed to the accidental increase in Max N_I, which implies an increase in the maximum message length. The slightly superlinear speed-up observed for $N_p = 4$ is not uncommon on the KSR-1. If the problem to be solved does not fit exactly into a single processor's memory, the solution on one processor still involves some interprocessor communication that is difficult to time by the user.

N_p	CPU Time	Mflop/s	Comm Time	% Comm	$S(p)$
1	8946 s	9	—	—	1.0
2	4532 s	17	28.7 s	0.63	1.9
4	2136 s	37	49.9 s	2.33	4.1
8	1168 s	68	46.3 s	3.96	7.6

Table 13 : Performance results for an internal flow simulation

Computations with *overlapping* mesh partitions on the KSR-1

N_p	Min N_V	Max N_V	Min N_T	Max N_T	Max N_I
2	6222	6229	11568	11668	113
4	3205	3334	5942	6183	171
8	1593	1774	2964	3204	86

Table 14 : Characteristics of the mesh partitions

5.2 Airfoil flutter and control surface

Next, we consider the flutter simulation of a NACA0012 airfoil in transonic flow. The structural dynamics behavior of the airfoil is represented by a two-spring two-degree of freedom system. The airfoil twist θ is associated with a torsional spring and monitors the angle of attack. The lateral deflection h is associated with a lineal spring and monitors the airfoil bending. The evolution of this system is governed by a set of differential equations that can be written in non-dimensional form as follows (for example, see [6]):

$$\begin{cases} \dfrac{d^2\bar{h}}{d\bar{t}^2} + \dfrac{x_\theta}{2}\dfrac{d^2\theta}{d\bar{t}^2} + \dfrac{4\xi_h\omega_h M_\infty}{V^*\omega_\theta}\dfrac{d\bar{h}}{d\bar{t}} + \dfrac{4\omega_h^2 M_\infty^2}{V^{*2}\omega_\theta^2}\bar{h} = -\dfrac{2M_\infty^2 C_l}{\pi\mu} \\ \dfrac{x_\theta}{2}\dfrac{d^2\bar{h}}{d\bar{t}^2} + \dfrac{\gamma_\theta^2}{4}\dfrac{d^2\theta}{d\bar{t}^2} + \dfrac{\xi_\theta\gamma_\theta^2 M_\infty}{V^*}\dfrac{d\theta}{d\bar{t}} + \dfrac{\gamma_\theta^2 M_\infty^2}{V^{*2}}\theta = \dfrac{2M_\infty^2 C_m}{\pi\mu} \end{cases} \tag{37}$$

In Eqs. (37) above, the bar superscript indicates a non-dimensional variable, and C_l and C_m denote respectively the lift coefficient and the torsional moment. These two quantities are related to the generalized aerodynamics forces Q_h and Q_θ by:

$$\begin{cases} Q_h &= \dfrac{1}{2}\rho_\infty V_\infty^2 (2b) C_l \\ Q_\theta &= \dfrac{1}{2}\rho_\infty V_\infty^2 (2b)^2 C_m \end{cases} \tag{38}$$

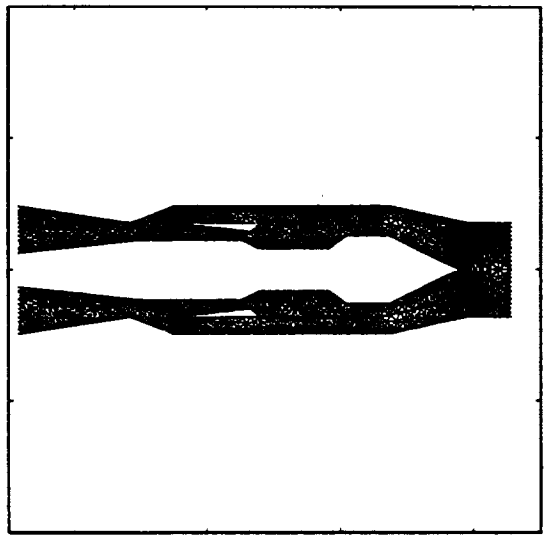

Figure 19: View of the discretization of a model jet engine

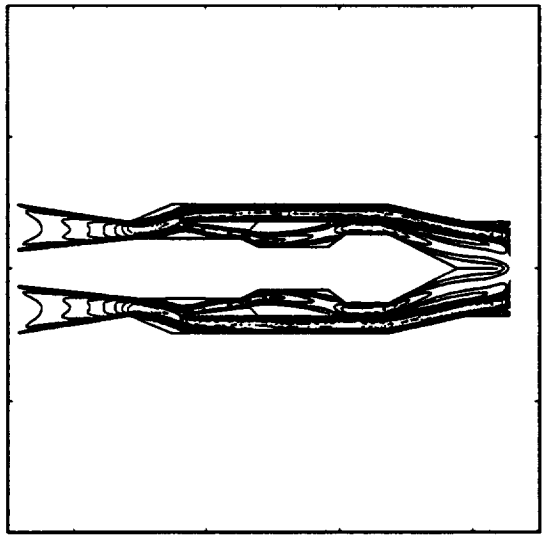

Figure 20: Mach lines : $Min = 0.0$, $Max = 0.6$, $\Delta M = 0.05$

where $2b$ is the airfoil chord, $V^\star = \dfrac{V_\infty}{b\omega_\theta}$ is the normalized velocity, V_∞ is the free stream velocity, and $\mu = \dfrac{m}{\pi\rho_\infty b^2}$ is the ratio of the airfoil mass per transversal unit to the free stream density.

The effect of an additional control surface such as a flap is simulated with the superposition of a controlled motion of a fraction of the airfoil trailing edge. This secondary motion is defined by the flap angle $\delta(t)$, which obeys the following control law:

$$\delta(t) = G_h h(t) e^{i\varphi_h} + G_\theta \theta(t) e^{i\varphi_\theta} \tag{39}$$

where G_h et G_θ are gain coefficients, and φ_h and φ_θ phase angles (see Figures 21-23).

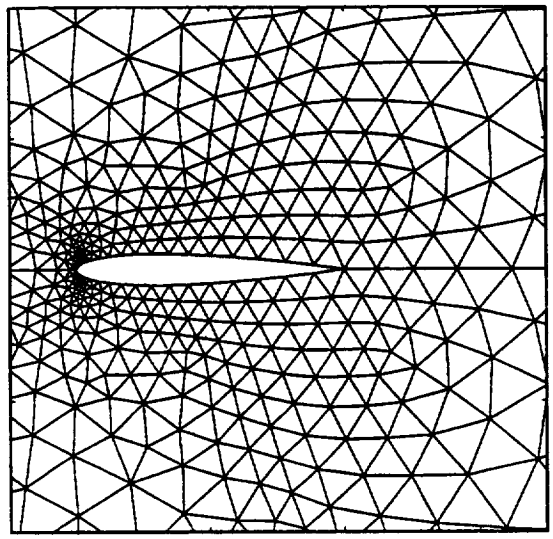

Figure 23: $\alpha = 12°$, $\delta = 12°$, $G_\theta = 1.0$
(mesh is coarsened for clarity)

Two aeroelastic simulations with and without the control surface are performed using mesh $M4$ and:

$$2b = 1, \quad m = 1, \quad \xi_\theta = \xi_h = 0$$

$$\omega_\theta = \omega_h = 100 \; rad/s$$

$$\gamma_\theta = 1.865, \quad \mu = 60, \quad a_h = -2, \quad x_\theta = 1.8$$

$$V^\star = 0.6, \quad Q^\star = 0.006, \quad V_\infty = 30.0 m/s$$

The flow initial conditions are identified with the steady solution at $M_\infty = 0.8$ and zero angle of attack. After the steady state is reached, a perturbation in the angle of attack $\Delta\theta = 0.01$ radian is introduced, which causes the airfoil to vibrate and the flow to become unsteady. The governing aeroelastic equations (37) are used to predict the dynamic response of the system. The unsteady fluid flow equations are time integrated with a global time step strategy. At each time step,

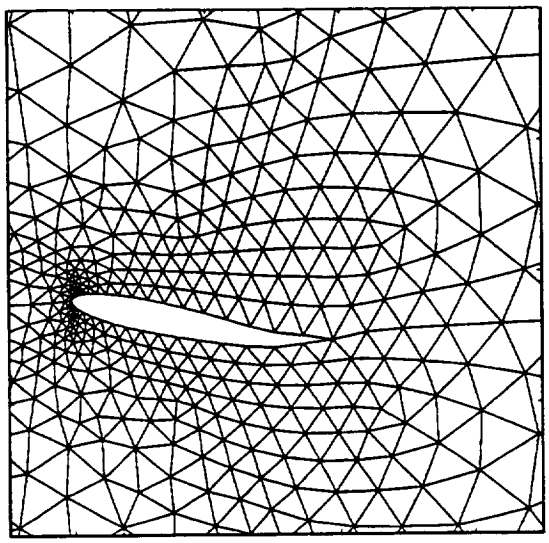

Figure 21: $\alpha = 0°$, $\delta = 0°$, $G_\theta = 0.0$
(mesh is coarsened for clarity)

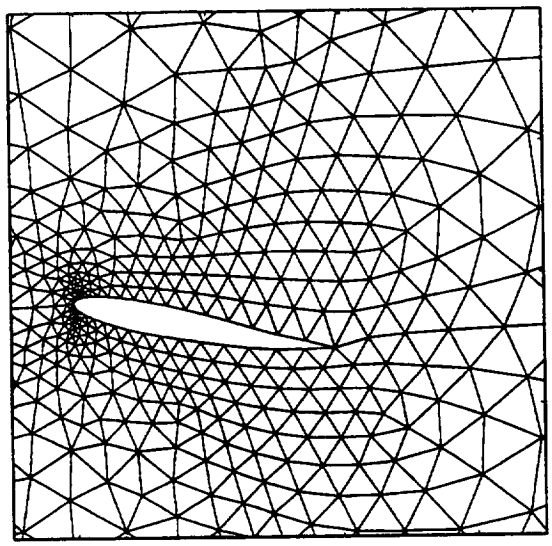

Figure 22: $\alpha = 12°$, $\delta = -6°$, $G_\theta = 0.5$
(mesh is coarsened for clarity)

the dynamic mesh is updated with 8 explicit Jacobi relaxations as described in Section 2.5. All computations are run on both the iPSC-860 and KSR-1 parallel processors.

Figures 24 and 25 report the evolution in time of the angle of attack θ and the lift coefficient C_l. Clearly, when the control surface is enabled with $G_h = G_\theta = 0.75$ and $\varphi_h = \varphi_\theta = \pi$, a stable aeroelastic response is observed. When it is disabled, a flutter instability is reached.

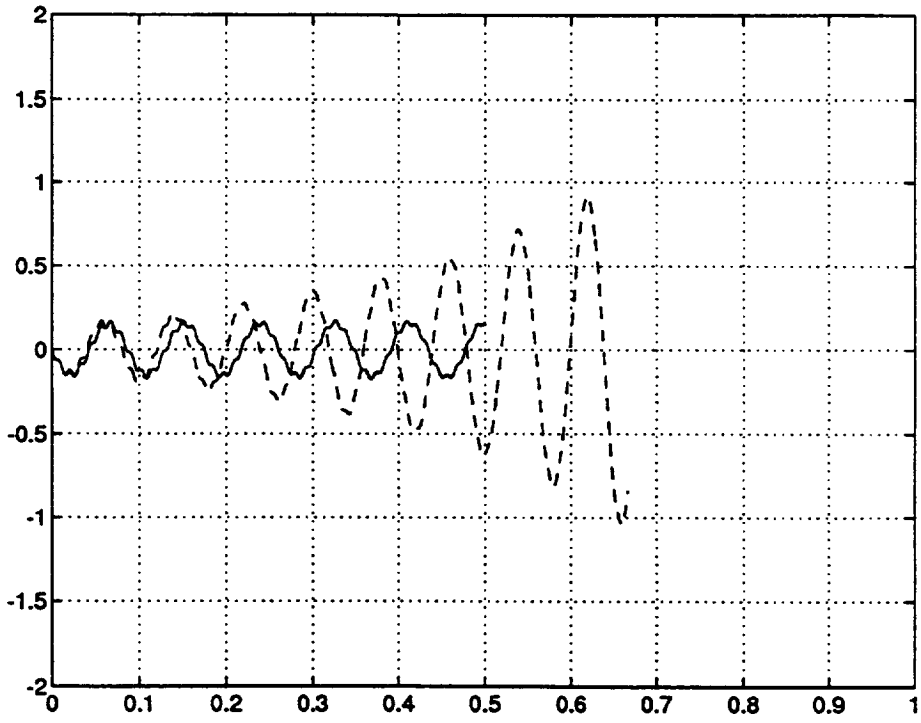

Figure 24: Angle of attack θ (in °) versus physical time t (in seconds)
$G_h = G_\theta = 0.75$, $\varphi_h = \varphi_\theta = \pi$

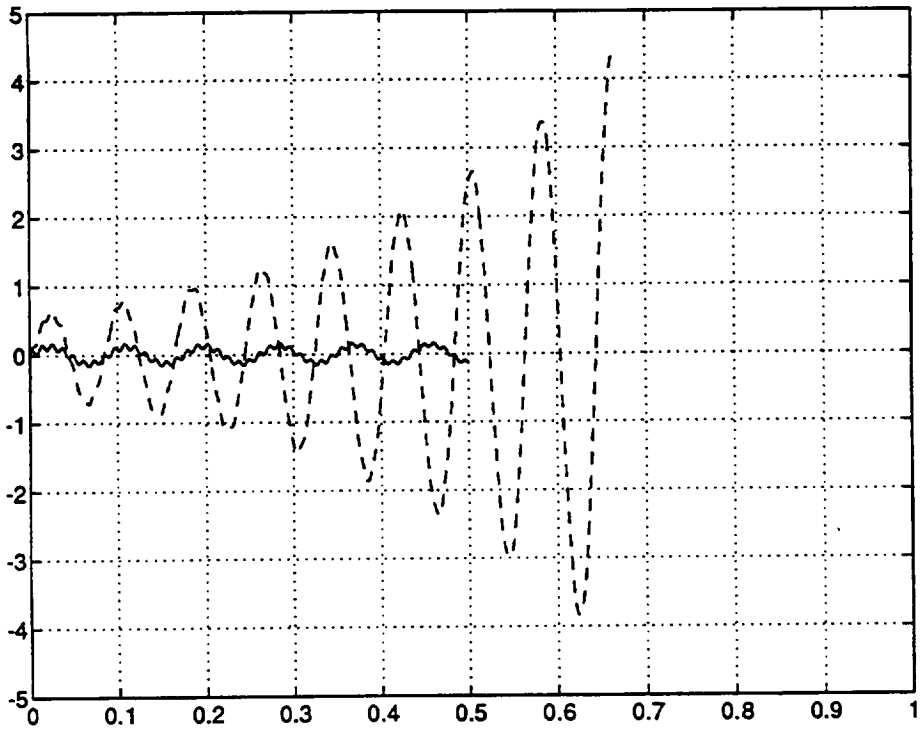

Figure 25: Lift coefficient C_l versus physical time t (in seconds)
$G_h = G_\theta = 0.75$, $\varphi_h = \varphi_\theta = \pi$
- - - - : Without active control surface
—— : With active control surface

For this application, the performance results obtained on the iPSC-860 and KSR-1 parallel processors are summarized in Tables 15-16. Simulation time is reported for 100 steps and includes sequential I/O costs for saving the computed solutions on disk.

N_p	Simulation Time	Flux Comp Time	Mesh Update	Comm Time
16	372.0 s	243.2 s	32.4 s	33.5 s
32	208.0 s	124.0 s	17.7 s	17.1 s
64	101.0 s	64.0 s	10.1 s	12.8 s

Table 15 : 100 steps of an aeroelastic simulation with mesh $M4$
Computations with *overlapping* mesh partitions on the iPSC-860

N_p	Simulation Time	Flux Comp Time	Mesh Update	Comm Time
16	326.0 s	135.0 s	35.8 s	18.8 s
32	176.0 s	69.2 s	19.1 s	18.9 s
64	99.7 s	35.1 s	11.0 s	15.8 s

Table 16 : 100 steps of an aeroelastic simulation with mesh $M4$
Computations with *overlapping* mesh partitions on the KSR-1

Both parallel processors are shown to deliver good speed-ups. Interprocessor communication time varies between 9% and 12.7 % on the iPSC-860, and 5.7% and 15.8% on the KSR-1. The cost of updating the dynamic mesh is about 10% of the total cost only. Note that for this simulation, the KSR-1 is not reported to be 1.5 times faster than the iPSC-860 for the same number of processors, unlike in all previous cases. This suggests that I/O on the KSR-1 is more expensive than on the iPSC-860.

Acknowledgments
The authors acknowledge partial support by the NASA Lewis Research Center under Grant NAG3-1425, and partial support by the National Science Foundation under Grant ASC-9217394.

References

[1] BATINA J.T., *Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes*, AIAA Paper No. 89-0115, AIAA 27th Aerospace Sciences Meeting, Reno, Nevada, June 9-12, (1989).

[2] DONEA J., *An Arbitrary Lagrangian-Eulerian Finite Element Method for Transient Fluid-Structure Interactions*, Comp. Meth. Appl. Mech. Engng., Vol. 33, pp. 689-723, (1982).

[3] FARHAT C. - FEZOUI L. - LANTERI S., *Two-Dimensional Viscous Flow Computations on the CM-2: Unstructured Meshes, Upwind Schemes and Massively Parallel Computations*, Comp. Meth. in Appl. Mech. and Eng., Vol 102, pp. 61-88, (1993).

[4] FARHAT C. - LANTERI S. - SIMON H., *TOP/DOMDEC : a Software Tool for Mesh Partitioning and Parallel Processing and Applications to CSM and CFD Computations*, Comput. Sys. Engrg. (To Appear)

[5] FARHAT C. - LESOINNE M., *Automatic Partitioning of Unstructured Meshes for the Parallel Solution of Problems in Computational Mechanics*, Internat. J. Numer. Meths. Engrg., Vol. 36, pp. 745-764, (1993).

[6] FARHAT C. - LIN T.Y., *Transient Aeroelastic Computations Using Multiple Moving Frames of Reference*, AIAA Journal, Vol. 31, No. 3, pp. 597-599, (1993).

[7] KENNEDY J. et al., *A Strategy for Implementing Implicit Finite Element Methods for Incompressible Fluids on the CM-5*, Symposium on Parallel Finite Element Computations, Minneapolis, Minnesota, October 24-27, (1993).

[8] LANTERI S. - C. FARHAT, *Viscous Flow Computations on M.P.P. Systems : Implementational Issues and Performance Results for Unstructured Grids*, Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing, Norfolk, Virginia, pp. 65-70, (1993).

[9] M. LORIOT, *MS3D : Mesh Splitter for 3D Applications, User's Manual*

[10] N'KONGA B. - GUILLARD H., *A Roe Scheme on Non-Structured Meshes for Moving Boundaries Problems*, Proceedings of the 13th IMACS World Congress on Computational and Applied Mathematics, Trinity College, Dublin, Ireland, July 22-26, Vol. 1, pp-447-449, (1191).

[11] ROE P. L., *Approximate Riemann Solvers, Parameters Vectors and Difference Schemes*, Journ. of Comp. Phys., **43**, pp. 357-371, (1981).

[12] STEGER J. - WARMING R.F., *Flux vector splitting for the inviscid gas dynamic with applications to finite-difference methods*, Journ. of Comp. Phys., 40, (2), pp. 263-293, (1981).

[13] VAN LEER B., *Towards the Ultimate Conservative Difference Scheme V : a Second-Order Sequel to Goudonov's Method*, Journ. of Comp. Phys., Vol 32, pp. 361-370, (1979).

[14] ZDENEK J. - MATHUR K. K. - JOHNSSON S. L. - HUGHES T. J. R., *An Efficient Communication Strategy for Finite Element Methods on the Connection Machine CM-5 System*, Thinking Machines Technical Report No. 256, (1993).

[15] TEZDUYAR T.E. - ALIABADI S. K. - BEHR M. - MITTAL S. - JOHNSON A., *Parallel Finite Element Computation of 3D Flows*, Computer, IEEE Computer Society, Vol. 26, pp. 27-36, (1993).

[16] SIMON H., *Partitioning of Unstructured Problems for Parallel Processing*, Comput. Sys. Engrg., Vol. 2, pp. 135-148, (1991).

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1994	Final Contractor Report

4. TITLE AND SUBTITLE

High Performance Parallel Analysis of Coupled Problems for Aircraft Propulsion

5. FUNDING NUMBERS

WU-509-10-11
G-NAG3-1425

6. AUTHOR(S)

C.A. Felippa, C. Farhat, S. Lanteri, N. Maman, S. Piperno, and U. Gumaste

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

College of Engineering
University of Colorado
Campus Box 429
Boulder, Colorado 80309

8. PERFORMING ORGANIZATION REPORT NUMBER

E-9007

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR-195335

11. SUPPLEMENTARY NOTES

Project Manager, Christos C. Chamis, Structures Division, NASA Lewis Research Center, organization code 5200, (216) 433-3252.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Category 39

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

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. In this paper, we present several partitioned procedures for time-integrating this focus coupled problem and discuss their merits in terms of accuracy, stability, heterogeneous computing, I/O transfers, subcycling, and parallel processing. All theoretical results are derived for a one-dimensional piston model problem with a compressible flow, because the complete three-dimensional aeroelastic problem is difficult to analyze mathematically. However, the insight gained from the analysis of the coupled piston problem and the conclusions drawn from its numerical investigation are confirmed with the numerical simulation of the two-dimensional transient aeroelastic response of a flexible panel in a transonic nonlinear Euler flow regime.

14. SUBJECT TERMS

Time integrating; Partition procedures; Parallel processors Lagrangian; Eulerian fluid/ structure; Moving mesh; Piston problem Navier-Stokes flow solution stability

15. NUMBER OF PAGES
107

16. PRICE CODE
A06

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	