

NASA Technical Memorandum 108805

108805
24 P

A Comparative Study of Serial and Parallel Aeroelastic Computations of Wings

Chansup Byun and Guru P. Guruswamy

(NASA-TM-108805) A COMPARATIVE
STUDY OF SERIAL AND PARALLEL
AEROELASTIC COMPUTATIONS OF WINGS
(NASA) 24 p

N94-26538

Unclas

G3/02 0208971

January 1994

NASA

National Aeronautics and
Space Administration



A Comparative Study of Serial and Parallel Aeroelastic Computations of Wings

Chansup Byun and Guru P. Guruswamy

January 1994



National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

A COMPARATIVE STUDY OF SERIAL AND PARALLEL AEROELASTIC COMPUTATIONS OF WINGS

Chansup Byun* and Guru P. Guruswamy
Ames Research Center

SUMMARY

This paper presents a procedure for computing the aeroelasticity of wings on parallel multiple-instruction, multiple-data (MIMD) computers. In this procedure, fluids are modeled using Euler equations, and structures are modeled using modal or finite element equations. The procedure is designed in such a way that each discipline can be developed and maintained independently by using a domain decomposition approach. In the present parallel procedure, each computational domain is scalable. A parallel integration scheme is used to compute aeroelastic responses by solving fluid and structural equations concurrently. The computational efficiency issues of parallel integration of both fluid and structural equations are investigated in detail. This approach, which reduces the total computational time by a factor of almost 2, is demonstrated for a typical aeroelastic wing by using various numbers of processors on the Intel iPSC/860.

INTRODUCTION

In recent years, significant advances have been made for single-discipline use of both computational fluid dynamics (CFD), using finite-difference approaches, and computational structural dynamics (CSD), using finite-element methods. In single disciplines, computations have been made on complete aircraft. However, only a limited amount of work has been completed in coupling these two disciplines for multidisciplinary applications. The prime reason is the lack of computational power for combining the two major computational fields. The development of a new generation of parallel computers can possibly alleviate the restriction of computational power.

A multidisciplinary code for computing unsteady flows and aeroelastic responses of aerospace vehicles, ENSAERO, has been developed on serial supercomputers at the Computational Aerosciences Branch of the NASA Ames Research Center (ref. 1). This multidisciplinary code computes unsteady aerodynamic responses of aircraft using the Euler/Navier-Stokes equations. The modal or finite element equations are used to model structures. An aeroelastic shape-conforming moving grid is used to include the effect of structural deformations on unsteady flows. This code is designed in a modular fashion to adopt several different numerical schemes suitable for accurate aeroelastic computations. The basic coding

*MCAT Institute.

of ENSAERO can accommodate zonal grid techniques for efficient modeling of full aircraft.

An early version of ENSAERO (ref. 2) has been successfully applied in computing aeroelastic responses of a rectangular wing by using the Euler equations for fluids and the modal equations for structures. The result demonstrates that the code can accurately predict the flutter dynamic pressure of a rectangular wing. The code was extended to compute aeroelastic responses using the Navier-Stokes equations for fluids (ref. 3). Later, it was updated by utilizing an upwind algorithm, and the code has been applied to fighter wings undergoing unsteady motions (refs. 4 and 5) at moderately large angles of attack. This code also has a capability of modeling moving control surfaces (ref. 6). Furthermore, ENSAERO has demonstrated the capability to simulate transonic flows on wing-body configurations using the Navier-Stokes equations (ref. 7).

For simple geometries such as clean wings, the modal approach can produce accurate response results. However, the modal approach may be less accurate for complex structures such as wing-body configurations. In order to accurately represent aeroelastic responses of general wing-body configurations, the finite element equations for structures have been implemented in ENSAERO. A typical wing-body configuration has been used to demonstrate aeroelastic responses at transonic Mach numbers using the Navier-Stokes equations for fluids and the finite element equations for structures (ref. 8). Two-noded beam elements are used to model the wing-body structures. Each node has three degrees of freedom (DOF) corresponding to transverse displacement and to transverse and torsional rotations, respectively.

In the past, all computations were accomplished serially, on computers such as the Cray Y-MP at Ames Research Center. Currently, a version of ENSAERO (ref. 9) that uses the Euler equations for fluids and the modal equations for structures has been parallelized on the Intel iPSC/860 at Ames. The Intel iPSC/860 is a distributed-memory, multiple-instruction, multiple-data (MIMD) computer with 128 processors. In this parallel implementation, a domain decomposition approach is used in which the fluid equations and the structural equations are modeled in separate computational domains. Each domain is mapped individually onto a group of processors, referred to as a cube on the Intel iPSC/860. As a result, each discipline can be developed and implemented independently of the others. However, because of the coupling between the disciplines, there is a need to exchange data, such as pressures and structural deformations at interfaces. This exchange between the fluid and structural domains is accomplished through an intercubes communication mechanism (ref. 10), which enables different processors in each cube to communicate directly.

In this work, procedures to compute aeroelastic responses on MIMD parallel computers using direct coupling of the Euler equations for fluids and the modal or finite element equations for structures are investigated for wings. The implementation of the structural domain on the iPSC/860 is described in detail. In addition, the computational efficiency issues of parallel integration of both fluid and structural

equations are investigated in detail. The proposed integration scheme exploits the architecture of MIMD computers. This research will provide an efficient procedure for aeroelastic analysis on MIMD computers.

This work was completed using the resources of the Numerical Aerodynamic Simulation (NAS) Program at NASA Ames Research Center. The work by C. Byun was supported by NASA Ames Research Center under Cooperative Agreement Number NCC2-740.

GOVERNING AERODYNAMIC EQUATIONS

The strong conservation-law form of the Euler equations is used for shock-capturing purposes. The Euler equations in generalized coordinates can be written as (ref. 11)

$$\partial_\tau \hat{Q} + \partial_\xi \hat{E} + \partial_\eta \hat{F} + \partial_\zeta \hat{G} = 0 \quad (1)$$

where \hat{Q} , \hat{E} , \hat{F} , and \hat{G} are flux vectors in generalized coordinates. The following transformations are used in deriving equation (1):

$$\begin{aligned} \tau &= t \\ \xi &= \xi(x, y, z, t) \\ \eta &= \eta(x, y, z, t) \\ \zeta &= \zeta(x, y, z, t) \end{aligned} \quad (2)$$

To solve equation (1), ENSAERO has time-accurate methods based on both central-difference and upwind schemes (ref. 12). In this work, the central-difference scheme based on the implicit approximate factorization algorithm of Beam and Warming (ref. 13) with modifications by Pulliam and Chaussee (ref. 14) for diagonalization was used. This scheme is first order accurate in time.

AEROELASTIC EQUATIONS OF MOTION

The governing aeroelastic equations of motion for structures can be written as

$$[M]\{\ddot{q}\} + [C]\{\dot{q}\} + [K]\{q\} = \{Z\} \quad (3)$$

where $[M]$, $[C]$, and $[K]$ are the global mass, damping, and stiffness matrices, respectively, and where $\{Z\}$ is the aerodynamic force vector corresponding to the displacement vector $\{q\}$. These quantities can be expressed in modal coordinates or finite element coordinates depending on the method used to obtain structural dynamic responses. One of the main efforts is concerned with the computation of the global force vector $\{Z\}$ of equation (3). In this work, for a given time t , $\{Z\}$ is computed by solving the Euler equations. From the solution of equation (1), pressure coefficients are computed at all grid points on the wing surface. Using

these pressure coefficients, the force vector $\{Z\}$ is calculated by means of the modal matrix or a fluid-structural interface, which is described in the next section.

The aeroelastic equations of motion (3) have been solved in past work (refs. 2 and 8) by a numerical integration technique based on the linear acceleration method (ref. 15). This method has been successfully used for integrating the modal equations, assuming a linear variation of the acceleration. However, this integration method requires a very small time step in order to integrate the finite element equations of motion. As a result, for the finite element equations of motion, the constant-average-acceleration method (ref. 16) is adopted to increase the time-step size. This method is an extension of the linear acceleration method. Assuming a constant average acceleration on a time interval, the velocities and displacements are obtained at a time t as

$$\begin{aligned}\{\dot{q}\}_t &= \{\dot{q}\}_{t-\Delta t} + \frac{\Delta t}{2} \{\ddot{q}\}_{t-\Delta t} + \frac{\Delta t}{2} \{\ddot{q}\}_t \\ \{q\}_t &= \{q\}_{t-\Delta t} + \Delta t \{\dot{q}\}_{t-\Delta t} + \frac{\Delta t^2}{4} (\{\ddot{q}\}_t + \{\ddot{q}\}_{t-\Delta t})\end{aligned}\quad (4)$$

Using these equations, the displacements at the end of a time interval can be obtained by solving

$$[D]\{q\}_t = \{Z\}_t + [M]\{a\} + [C]\{v\} \quad (5)$$

where

$$\begin{aligned}[D] &= [K] + \frac{4}{\Delta t^2}[M] + \frac{4}{\Delta t}[C] \\ \{a\} &= \frac{4}{\Delta t^2}\{q\}_{t-\Delta t} + \frac{4}{\Delta t}\{\dot{q}\}_{t-\Delta t} + \{\ddot{q}\}_{t-\Delta t} \\ \{v\} &= \frac{2}{\Delta t}\{q\}_{t-\Delta t} + \{\dot{q}\}_{t-\Delta t}\end{aligned}$$

This is an unconditionally stable scheme, whereas the linear acceleration method is conditionally stable.

FLUID-STRUCTURAL INTERFACES

In aeroelastic analysis, it is necessary to represent the equivalent aerodynamic loads at the structural nodal points and to represent the deformed structural configurations at the aerodynamic grid points. In the present domain decomposition approach, coupling between the fluid and structural domains is achieved by interfacing the boundary data, such as aerodynamic pressures and structural deflections, at each time step. An analytical moving-grid technique has been successfully used to deform the aerodynamic grid according to the structural deflections at the end of every time step (refs. 1, 2, and 8). There are different approaches for obtaining the global force vector $\{Z\}$ of equation (3), depending on the equations used for the structural dynamic analysis.

For the modal equations of motion, the global force vector can be easily obtained in terms of the preselected mode shapes (modal matrix) as

$$\{Z\} = \frac{1}{2}\rho U_\infty^2 [\Phi]^T [A] \{\Delta C_p\} \quad (6)$$

where $[\Phi]$ is the modal matrix and $[A]$ the diagonal area matrix of the aerodynamic control points. The unsteady differential pressure coefficients on the wing surface are defined as $\{\Delta C_p\}$. It is noted that the modal matrix is also used to represent the deformation of the wing.

Solution of the equations of motion based on the finite element discretization requires a fluid-structural interface similar to the modal matrix. Several numerical procedures have been developed for exchanging the necessary information between the fluid and structural domains (refs. 17–19). However, in this study, a linear interpolation scheme is first developed for the interface so that coupling of fluid and structural equations could be simple for implementation on the new parallel computers. This scheme is called the lumped load interface. In this method, the force acting on each element of the structural mesh is first calculated and then the element nodal force vector is obtained by distributing the force. The global force vector is obtained by assembling the nodal force vectors of each element. In addition, the deformed configuration of the CFD grid at the surface is obtained by linearly interpolating nodal displacements at finite element nodes.

Next, a mapping matrix developed by Appa (ref. 18) is selected to accurately exchange data between the fluid and structural interface boundaries. The reason for selecting Appa's method is that the mapping matrix is general enough to accommodate changes in fluid and structural models easily. In addition, this approach conserves the work done by aerodynamic forces when obtaining the global nodal force vector. This method introduces a virtual surface between the CFD surface grid and the finite element mesh for the wing. The virtual surface is discretized by a number of finite elements, which are not necessarily the same elements used in the structural surface modeling. This method is called the virtual surface interface.

By forcing the deformed virtual surface to pass through the given data points of the deformed structure, a mapping matrix relating displacements at structural and aerodynamic grid points is derived as

$$[T] = [\psi_a] (\delta^{-1}[K] + [\psi_s]^T [\psi_s])^{-1} [\psi_s]^T \quad (7)$$

where

$[K]$ is the free-free stiffness of the virtual surface

ψ_s is the displacement mapping from virtual to structural grids

ψ_a is the displacement mapping from virtual to aerodynamic grids

δ is the penalty parameter

Then, the displacement vector at the aerodynamic grid $\{q_a\}$ can be expressed in terms of the displacement vector at the structural nodal points q_s as

$$\{q_a\} = [T] \{q_s\}$$

From the principle of virtual work, the nodal force vector $\{Z_s\}$ can be obtained as

$$\{Z_s\} = [T]^T \{Z_a\}$$

where $\{Z_a\}$ is the force vector at the aerodynamic grids (ref. 20).

PARALLEL IMPLEMENTATION OF THE AEROELASTIC EQUATIONS

The domain decomposition approach used in this study enables data structures and solution methods for fluid and structural equations to be developed independently. Fluid and structural equations are modeled in separate computational domains. However, coupling of the disciplines requires the exchange of the interface boundary data, which is accomplished through an intercube communication mechanism (ref. 10). This intercube communication facility enables different processors in each cube to communicate directly on the iPSC/860. It is important to keep the specification of data exchange routine the same on both computational domains.

The domain for fluids is capable of solving the Euler equations using 3-D uni-partitioning of the computational domain. The uni-partitioning scheme denotes that one grid subdomain is assigned to each of the processors. The arrangement of processors is described in figure 1. The arrows denote bi-directional data communication. There are a variety of concurrent algorithms available for solving the system of equations for fluids. Currently, the solver for the fluid equations can use three different concurrent algorithms: (1) complete exchange-based implementations (CE-GE), (2) pipelined-Gaussian elimination (PGE), and (3) substructured Gaussian elimination followed by solution of the reduced system by means of balanced odd-even cyclic reduction (SGE-BCR) (ref. 21). In this work, the one-way PGE scheme is used. The choice of algorithms was made largely on the basis of memory use. The one-way PGE method allows the use of larger computational grids, or of fewer processors, than do the other schemes. More details about the implementation of the fluid domain can be found in reference 21.

For the structural domain, modal equations were first used on the iPSC/860. Since a limited number of preselected mode shapes were used, only a single processor was assigned to the structural domain. However, in replacing modal equations with the finite element equations, it is necessary to use a cube of multiple processors for structures. In this study, it is assumed that each subdomain of the entire structure is mapped onto a single processor. Each processor stores only the information relevant to the subdomain assigned to it. The information can be the stiffness and

mass matrices and the applied nodal force vector of a subdomain. Then equation (5) is expressed at the subdomain level.

In this work, a regular finite element mesh is used to model wings as a plate, and the domain decomposition is made by using 2-D uni-partitioning as shown in figure 1. This type of domain decomposition enables an efficient and simple message communication mechanism within the structural domain. Only chordwise and spanwise bi-directional messages are exchanged along the subdomain interfaces in the structural domain.

At present, the solver for the structural domain is based on a Jacobi-preconditioned conjugate gradient (JPCG) algorithm on the Intel iPSC/860. The present JPCG algorithm is based on a parallel conjugate gradient algorithm proposed by Law (ref. 22). The algorithm is described in the appendix for completeness. The advantage of Law's algorithm is that it does not form the global system matrices. In this method, the multiplication of a matrix by a trial vector, which is the major operation of the conjugate gradient algorithm, is performed at the subdomain level. The interprocessor communication is confined to the solution phase.

INTEGRATION SCHEMES FOR COUPLED DOMAINS

In a serial computer, the integrations of both fluid and structural equations are performed one after the other in a sequential nature. Figure 2(a) shows the sequential integration scheme implemented on MIMD computers. In the sequential integration scheme, the fluid domain has to wait to proceed to the next time step until it receives information about structural deformations. The structural domain also has to wait for surface pressure data, so both cubes have their own idle times while they wait for data communications. However, since the size of the structural equations was small for modal analysis, the effect of the waiting time was negligible relative to the computational time per integration step.

On the contrary, if a large number of modes or direct finite element equations are used in order to accurately predict dynamic responses of complex structures, the computational time per integration step may be increased rapidly. This is due to the increase in the idle time when a sequential integration scheme is used on the iPSC/860. However, this situation can be avoided by executing the integration of both fluid and structural equations concurrently as shown in figure 2(b). In the proposed parallel integration scheme, both solvers start computations independently and one of the solvers waits until the other finishes its calculation. Then they exchange the required data with each other for the next time step. By doing so, the parallel integration can reduce the idle time since only one cube may have partial idle time. The resulting speedup achieved by the parallel integration scheme is theoretically by a factor of almost 2, provided that computational times required for the fluid and structural domains are well balanced.

COMPUTATIONAL RESULTS

To demonstrate an aeroelastic computation, a clipped delta wing of aspect ratio 3 and taper ratio 1/7 with the NACA 65A006 airfoil section was selected. The sweep angle at the quarter chord line ($\Lambda_{c/4}$) is 45° . The transonic flutter characteristics of this wing are available from wind tunnel tests (ref. 23) for various flow parameters.

In this computation, the flow field is discretized by using a C-H grid topology of size $151 \times 30 \times 25$. The CFD grid at the root and the upper surface of the wing is shown in figure 3. The CFD grid is assigned to 32 processors on the iPSC/860. The processors are arranged as a three-dimensional mesh of eight processors in the chordwise direction and two processors in each of the spanwise and surface-normal directions. This same arrangement for fluids is kept throughout the computations so that the performance of the structural domain can be studied in detail.

A 20-DOF ANS4 shell element (ref. 24) was used for the finite element modeling of the wing structure. The wing is modeled as a plate. Considering the wing structure used in the experiment, variation of mass density is allowed along the chordwise and spanwise directions. But the thickness of the finite element model is kept constant. This is based on assumptions that the stiffness of the wing is dominated by the aluminum-alloy insert and that mass distribution of the wing is significantly changed as a result of plastic foams covering the aluminum-alloy insert. This finite element plate model predicts natural vibration modes of the wing that compare well with the experiment. The first three modal frequencies computed by using the finite element model are 21.8, 78.1, and 126 Hz, and corresponding values measured in the experiment are 21.6, 79.7, and 121 Hz, respectively.

Modal Analysis

The first parallel version of ENSAERO was capable of using the Euler equations for fluids and the modal equations for structures. Using this version of ENSAERO, aeroelastic responses were computed. In figure 4, the computed generalized displacements of the first three modes for the wing are presented. The results were obtained for a freestream Mach number (M_∞) of 0.854 and a given dynamic pressure (P) of 0.7 psi. In this calculation, the first six mode shapes of the wing were used to predict the structural dynamic responses. Identical results were obtained from serial and parallel computers. At this point, it is verified that the fluid and structural domains of the parallel ENSAERO and of the intercubes communication mechanism are properly working. It should be noted that only one processor is assigned for the structural analysis since only six mode shapes are used to represent the structural properties of the wing. The wall-clock times required for each integration step on a single processor of the Cray Y-MP and on the Intel iPSC/860 are 1.36 and 3.03 seconds, respectively.

The proposed parallel integration scheme is compared with the sequential integration scheme used on serial computers. Using both sequential and parallel integration schemes, aeroelastic responses of the wing were computed on the iPSC/860. Aeroelastic responses were computed by simulating experimental conditions for a freestream Mach number of 0.977 and for a given dynamic pressure of 0.65 psi. Results from sequential and parallel integration schemes agree well, as shown in figure 5. Since only six modes were used for the modal analysis, the reduction in computational time was marginal (less than 2% of the time per integration step used in the sequential integration scheme). By increasing the number of modes to 50, the sequential integration scheme required 5% more computational time per integration step whereas the time for the parallel integration scheme remained the same. This trend is more evident as the number of equations increases.

Finite Element Analysis

For simple geometries such as rectangular wings, the modal analysis can accurately predict dynamic responses. However, the modal analysis with a limited number of preselected mode shapes may be less accurate for complex structures such as wing-body configurations. In order to accurately represent aeroelastic responses of general aircraft configurations, the modal analysis was replaced with the direct finite element analysis in ENSAERO. The finite element equations were first tested on the Cray Y-MP version of the code and then were parallelized on the Intel iPSC/860.

The lumped load and virtual surface interfaces on the Y-MP are shown in figures 6 and 7. The wing structure was modeled using 100 ANS4 elements. Ten elements each were assigned along the chordwise and spanwise directions, respectively. The time history of total lift on the wing for a given dynamic pressure of 1.0 psi and initial acceleration of 1.0×10^5 inches/sec², is presented in figure 6. The exact solution is the total lift obtained by integrating pressure coefficients at CFD grid points. Both virtual surface (VS) and lumped load (LL) interfaces obtain the total lift by summing the forces at the finite element (FE) nodal points, which were transformed from pressure coefficients through interfaces. The virtual surface interface transfers pressure data more accurately than the lumped load interface. The lumped load interface shows a favorable result although the response around peaks deviates from the exact solution. In addition, the tip displacements of the wing at the leading edge are presented in figure 7. The lumped load approach shows favorable agreement with the virtual surface approach.

The lumped load approach was first used as the fluid-structural interface for the finite element equations on the iPSC/860. The choice of interfaces was made largely on the basis of memory use. The size of the mapping matrix for the virtual surface interface becomes too large to fit on a single processor on the iPSC/860 when the number of fluid grid points or structural nodal points on the wing increases. However, the lumped load approach requires only a small amount of memory to identify the location of fluid grid points on a finite element discretization.

In order to check the parallel implementation of the finite element equations, the tip displacements of the wing were computed, without aerodynamic forces, on the Y-MP and the iPSC/860. This wing was modeled using 64 ANS4 elements. For this computation, processors were assigned as a 2-D mesh of two processors in the chordwise and spanwise directions, respectively, on the iPSC/860. Identical results were obtained on both the Y-MP and the iPSC/860.

For the structural model using finite element equations, the aeroelastic responses were obtained using both sequential and parallel integration schemes on the iPSC/860. The results are presented in figure 8. The responses were obtained for a given dynamic pressure of 1.0 psi. The same finite element mesh and processor arrangement used in the previous case are used for this computation. The two results agree well. The wall-clock times per integration step achieved are 3.45 and 3.00 seconds by using sequential and parallel integration schemes, respectively. The speedup is still marginal since the total number of equations is relatively small (360 DOF) for the structural dynamic analysis. However, for 256 finite elements (1360 DOF) with four processors on the structural domain, the wall-clock times per integration step are 6.22 and 3.33 seconds by using sequential and parallel integration schemes, respectively. The speedup achieved is 1.87 by using the parallel integration scheme. When the computational time between the fluid and structural domains is balanced, maximum speedup can be achieved.

The parallel integration scheme enables the combination of advanced CFD and CSD technologies with minimal increase in the computational time per integration step. The required computational time per integration step is determined by both the fluid and structural domains on serial computers. However, using the parallel integration scheme on MIMD computers, the time is solely determined by the computational domain that requires more time per integration step. This parallel integration is one of the advantages of using MIMD computers for multidisciplinary analysis.

Aeroelastic responses were also computed for various dynamic pressures in order to predict the flutter dynamic pressure. Figure 9 shows the stable, near neutrally stable, and unstable responses of wing tip displacements at the leading edge for dynamic pressures of 0.80, 0.85, and 0.90 psi, respectively. From the responses shown in figure 9, the interpolated dynamic pressure for the neutrally stable condition is 0.84 psi. It is noted that the experimental dynamic pressure measured at the neutrally stable condition was 0.91 psi. Considering the lack of experimental pressure data on the wing and the error involved in modeling the wing as a plate with constant thickness, the computational result is an acceptable prediction of the flutter dynamic pressure.

Performance

In order to support multidisciplinary analysis with practical computational turnaround times for design work, the computational domain on parallel machines must be scalable. This section describes several aspects of performance, including

single processor computational rates, domain decomposition strategy, and scalability of the structural domain in ENSAERO on the iPSC/860. It is noted that only a linear model is used for the dynamic analysis of the structural domain. The performance of the fluid domain can be found in reference 21. All performance data reported are for 64-bit arithmetic.

In order to measure the performance of the structural domain on the Intel iPSC/860, the identical code, except message passing routines, was run on the Y-MP and the averaged time per integration step was obtained. All FLOP rates quoted are calculated by comparing the time per integration step on the iPSC/860 with that on the Y-MP using a single processor. Operation counts from the Cray Hardware Performance Monitor are used.

A single processor of the iPSC/860 was able to house 256 ANS4 elements. The Y-MP equivalent MFLOPS obtained is about 4.2 MFLOPS for this size of problem, and the corresponding rate is about 77 MFLOPS on a single Y-MP processor. This rate is about 7% of the peak performance of a single processor on the iPSC/860. Similar performance was reported by Ryan and Weeratunga for the fluid domain (ref. 21).

The performance of the structural domain in parallel ENSAERO has been measured over a wide range of processor numbers and problem sizes as shown in figure 10. The speedup relative to the Y-MP is defined as

$$speedup = \frac{t_{Cray}}{t_{Intel}}$$

where t_{Cray} and t_{Intel} are the computational time per integration step measured on the Y-MP and the iPSC/860, respectively. Only a single processor is used to measure t_{Cray} on the Y-MP. The open and filled symbols denote the domain decomposition which results in the minimum and maximum bandwidths of the stiffness matrix of each subdomain for a given number of processors.

For the case of 1,360 DOF, the computational time per integration step on the iPSC/860 is barely closed to that on the Y-MP when 64 processors are in use. However, as increasing the size of problem (10,560 and 20,800 DOF), the iPSC/860 achieves about the speed of the Y-MP by using 16 processors. It is evident that the JPCG solver on the iPSC/860 performs better as the size of problem increases. For the case of 20,800 DOF, the relative speedup achieved is about 8 by the time 64 processors are in use.

For a given number of processor, the obtained speedup varies depending on the domain decomposition strategy as shown in figure 10. Only the results for the minimum and maximum bandwidths are presented for clarity. The speedup increases as decreasing the matrix bandwidth of each subdomain for a given number of processors on the iPSC/860. This is due to the fact that the conjugate gradient algorithm is subjected to the multiplication of the coefficient matrix and a trial vector. Since the multiplication is performed only at the subdomain level in the JPCG

solver, the smaller bandwidth results in fewer operations and quicker computational time.

The overall performance of ENSAERO on both the Y-MP and the iPSC/860 is shown in figure 11 for the case of 113,250 grid points for the fluids and 10,560 DOF for the structure. In this computation, 32 processors are assigned to the fluid domain and 16 to 64 processors to the structural domain. Both the skyline reduction and JPCG solvers are compared on the Y-MP but only the JPCG solver is used for the structural domain on the iPSC/860 at the present time. A parallel version of the skyline reduction solver is under implementation.

The height of each column is the time per integration step. Each column is divided into zones representing the time spent for the fluid domain, the structural domain, and for idle/intercube communication. It should be noted that the time per integration step for the skyline reduction solver included only time spent for forward reduction and back substitution without the factorization time. The reason is that the contribution of the factorization time to the computation time per integration step is negligible when a large number of time steps are required to obtain linear dynamic responses. Providing that 7,000 steps are required for a typical aeroelastic computation of the given wing, the increase of the time per integration step is about 0.5% of the time used for structure.

It is evident that the skyline reduction solver outperforms the JPCG solver on the Y-MP. However, the JPCG solver is first implemented on the iPSC/860. The reason is that it is desirable to compare the performance of the two solvers on the iPSC/860. The JPCG solver on the iPSC/860 could not achieve the performance of the skyline reduction solver without including the factorization time on the Y-MP. However, as far as the JPCG solver is concerned, 16 processors on the iPSC/860 can obtain the performance of a single processor on the Y-MP. In addition, the overall performance of ENSAERO using 96 processors on the iPSC/860 is about one-third of that obtained using the skyline reduction solver with a single processor on the Y-MP. This result is based on the averaged time per integration step. It should be noted that the structural domain determined the time per integration step for this particular problem on the iPSC/860. Most of the time on the fluid domain was spent waiting for the interface boundary data. This means that fewer processors can be assigned to the fluid domain without sacrificing computational performance as long as the memory on each processor can accommodate the assigned grid partitioning.

CONCLUSIONS

A parallel version of a multidisciplinary code, ENSAERO, was developed on the Intel iPSC/860. A domain decomposition approach was used to enable the fluid and structural domains to be developed and maintained independently. This approach provides an efficient and effective environment to researchers. A researcher concerned with the fluid or the structural domain can develop his own discipline independent of the others. The only thing to be done together is coupling

of the disciplines. Since coupling of the disciplines is achieved by an intercube communication mechanism, coupling should not cause any problem as long as each domain maintains the specification for intercube communication. This makes it easy for each discipline to incorporate and develop new algorithms or data structures without interferences.

In addition to the modal analysis, the capability of finite element analysis for structural dynamic analysis has been added to parallel ENSAERO. The structural domain on the iPSC/860 is again divided into a number of subdomains. The solution of the structural domain is obtained by the Jacobi preconditioned conjugate gradient (JPCG) solver. The partition of the structural domain for a wing is two-dimensional uni-partitioning since the wing is modeled as a plate and the discretization is a regular mesh. This enables the message communication within the structural domain to be very simple and efficient.

As far as the structural domain is concerned, the performance on the iPSC/860 is still far behind the best performance of the Y-MP a result of the poor performance of the JPCG algorithm. A parallel version of the skyline reduction solver is being implemented for the structural domain on the iPSC/860. This will provide increased performance for the structural domain on the iPSC/860. For a problem size of 10,560 DOF, the overall performance of parallel ENSAERO using 96 processors on the iPSC/860 is about one-third of that obtained using the skyline reduction solver for the structural domain on a single Y-MP processor.

The parallel integration scheme enables the combination of advanced CFD and CSD technologies with minimal increase in computational time per integration step. The computational time per integration step is solely determined by the domain that requires more computational time on the iPSC/860 whereas that time it is determined by both domains on serial computers. This parallel integration is one of the advantages of using MIMD computers for multidisciplinary analysis. The procedure developed in this research will provide an efficient tool for solving aeroelastic problems of complete aerospace vehicle configurations on MIMD computers.

APPENDIX

1. Set diagonal elements of global stiffness matrix
 - Send $\{K_{j,j}^e\}$ to neighboring processors
 - Receive $\{K_{j,j}^i\}$ from neighboring processors
 - $\{d^e\} = \Sigma \{K_{j,j}^i\} + \{K_{j,j}^e\}$
2. Set initial trial and residual vectors
 - $\{q^e\} = 0$
 - $\{r^e\} = \{z^e\}$
3. Set initial search direction
 - Send $\{r^e\}$ to neighboring processors
 - Receive $\{r^e\}$ from neighboring processors
 - $\{s^e\} = \Sigma\{r^i\} + \{r^e\}$
 - $z_j^e = s_j^e/d_j^e, \quad j = 1, \dots, neq^e$
 - $\rho^e = \{r^e\}^T \{z^e\}$
 - $\gamma_0 = \Sigma \rho^e, \quad e = 1, \dots, np$ (global sum)
 - $\gamma = \gamma_0$
 - $\{p^e\} = \{s^e\}$
4. Operations at subdomain level
 - $\{u^e\} = [K^e]\{p^e\}$
 - $\beta^e = \{p^e\}^T \{u^e\}$
 - $\sigma^e = \beta^e/\gamma$
5. Update solution and residual
 - $1/\alpha = \Sigma \sigma^e, \quad e = 1, \dots, np$ (global sum)
 - $\{q^e\} = \{q^e\} + \alpha\{p^e\}$
 - $\{r^e\} = \{r^e\} - \alpha\{u^e\}$
6. Update search direction and check convergence
 - Send $\{r^e\}$ to neighboring processors
 - Receive $\{r^e\}$ from neighboring processors
 - $\{s^e\} = \Sigma\{r^i\} + \{r^e\}$
 - $z_j^e = s_j^e/d_j^e, \quad j = 1, \dots, neq^e$
 - $\rho^e = \{r^e\}^T \{z^e\}$
 - $\gamma_{new} = \Sigma \rho^e, \quad e = 1, \dots, np$ (global sum)
 - IF*($\gamma_{new}/\gamma_0 < TOLERANCE$) *STOP*
 - $\{p^e\} = \{s^e\} + (\gamma_{new}/\gamma) \{p^e\}$
 - $\gamma = \gamma_{new}$
7. Repeat 4 to 6 until converged

REFERENCES

1. Guruswamy, G. P.: ENSAERO—A Multidisciplinary Program for Fluid/Structural Interaction Studies of Aerospace Vehicles. *Computing Systems Engineering*, vol. 1, 1990, pp. 237–256.
2. Guruswamy, G. P.: Unsteady Aerodynamic and Aeroelastic Calculations for Wings Using Euler Equations. *AIAA J.*, vol. 28, 1990, pp. 461–469.
3. Guruswamy, G. P.: Navier-Stokes Computations on Swept-Tapered Wings, Including Flexibility. *AIAA Paper 90-1152*, Apr. 1990.
4. Obayashi, S.; Guruswamy, G. P.; and Goorjian, P. M.: Application of Streamwise Upwind Algorithm for Unsteady Transonic Computations over Oscillating Wings. *AIAA Paper 90-3103*, Aug. 1990.
5. Obayashi, S.; and Guruswamy, G. P.: Unsteady Shock-Vortex Interaction on a Flexible Delta Wing. *AIAA Paper 91-1109*, Apr. 1991.
6. Obayashi, S.; and Guruswamy, G. P.: Navier-Stokes Computations for Oscillating Control Surface. *AIAA Paper 92-4431*, Aug. 1992.
7. Obayashi, S.; Guruswamy, G. P.; and Tu, E. L.: Unsteady Navier-Stokes Computations on a Wing-Body Configuration in Ramp Motions. *AIAA Paper 91-2865*, Aug. 1991.
8. Guruswamy, G. P.: Coupled Finite-Difference/Finite-Element Approach for Wing-Body Aeroelasticity. *AIAA Paper 92-4680*, Sep. 1992.
9. Guruswamy, G. P.; Weeratunga, S.; and Pramono, E.: Direct Coupling Approach to Fluids/Structures Interaction Problems for Parallel Computing. *NASA CAS Conference, Moffett Field, CA 94035-1000*, Aug. 18-20, 1992, pp. 46–47.
10. Barszcz, E.: Intercube Communication on the iPSC/860. *Proceedings of the Scalable High Performance Computing Conference, Williamsburg, VA, April 26-29, 1992*, pp. 307–313.
11. Peyret, R.; and Viviand, H.: *Computation of Viscous Compressible Flows Based on Navier-Stokes Equations*. AGARD AG-212, 1975.
12. Obayashi, S.; Guruswamy, G. P.; and Goorjian, P. M.: Streamwise Upwind Algorithm for Computing Unsteady Transonic Flows Past Oscillating Wings. *AIAA J.*, vol. 29, 1991, pp. 1668–1677.

13. Beam, R.; and Warming, R. F.: An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form. *J. Computational Physics*, vol. 22, 1976, pp. 87–110.
14. Pulliam, T. H.; and Chaussee, D. S.: A Diagonal Form of an Implicit Approximate Factorization Algorithm. *J. Computational Physics*, vol. 39, no. 2, 1981, pp. 347–363.
15. Guruswamy, P.; and Yang, T. Y.: Aeroelastic Time-Response Analysis of Thin Airfoils by Transonic Code LTRAN2. *Computers and Fluids*, vol. 9, 1980, pp. 409–425.
16. Newmark, N. M.: A Method of Computation for Structural Dynamics. A.S.C.E., *J. Engineering Mechanics Division*, vol. 85, 1959, pp. 67–94.
17. Harder, R. L.; and Desmarais, R. N.: Interpolation Using Surface Spline. *J. Aircraft*, vol. 9, 1972, pp. 189–191.
18. Appa, K.: Finite-Surface Spline. *J. Aircraft*, vol. 26, 1989, pp. 495–496.
19. Pidaparti, R. M. V.: Structural and Aerodynamic Data Transformation Using Inverse Isoparametric Mapping. *J. Aircraft*, vol. 29, 1992, pp. 507–509.
20. Appa, K.; Yankulich, M.; and Cowan, D. L.: The Determination of Load and Slope Transformation Matrices for Aeroelastic Analyses. *J. Aircraft*, vol. 22, 1985, pp. 734–736.
21. Ryan, J. S.; and Weeratunga, S. K.: Parallel Computation of 3-D Navier-Stokes Flowfields for Supersonic Vehicles. *AIAA Paper 93-0064*, Jan. 1993.
22. Law, K. H.: A Parallel Finite Element Solution Method. *Computers and Structure*, vol. 23, 1986, pp. 845–858.
23. Dogget, R. V.; Rainey, A. G.; and Morgan, H. G.: An Experimental Investigation of Aerodynamics Effects of Airfoil Thickness on Transonic Flutter Characteristics. *NASA TM X-79*, 1959.
24. Park, K. C.; Pramono, E.; Stanley, G. M.; and Cabiness, H. A.: The ANS Shell Elements: Earlier Developments and Recent Improvements. In *Analytical and Computational Models of Shells*, A. K. Noor (ed.), CED–Vol. 3, ASME, New York, 1989, pp. 217–240.
25. Bathe, K.-J.: *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982, pp. 702–706.

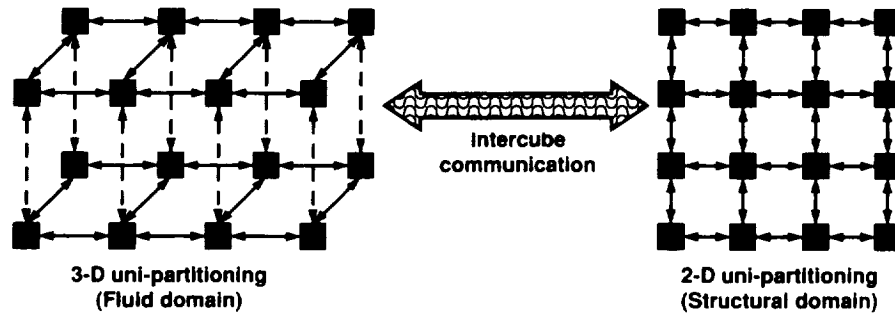
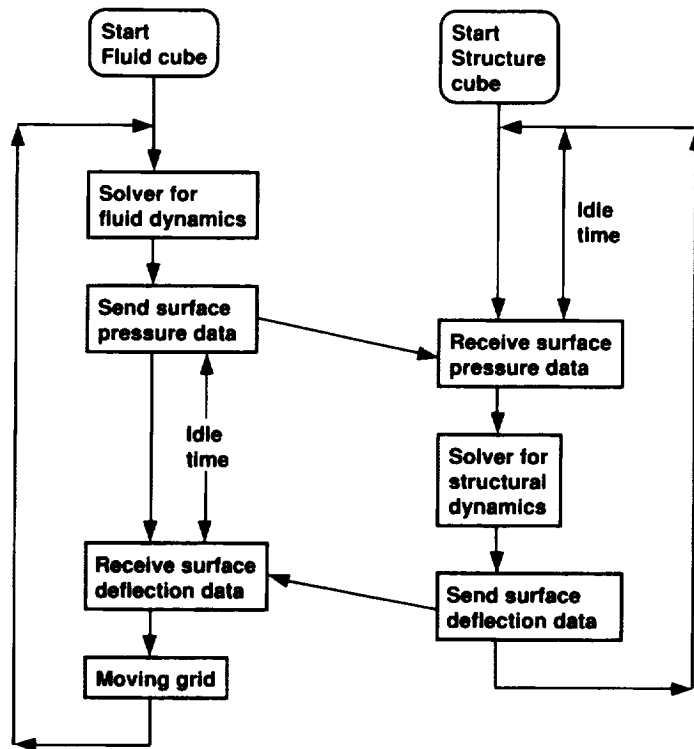
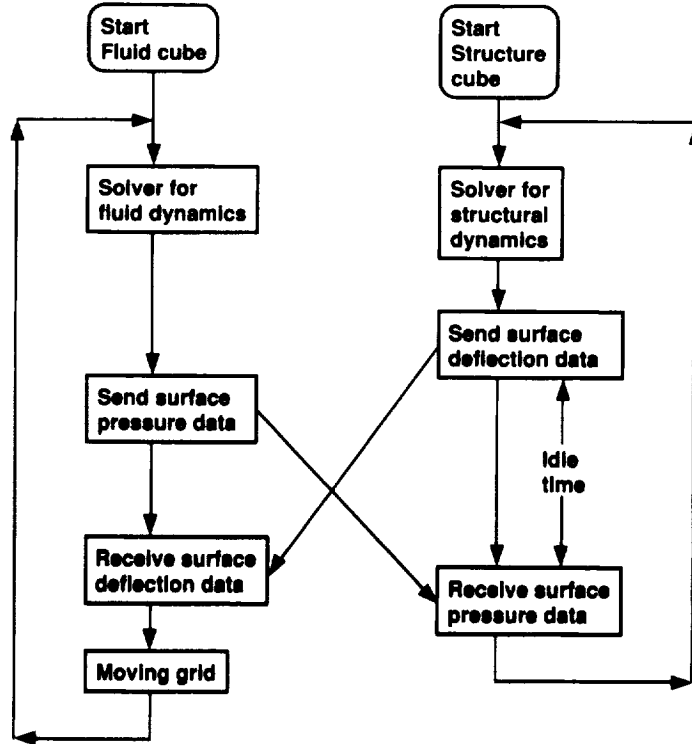


Figure 1. Processor arrangements and message exchanges through interprocessor and intercube communications.



(a) Sequential integration



(b) Parallel integration

Figure 2. Flow diagrams for sequential and parallel integration schemes.

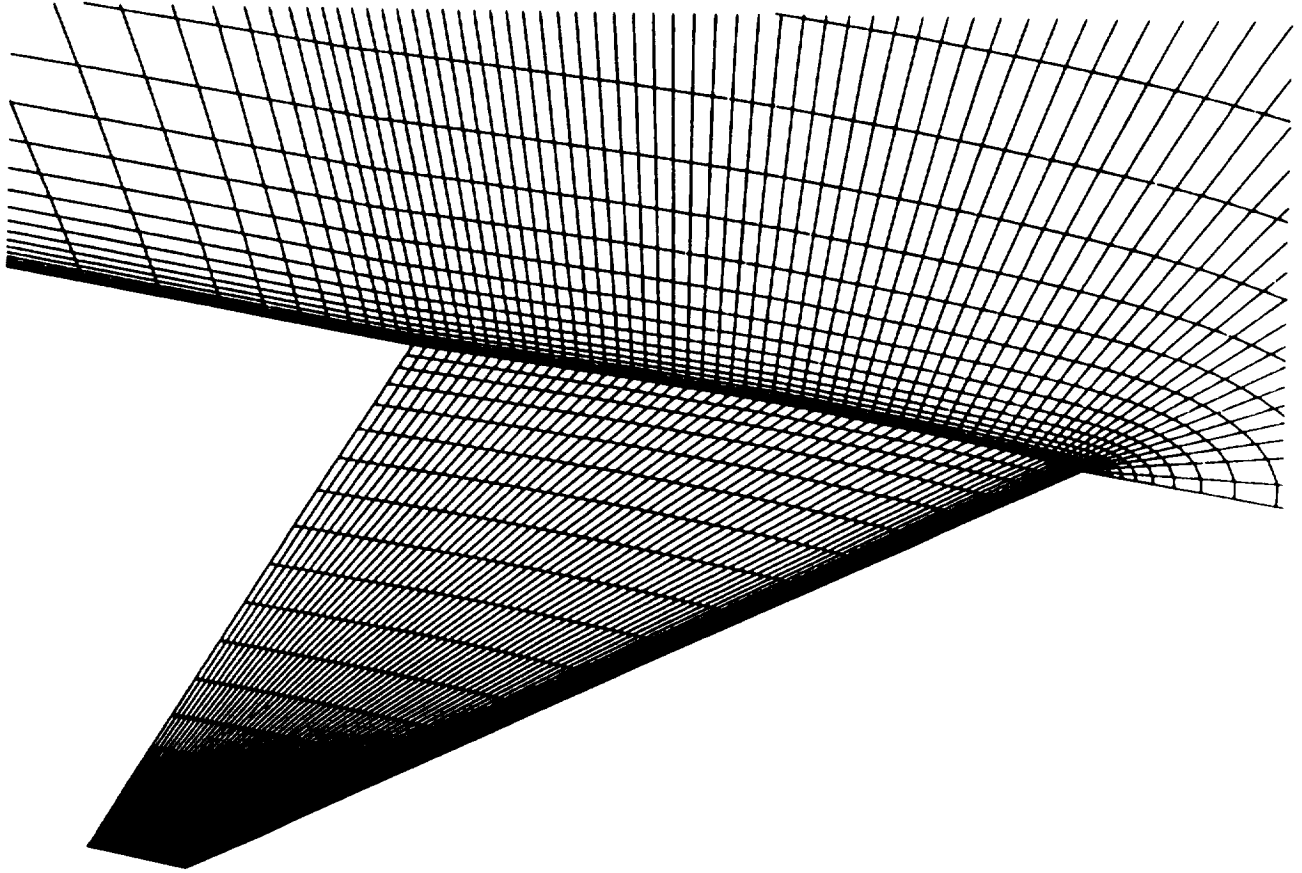


Figure 3. The CFD grid at the root and surface of a clipped delta wing with the NACA 65A006 airfoil section. (Aspect ratio = 3.0, Taper ratio = 1/7, $\Lambda_{c/4} = 45^\circ$)

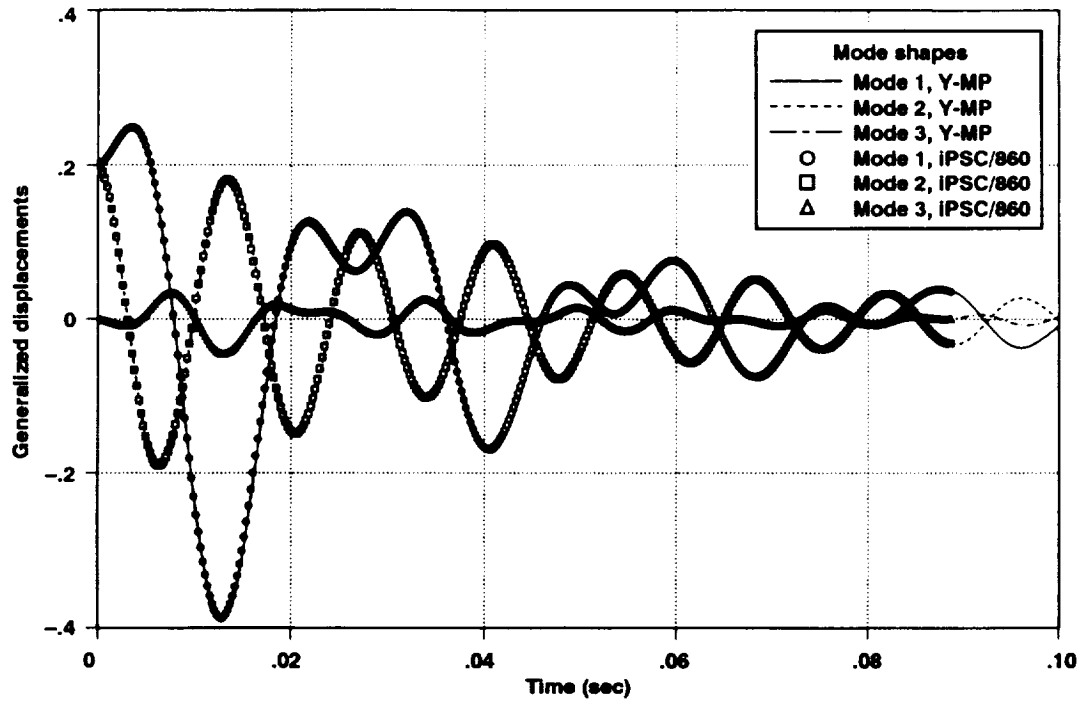


Figure 4. Comparison among the first three generalized displacement histories obtained by the serial and parallel ENSAERO codes. ($M_\infty = 0.854$, $P = 0.70$ psi)

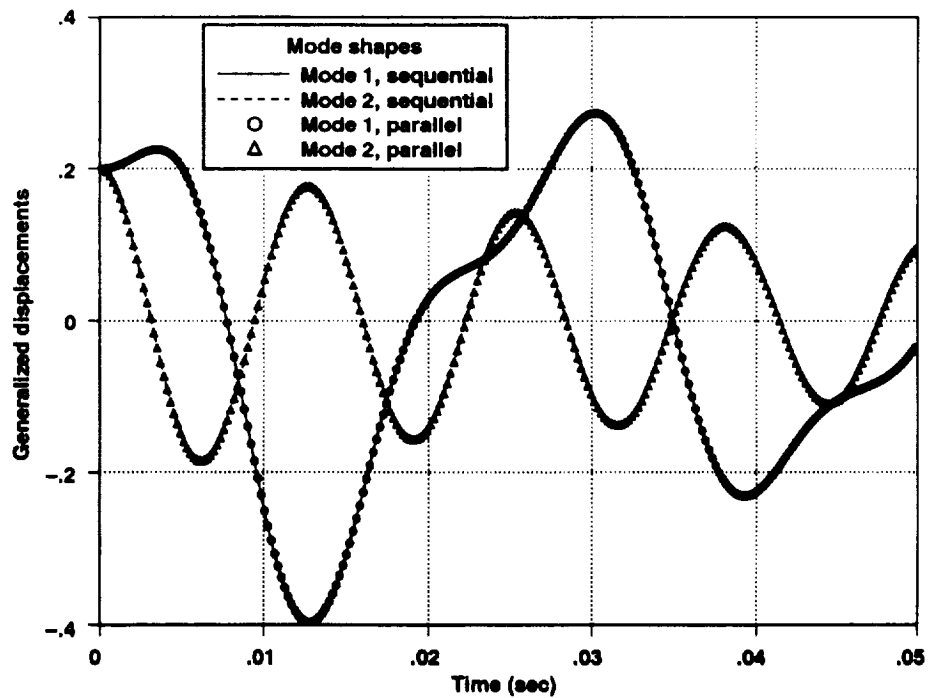


Figure 5. Generalized displacement histories obtained from sequential and parallel integration schemes. ($M_\infty = 0.977$, $P = 0.65$ psi)

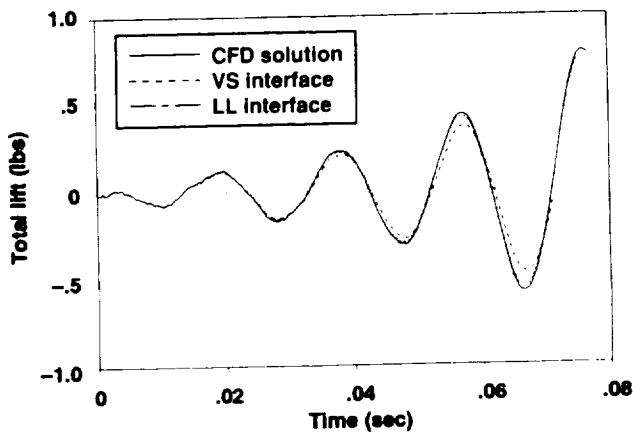


Figure 6. Comparison of total lift responses. ($M_\infty = 0.854$, $\alpha = 0$ deg., $P = 1.0$ psi)

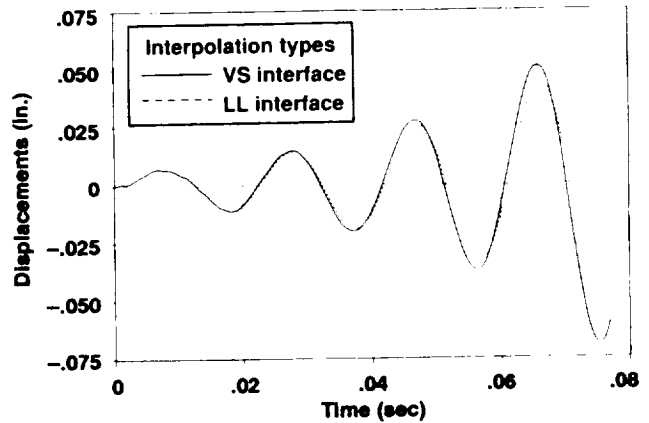


Figure 7. Comparison of wing tip displacements at the leading edge obtained by using virtual surface and lumped load interfaces. ($M_\infty = 0.854$, $\alpha = 0$ deg., $P = 1.0$ psi)

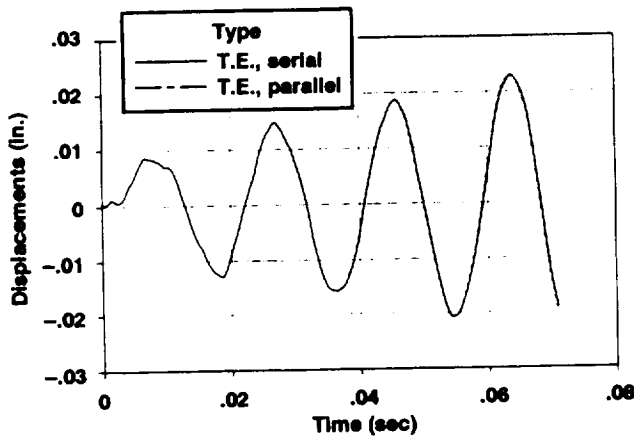


Figure 8. Aeroelastic responses obtained by using sequential and parallel integration schemes. ($M_\infty = 0.854$, $\alpha = 0$ deg., $P = 1.0$ psi)

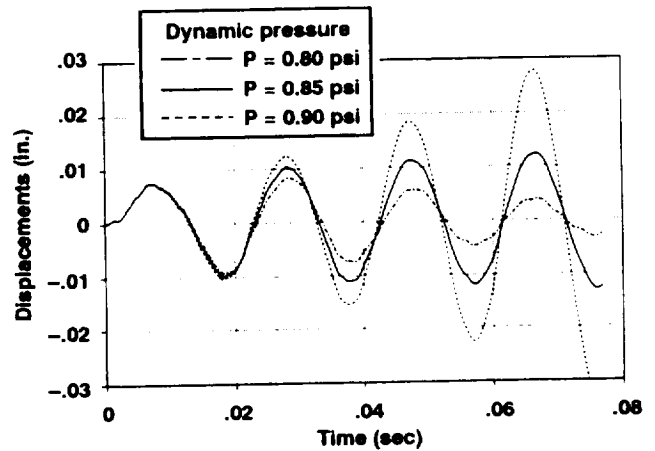


Figure 9. Aeroelastic responses of a clipped delta wing at three different dynamic pressures using Euler equations for the fluid and finite element equations for the structural domains. ($M_\infty = 0.854$, $\alpha = 0$ deg.)

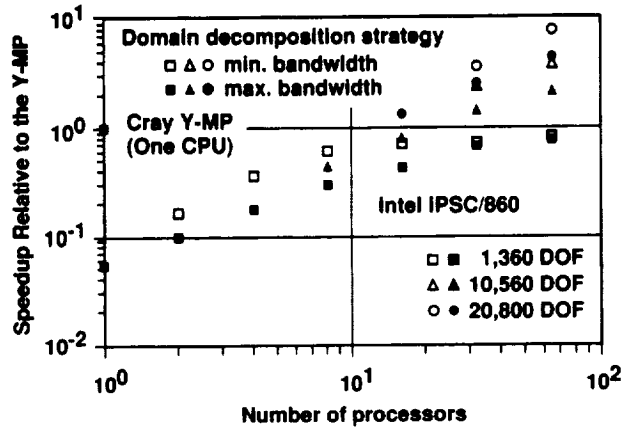


Figure 10. Computational performance of the structural domain in ENSAERO with various problem sizes and domain decompositions on the Intel iPSC/860.

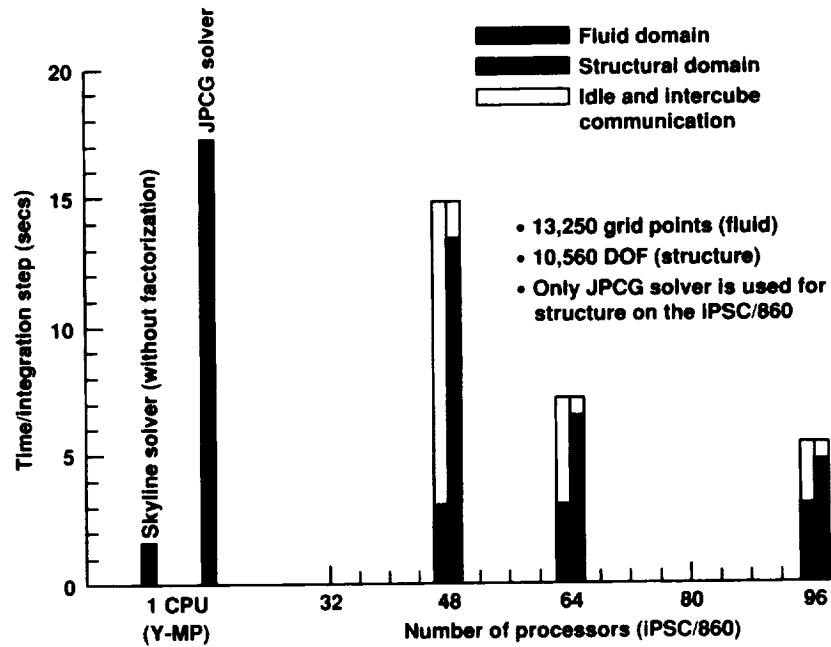


Figure 11. Overall computational performance of ENSAERO on the Cray Y-MP and the Intel iPSC/860.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE January 1994	3. REPORT TYPE AND DATES COVERED Technical Memorandum	
4. TITLE AND SUBTITLE A Comparative Study of Serial and Parallel Aeroelastic Computations of Wings		5. FUNDING NUMBERS 505-10-11	
6. AUTHOR(S) Chansup Byun and Guru P. Guruswamy			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Ames Research Center Moffett Field, CA 94035-1000		8. PERFORMING ORGANIZATION REPORT NUMBER A-94039	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001		10. SPONSORING/MONITORING AGENCY REPORT NUMBER NASA TM-108805	
11. SUPPLEMENTARY NOTES Point of Contact: Byun Chansup, Ames Research Center, MS 258-2, Moffett Field, CA 94035-1000; (415) 604-4526			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified — Unlimited Subject Category 02		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This paper presents a procedure for computing the aeroelasticity of wings on parallel multiple-instruction, multiple-data (MIMD) computers. In this procedure, fluids are modeled using Euler equations, and structures are modeled using modal or finite element equations. The procedure is designed in such a way that each discipline can be developed and maintained independently by using a domain decomposition approach. In the present parallel procedure, each computational domain is scalable. A parallel integration scheme is used to compute aeroelastic responses by solving fluid and structural equations concurrently. The computational efficiency issues of parallel integration of both fluid and structural equations are investigated in detail. This approach, which reduces the total computational time by a factor of almost 2, is demonstrated for a typical aeroelastic wing by using various numbers of processors on the Intel iPSC/860.			
14. SUBJECT TERMS Aeroelasticity, Parallel computation, Wings			15. NUMBER OF PAGES 24
			16. PRICE CODE A03
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT

