JOURNAL OF AIRCRAFT
Vol. 33, No. 2, March–April 1996

Wing–Body Aeroelasticity on Parallel Computers

Chansup Byun* and Guru P. Guruswamy†
NASA Ames Research Center, Moffett Field, California 94035-1000

This article presents a procedure for computing the aeroelasticity of wing–body configurations on multiple-instruction, multiple-data parallel computers. In this procedure, fluids are modeled using Euler equations discretized by a finite difference method, and structures are modeled using 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. A parallel integration scheme is used to compute aeroelastic responses by solving the coupled fluid and structural equations concurrently while keeping modularity of each discipline. The present procedure is validated by computing the aeroelastic response of a wing and comparing with experiment. Aeroelastic computations are illustrated for a high speed civil transport type wing–body configuration.

Introduction

THE analysis of aeroelasticity involves solving fluid and structural equations together. Both uncoupled and coupled methods can be used to solve problems in aeroelasticity associated with nonlinear systems.[1] Uncoupled methods are less expensive but are limited to very small perturbations with moderate nonlinearity. However, aeroelastic problems of aerospace vehicles are often dominated by large structural deformations and high flow nonlinearities. Fully coupled procedures are required to solve such aeroelasticity problems accurately.

Such coupling procedures result in an increased level of complication. Therefore, aeroelastic analysis has been mostly performed by coupling advanced computational fluid dynamics (CFD) methods with simple structural modal equations or advanced computational structural dynamics (CSD) methods with simple flow solutions. However, these approaches can be less accurate for the aeroelastic analysis of practical problems such as a full aircraft configuration in transonic regime. It is necessary to develop a fully coupled procedure utilizing advanced computational methods for both disciplines.

Recently, coupled fluid–structural interaction problems are being studied using finite difference Euler or Navier–Stokes flow equations and finite element structural equations of motion as demonstrated by an aeroelastic code ENSAERO.[2,3] However, applications are limited to simple structural models. For the complicated fluid and structural models, computations are performed in a step-by-step fashion.[4] The main reason is that the use of detailed models for both disciplines requires unprecedented computing speeds and amounts of memory. The emergence of a new generation of parallel computers can possibly alleviate the restriction on the computational power.

To solve the coupled fluid–structural equations some attempts have been made to solve both fluids and structures in a single computational domain.[5,6] The main defect of this approach is the ill-conditioned matrices associated with two physical domains with large variations in stiffness properties. So far, such attempts have been limited to simple two-dimensional problems.

To overcome the difficulties arising from a single domain approach, the domain decomposition approach reported in Ref. 1 has been incorporated in several advanced aeroelastic codes such as XTRAN3S,[7] ATRANS3S,[8] and CAP-TSD,[9] based on the transonic small perturbation theory. This domain decomposition approach models fluids and structures independently. The coupling of two disciplines is accomplished by exchanging data at interfaces between fluids and structures. This allows one to take full advantage of the numerical procedures for individual disciplines such as finite difference methods for fluids and finite element methods for structures. It was later demonstrated that the same technique can be used for modeling the fluids with Euler/Navier–Stokes equations on moving grids.[10,11] The accuracy of the coupling is maintained by matching the surface grid deformation with the structural displacements at the surface. This new development is incorporated in the computer code ENSAERO.[12] Similar work has also been reported recently in Ref. 13.

For the implementation of the ENSAERO code on parallel computers, two types of parallel computers were considered. These are the single-instruction, multiple-data (SIMD) and multiple-instruction, multiple-data (MIMD) type computers. However, MIMD type parallel computers are more suitable for computationally efficient implicit solvers and the domain decomposition approach used in the ENSAERO code. By decomposing the computational domain into a number of subdomains and solving an implicit problem on each subdomain, a MIMD computer can reduce the interprocessor communication required for the inversion of a large matrix resulting from an implicit method. Furthermore, a MIMD parallel computer can exploit the parallelism offered by the domain decomposition approach for the coupled fluid and structural disciplines; each computational domain can be treated concurrently. In addition, each fluid and structural algorithms can be designed in a modular fashion on MIMD parallel computers.

In this work, a procedure to compute aeroelasticity on MIMD parallel computers is described. The fluid and structural equations on separate computational domains are coupled by the exchange of interface data. The computational–efficiency issues of parallel integration of both fluid and structural equations are investigated using a parallel version of ENSAERO. The fluid and structural disciplines are modeled using finite difference (FD) and finite element (FE) approaches, respectively. The coupled equations are solved using a time integra-

Presented as Paper 94-1487 at the AIAA 35th Structures, Structural Dynamics, and Materials Conference, Hilton Head, SC, April 18–20, 1994; received May 15, 1994; revision received June 20, 1995; accepted for publication Sept. 4, 1995.

*Research Scientist, MCAT Institute. Member AIAA.
†Research Scientist, Computational Sciences Branch. Associate Fellow AIAA.

tion method with configuration–adaptive moving grids. The procedure is designed in a modular fashion so that each computational discipline can be developed independently and be modified easily. The aeroelastic computations are demonstrated for a high speed civil transport (HSCT) type wing–body configuration on the Intel iPSC/860 parallel computer.

Aeroelastic Computation

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

$$[M]\{\ddot{q}\} + [C]\{\dot{q}\} + [K]\{q\} = \{Z\} \quad (1)$$

where $[M]$, $[C]$, and $[K]$ are the global mass, damping, and stiffness matrices, respectively. $\{Z\}$ is the aerodynamic force vector corresponding to displacement vector $\{q\}$. One of the main efforts is computing the aerodynamic force vector $\{Z\}$, which is obtained by solving the fluid flow equations. After obtaining the aerodynamic force, aeroelastic responses can be obtained by solving Eq. (1). A numerical integration technique based on the constant–average–acceleration method[14] is used to integrate the aeroelastic equations. This is an unconditionally stable scheme.

A domain decomposition approach is selected to solve Eq. (1) in conjunction with the flow equations. Each of the fluid and structural equations is modeled in a separate computational domain. Coupling between the fluid and structural equations is accomplished by exchanging boundary interface data at the end of every time step when solving Eq. (1). The advantage of this approach is that one can select an efficient algorithm for the fluid domain regardless of the structural domain and vice versa. In this work, a finite difference method is selected for fluids and a finite element method for structures.

In the fluid domain the strong conservation law form of the Euler equations is used to model the flow. To solve the Euler equations, the central-difference scheme based on the implicit approximate factorization algorithm of Beam and Warming[15] with modifications by Pulliam and Chaussee[16] for diagonalization is used. The scheme is first-order accurate in time.

To exchange boundary interface data, it is necessary to represent the equivalent aerodynamic loads (i.e., normal stress) at the structural nodal points and to represent the deformed structural configurations at the aerodynamic grid points. Several numerical procedures have been developed to exchange the necessary information between the fluid and structural domains.[17–20]

A grid-to-element approach is used to define the location of the points of the fluid surface grid relative to finite elements at the surface of the structure for coupling purposes. In this approach, every grid point of the fluid that lies on the fluid–structural interface is identified with respect to a finite element as shown in Fig. 1. However, in general, it is not straightforward to determine the local coordinate information of each

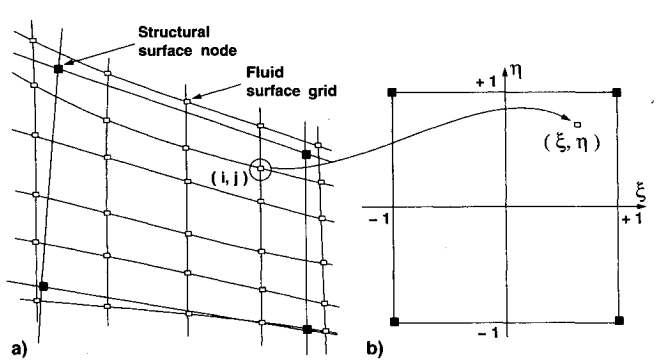

Fig. 1 Schematic diagram of the grid-to-element approach for the fluid–structural interface: a) physical and b) computational domains.

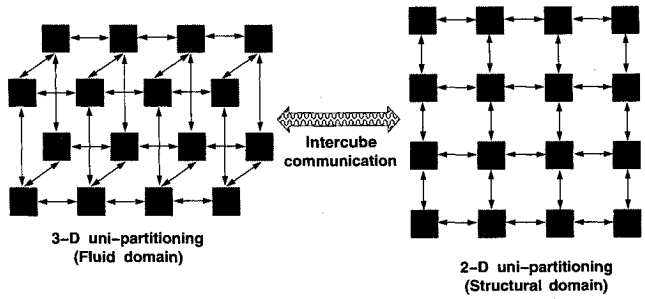

Fig. 2 Processor arrangements and message exchanges through interprocessor and intercube communications.

grid point within a finite element. A numerical inverse mapping technique developed by Murti and Valliappan[21] is used to obtain the local coordinate information of all the interface points of the fluid grid with respect to surface elements of the structure. Once the location of each fluid grid point is obtained, the nodal force vector can be easily obtained. Also, the deformation of the fluid surface grid is determined by using shape functions of the finite elements used to model the structure. In addition, singular planes associated with the fluid grid topology have to be moved according to the deformation of the structural surface since the fluid grid lines from a singular plane continue on to the structural surface smoothly. For this purpose, a linear extrapolation is used in this study. Starting from the deformed fluid surface grid, the field grid is generated as explained in a later section.

Parallelization of ENSAERO

The domain decomposition approach 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. Each domain is mapped individually onto a group of processors, referred to as a cube on the Intel iPSC/860, which is selected for this work. The Intel iPSC/860 is a distributed-memory, MIMD computer with 128 processors.

Because of coupling between the two disciplines, the interface boundary data such as surface pressures and structural displacements should be exchanged. This exchange between the fluid and structural domains is accomplished through an intercube communication mechanism.[22] This intercube communication facility enables different processors in each cube on the iPSC/860 to communicate directly.

The fluid flow algorithm solves the Euler equations using three-dimensional unipartitioning of the computational domain. The unipartitioning scheme assigns one subdomain grid to each of the processors. The mapping of subdomain grids to processors is described in Fig. 2. The arrows denote bidirectional data communication. There are a variety of concurrent algorithms available for solving the system of equations for fluids. More details about the implementation of the fluid flow algorithms can be found in Ref. 23.

For the structural domain, regular finite element meshes are used to model the wing and the body as plate and shell structures, respectively. This is a simple representation of the structure since the actual wing and body are built-up with sections, i.e., internal structural members supporting the external skin. However, for the domain decomposition approach used in this study, the fluid–structural coupling is achieved by exchanging the required information at the boundary interface, i.e., the surface in this case. Therefore, this model is considered to be adequate for the simulation of the fluid–structural coupling. It should be noted that the data structure for the finite element modeling developed in this study is general so that complex structures can be easily analyzed. This approach has been applied to a wing–box finite element model.[24] The domain decomposition is made by using two-dimensional unipartitioning

as shown in Fig. 2. This type of domain decomposition enables an efficient and simple message communication mechanism within the structural domain.

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 obtained by implementing the diagonal preconditioner to a parallel conjugate gradient algorithm proposed by Law.[25] In this method, the structural finite element model is divided into subdomains and only local matrices related to the subdomains are assembled. 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. Interprocessor communication is confined to the solution phase and the communication is only performed between processors that have common finite element nodes.

Aeroelastic Configuration Adaptive Grids

One of the major difficulties in solving the Euler equations for computational aerodynamics lies in the area of grid generation. For steady flows, advanced techniques such as blocked zonal grids[26] are currently being used. However, grid-generation techniques for aeroelastic calculations, which involve moving components, are still in the early stages of development. Guruswamy has developed analytical schemes for aeroelastic configuration-adaptive dynamic grids and demonstrated time-accurate aeroelastic responses of wing[10] and wing–body[2,3] configurations.

In this work, an H–O type grid topology is used (H in the streamwise and O in the spanwise directions) for wing–body configurations. This type of grid topology is more suitable for general wing–body configurations. It gives better surface grid resolution on the body when compared to the C–H grid topology used in Ref. 10. The grid is designed so that flow phenomena such as shock waves, vortices, etc., and their movement around the wing–body configurations are accurately simulated. The base surface grid is prepared using the S3D code.[27] From the surface grid, the field grid is generated using an analytical approach. In this approach, grid lines in the radial direction away from the surface are generated line-by-line in the planes normal to the longitudinal body axis. First, the radial lines are generated approximately normal to the surface. Then, the new grid lines in the azimuthal direction are generated in such a way that the spacing between lines are exponentially increased away from the surface.

This method can be used for generating the base field grid of the rigid configuration and the aeroelastically deformed field grid of the flexible configuration. For aeroelastic analysis, the displacements at the structural nodes are computed first using Eq. (1). Then the displacements are mapped onto the interface grid points by the grid-to-element approach mentioned earlier. Finally, the field grid is analytically regenerated starting from the deformed surface grid.

Parallelization of this approach is accomplished using the unipartitioning scheme in the fluid domain. The present approach for aeroelastic configuration-adaptive grids only requires the deformed surface grid and the coefficients used in the exponential function to define the grid spacing between lines away from the surface. The interprocessor communication needed to generate the deformed field grid within the fluid domain is minimal and takes place only between processors assigned along the surface-normal direction. Each of the processors can generate the assigned subdomain grid of the deformed field grid concurrently once information about the local surface grid has been broadcast.

The grid is generated at every time step based on the aeroelastically deformed position of the structure. First, the displacements at the points of the fluid surface grid on the structure are obtained on the processors assigned to the structural domain. This is done by using the local coordinate information and the finite element shape functions. The displacements at the points of the fluid surface grid are sent only to the appropriate processors on the fluid domain that contain the surface grid points. Then, a linear extrapolation is used to move the remaining points on singular planes of the fluid surface grid according to the structural deformation of the adjacent points on the structural surface. At this stage, the deformed surface grid is distributed only to processors of the fluid domain that contain the local surface grid points. It should be noted that the deformed surface grid residing on each processor of the fluid domain is only the part of the whole surface grid according to the grid partitioning. If two or more processors are assigned along the surface-normal direction, each of the deformed surface partitions is sent to processors that have the same partitioning indices of the surface grid. Finally, all processors of the fluid domain generate their subdomain of the deformed field grid concurrently.

Parallel Integration for Coupled Domains

In a serial computer, the integration of both fluid and structural equations is performed one after the other in a sequential nature. Figure 3a shows the sequential integration scheme implemented on MIMD parallel 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 sur-

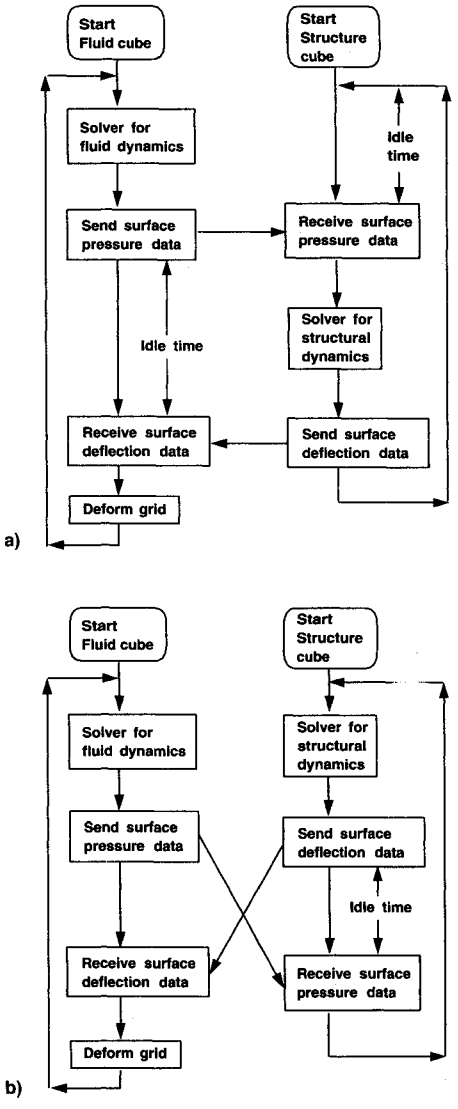

Fig. 3 Flow diagrams for a) sequential and b) parallel integration schemes.

face pressure data. Both cubes have their own idle times waiting for data communications. The computational time per integration step will be determined by times spent on both domains when a sequential integration scheme is used. To avoid the idle times between the fluid and structural computations all processors can be used to solve the fluid and structural equations sequentially as done in serial computations. But this approach requires more memory per processor and two disciplines have to be implemented in a single program. As a result, modularity of each algorithm for individual disciplines will have to be sacrificed to a significant degree. In addition, this approach will be less efficient as increasing the number of processors because the problem is not linearly scaled.

However, while keeping modularity of each discipline, computations can be done more efficiently on MIMD parallel computers by executing the integration of both fluid and structural equations concurrently, as shown in Fig. 3b. In the proposed parallel integration scheme, both domains 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 (the fastest) will have to wait. This integration scheme exploits the parallelism offered by the domain decomposition approach to solve the coupled fluid–structural interaction problems.

Results

Wing Aeroelasticity

To validate the present development, computations were done for a clipped delta wing configuration.[28] The transonic flutter characteristics of this wing are available from wind-tunnel tests for various flow parameters. For this computation, the flowfield is discretized using a C–H grid topology of size 151 × 30 × 25. The fluid 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 both the spanwise and surface-normal directions.

A 20 degrees-of-freedom (DOF) ANS4 shell[29] element was used for the finite element modeling of the structure. Since the wing model used in the experiment was built by using an aluminum–alloy flat plate insert covered with a light, flexible material to obtain the desired airfoil shape, in this computation, the wing is modeled as a plate. Considering the wing model 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 to better match the computed natural frequencies with those obtained from the experiment. This is based on the assumptions that the stiffness of the wing is dominated by an aluminum–alloy insert and that the mass distribution of the wing is significantly changed due to plastic foams covering the aluminum–alloy insert. For the structures part of the computation, processors were assigned as a two-dimensional mesh of two processors in the chordwise and spanwise directions, respectively, on the iPSC/860.

To compare sequential and parallel integration schemes the aeroelastic responses were obtained using both schemes on the iPSC/860. The results are presented in Fig. 4. The responses were obtained for 0-deg angle of attack (AOA) at $M_\infty = 0.854$ and a given dynamic pressure of 1.0 psi. The two results are in excellent agreement. When the domain decomposition approach is used to solve the coupled fluid–structural equations, the parallel integration scheme can reduce the computational time per integration step. For 256 finite elements (1360 DOF) with four processors on the structural domain, the computational times per integration step are 6.22 and 3.33 s by using sequential and parallel integration schemes, respectively. A speed-up factor of 1.87 is achieved by using the parallel integration scheme. The parallel integration scheme enables con-

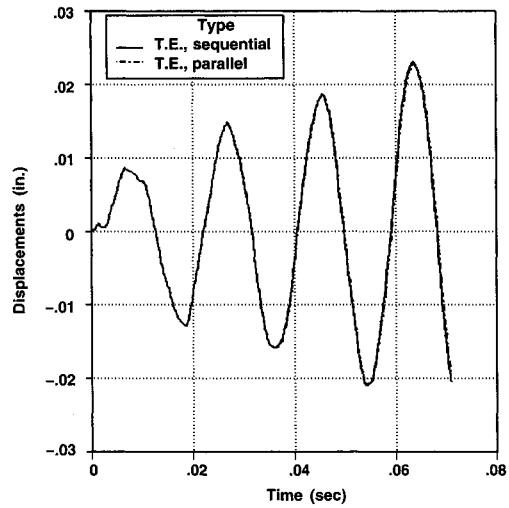

Fig. 4 Aeroelastic responses obtained by using sequential and parallel integration schemes ($M_\infty = 0.854$, $\alpha = 0$ deg, and $P = 1.0$ psi).

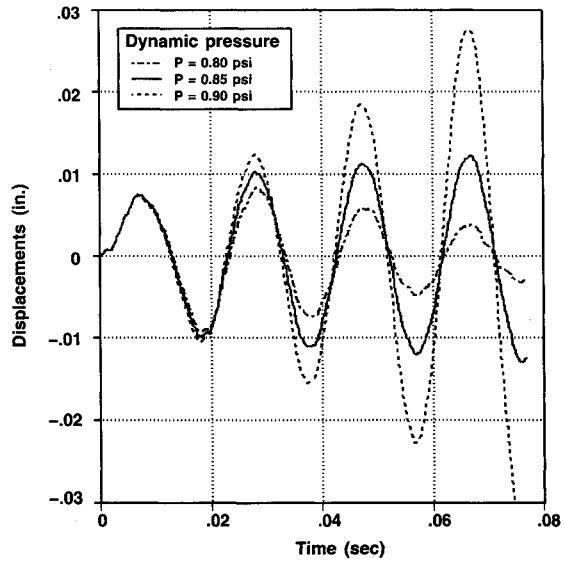

Fig. 5 Aeroelastic responses of a clipped delta wing obtained by solving finite difference Euler equations and finite element structural equations of motion ($M_\infty = 0.854$ and $\alpha = 0$ deg).

current solution of the coupled fluid and structural equations without causing any significant inaccuracy or instability problems. The computation time per integration step required is determined by the computational domain that requires most time per integration step. This parallel integration is one of the advantages of using MIMD computers for multidisciplinary analysis using the domain decomposition approach.

Aeroelastic responses were also computed for various other dynamic pressures to predict the flutter dynamic pressure and compare with the experiment. Figure 5 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 Fig. 5, 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.[28] Considering the lack of experimental pressure data on the wing and the error involved in modeling the wing since there was no data available for the material properties of the plastic foam covered the aluminum insert to form airfoil shape,

the computational result is deemed an acceptable prediction of the flutter dynamic pressure.

Wing–Body Aeroelasticity

The main purpose of this work is to compute the aeroelastic responses of fully flexible wing–body configurations on MIMD parallel computers. For this purpose, a general-purpose moving-grid capability is required. In the present work, an analytical scheme[3] that will generate a moving H–O grid is implemented on the iPSC/860. This scheme generates the field grid according to the surface grid deformation. For demonstration purposes, an HSCT type wing–body configuration (1807 model) is selected. Figure 6 shows the baseline grid. The size of the baseline grid is 95 × 89 × 30. However, it should be noted that the technology developed in this work for moving grid is independent of grid size. The grid generated by the code when the structure is deformed is shown in Fig. 7. Note that the singular planes upstream of the leading edge and downstream of the trailing edge are deformed according to the deformed shape of the configuration.

To verify the coupling of the surface movement with the grid movement, dynamic aeroelastic responses are obtained for the above wing–body configuration. Both the body and wing are allowed to be flexible. The wing–body configuration is modeled as a plate/shell structure using 308 elements. The

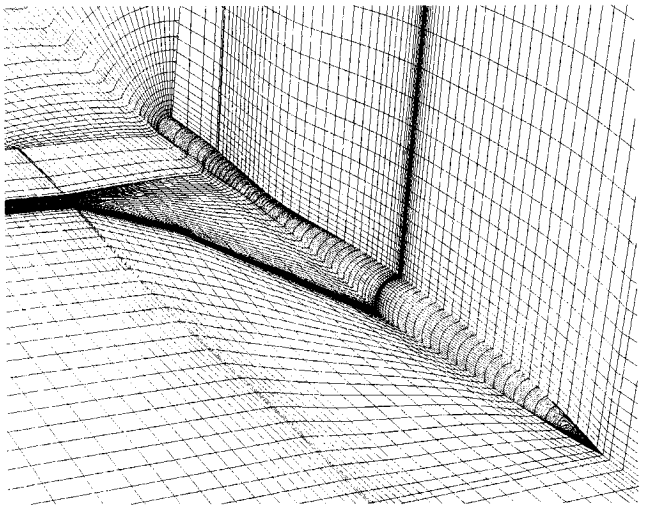

Fig. 6 HSCT type wing–body configuration with portions of surface and field physical grids.

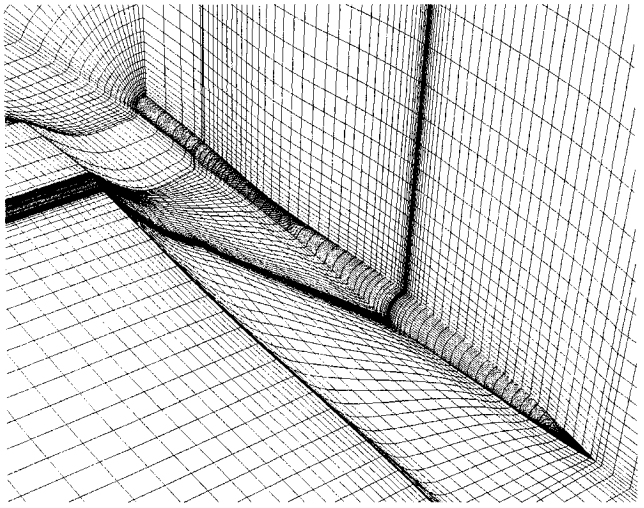

Fig. 7 Deformed surface and field grids.

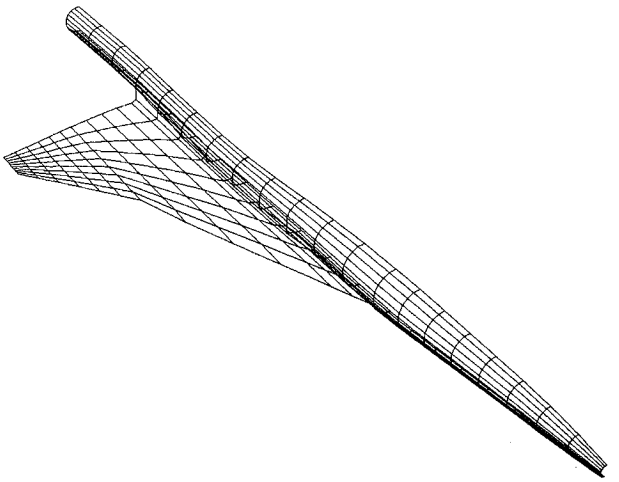

Fig. 8 Finite element modeling of the wing–body configuration. Total number of elements = 308; total number of nodes = 351; boundary conditions = symmetry conditions at top and bottom of the body and fix all DOFs along bottom of midbody; total number of equations = 1641.

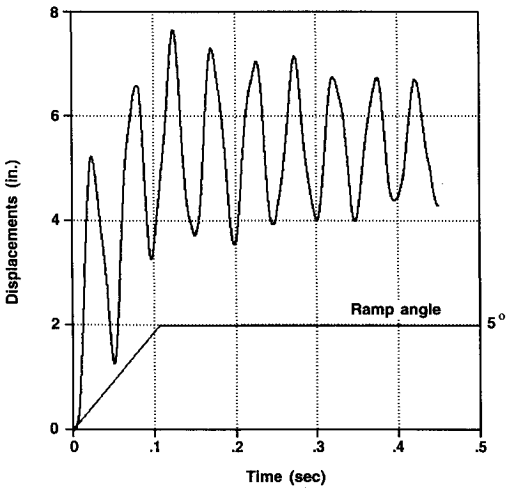

Fig. 9 Dynamic aeroelastic displacements at the wingtip leading edge of an HSCT type wing–body with ramp motion (M_∞ = 2.1, α = 4.75 deg, and P = 3.0 psi).

structural model does not have internal structures such as ribs or spars. However, this model is considered to be adequate for the purpose of this study. The finite element layout is shown in Fig. 8. The structural properties are chosen so that the structure can demonstrate aeroelastic responses for a given flow condition. Symmetric boundary conditions are applied at the top and bottom of the body symmetry lines. All DOF are fixed along the bottom symmetry line of the midbody. This results in a total of 1641 DOF for the structure.

Aeroelastic computations are done on the flexible wing–body configuration by directly coupling the pressures computed solving the Euler equations with the FE structural equations of motion. A demonstration calculation is done for a dynamic aeroelastic case when the configuration is ramping up from 0- to 5-deg AOA at M_∞ = 2.1 as shown in Fig. 9. This ramping motion is started from the steady state of 4.75-deg AOA and M_∞ = 2.1. It is assumed that the wing root is 300 in. long and aeroelastic computations are done at a dynamic pressure of 3.0 psi. The configuration is pitched up about the axis perpendicular to the symmetry plane and located at the leading edge of the wing root. Starting from the steady-state solution, the configuration is pitched up at a rate of 0.0015 deg per time step. At the end of each time step a new field

grid is generated that conforms to the deformed surface. Figure 9 shows the response of the leading edge of the tip section. It is noted that the wing continues to oscillate after the ramp motion has stopped. This is because the inertial force on the structure is still dominating the aeroelastic motion.

For the purpose of flutter speed prediction, the aeroelastic configuration adaptive grid should be able to handle large structural deformations that cause severe distortions of the original fluid grid. The capability of the current grid deforming scheme to handle a large deflection is demonstrated in Fig. 10. The computations are started from the steady-state solution at 4.75-deg AOA and $M_\infty = 2.1$ with an initial motion of the structure due to uniform accelerations to simulate gust loads. To produce a large deflection a dynamic pressure of 15.0 psi is applied. The wingtip deflection increased up to about 13% of the root chord length in the beginning and then decreased gradually due to the aerodynamic damping effect. It shows that the response approaches to a steady state that is due to the aerodynamic force at the flight configuration. The current aeroelastically deforming grid scheme could generate deformed fluid grids following structural deformations without failure. Including the data exchange between fluid and structural domains, the current scheme requires about 12% of the computational time per each integration step.

Performance

To measure the performance of the structural domain on the Intel iPSC/860, the floating-point operations (FLOP) rate on the iPSC/860 is calculated by comparing time per integration step on the iPSC/860 to the time 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 achieve the Y-MP equivalent of 4.2 Mflops, while the corresponding rate is about 77 Mflops on a single Y-MP processor. The Intel rate is about 7% of the peak performance of a single processor on the iPSC/860. Similar performance was reported by Ryan and Weeratunga[23] for the fluid domain. All performance data reported are for 64-bit arithmetic.

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 Fig. 11. The speed up relative to the Y-MP is defined as

$$\text{speed up} = t_{\text{Cray}}/t_{\text{Intel}}$$

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 approach that results in the minimum and maximum bandwidths of the stiffness matrix of each subdomain for a given number of processors, respectively.

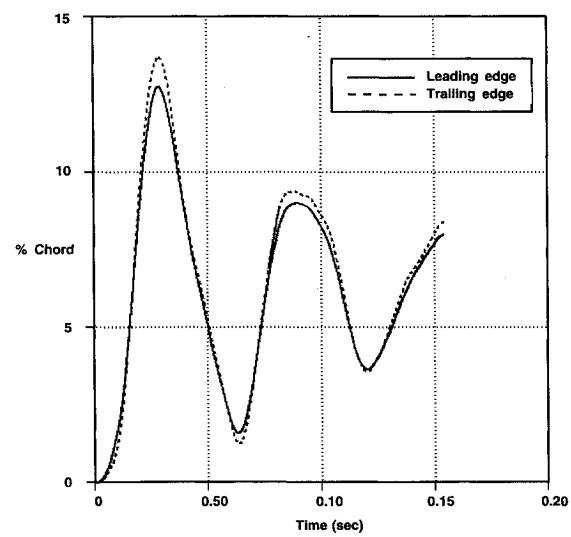

Fig. 10 Demonstration of aeroelastic configuration adaptive grid scheme for a large deformation of an HSCT type wing–body ($M_\infty = 2.1$, $\alpha = 4.75$ deg, and $P = 15.0$ psi).

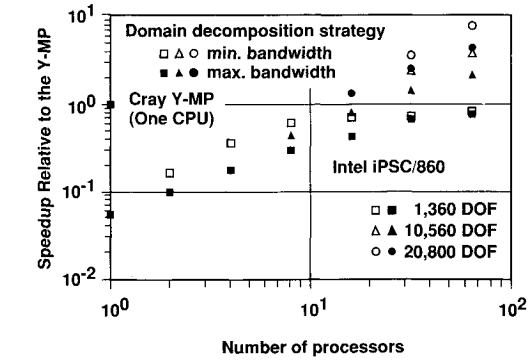

Fig. 11 Computational performance of the structural domain in ENSAERO with various problem sizes and domain decompositions on the Intel iPSC/860.

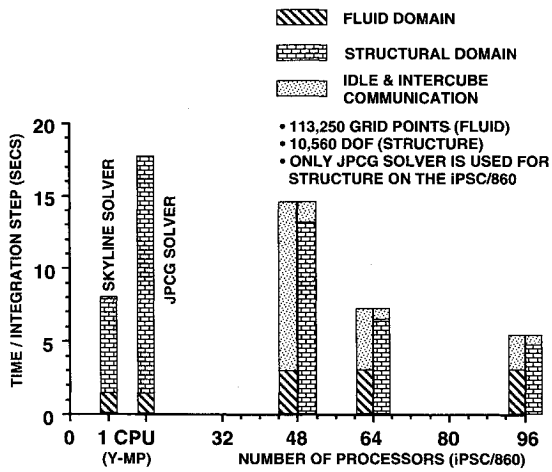

Fig. 12 Overall computational performance of ENSAERO on the Cray Y-MP and the Intel iPSC/860.

For the case of 1360 DOF, the computational time per integration step for 64 processors on the iPSC/860 is close to that on the Y-MP. However, by increasing the size of the problem (10,560 and 20,800 DOF), 16 processors of the iPSC/860 achieve about the same speed as a single Y-MP processor. It is evident that the JPCG solver on the iPSC/860 performs better as the size of the problem increases. For the case of 20,800 DOF, the relative speed up achieved is about 8 when 64 processors are in use on the iPSC/860.

The overall performance of ENSAERO on both the Y-MP and the iPSC/860 is shown in Fig. 12 for the case of 113,250 grid points for the fluid domain and 10,560 DOF for the structural domain. In this computation, 32 processors are assigned to the fluid domain and 16–64 processors to the structural domain. Both the skyline reduction and JPCG solvers are compared on the Y-MP, whereas only the JPCG solver is used for the structural domain. The height of each column stands for the time per integration step. Each column is divided into times spent for the fluid domain, the structural domain, and for idle/intercube communication.

For the structural domain, 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. Direct solvers are still under development on parallel computers. The main purpose of this work is to compute aero-

elastic responses of aerospace vehicles on MIMD parallel computers. Therefore, the well-developed JPCG solver is selected. Because of the domain decomposition approach used in this work, the JPCG solver can be easily replaced with more efficient solvers when they become available.

When using 32 processors for the structural domain, the JPCG solver on the iPSC/860 achieves the performance of the skyline reduction solver on the Y-MP. The time per integration step of ENSAERO using 96 processors in total on the iPSC/860 is about 60% of that obtained using the skyline reduction solver with a single processor on the Y-MP. This result is based on the computation time for the case of 113,250 fluid grid points and 10,560 structural equations. 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. However, due to the domain decomposition approach, it is possible to balance the computational time between the two domains by assigning more processors to the structural domain.

Conclusions

A parallel wing–body version of a multidisciplinary code, ENSAERO, has been developed on the Intel iPSC/860. A domain decomposition approach was used to enable algorithms for the fluid and structural disciplines to be developed and maintained independently. This approach provides an efficient and effective environment to researchers. A researcher working in the fluid or the structural discipline can develop his own algorithms independent of the others. The only thing to be done together is coupling of the disciplines. Since coupling of the disciplines is achieved by exchanging boundary data through an intercube communication mechanism that does not interfere with interprocessor communication within a cube, coupling should not cause any problem. This makes it easy for each discipline to incorporate and develop new algorithms or data structures without interferences. For example, the simple structural model has been replaced with a more realistic model, such as a wing–box structure for the wing configurations.[24]

The performance of the structural domain is far behind that of the fluid domain. This is due to the less desirable performance of the JPCG algorithm. It is noted that direct solvers are still in the early stages of development. However, since the procedure developed here allows for one domain to select algorithms independent of others, the JPCG algorithm can be easily replaced with more efficient algorithms when available. Although the solver for the structure is not efficient on a serial computer, reasonable computational speed and a good load balance can be achieved by assigning more processors to the structural domain. The overall time per integration step of parallel ENSAERO using 96 processors on the iPSC/860 is reduced to about 60% of the best time obtained on a single Y-MP processor for the particular problem considered. This shows the advantage of using the domain decomposition approach for the multidisciplinary analysis on MIMD parallel computers.

The parallel integration scheme enables the combination of advanced CFD and CSD technologies with minimal increase in computational time per integration step while keeping modularity of each discipline. The time per integration step is solely determined by the domain that requires most computational time on the iPSC/860. 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.

Acknowledgments

This work was completed using the resources of the Numerical Aerodynamic Simulation (NAS) Program at NASA Ames Research Center. The work done by the first author was funded through NASA Ames Research Center Cooperative Agreement NCC2-740 under the HPCC program. The authors would like to thank S. Weeratunga for the parallel version of ENSAERO for the fluid domain.

References

[1]Guruswamy, P., and Yang, T. Y., "Aeroelastic Time-Response Analysis of Thin Airfoils by Transonic Code LTRAN2," *Computers and Fluids*, Vol. 9, No. 4, 1980, pp. 409–425.

[2]Guruswamy, G. P., "Coupled Finite-Difference/Finite-Element Approach for Wing-Body Aeroelasticity," AIAA Paper 92-4680, Sept. 1992.

[3]Guruswamy, G. P., and Byun, C., "Fluid-Structural Interactions Using Navier-Stokes Flow Equations Coupled with Shell Finite Element Structures," AIAA Paper 93-3087, July 1993.

[4]Weaver, M. A., Gramoll, K. C., and Roach, R. L., "Structural Analysis of a Flexible Structural Member Protruding into an Interior Flow Field," AIAA Paper 93-1446, April 1993.

[5]Bendiksen, O. O., "A New Approach to Computational Aeroelasticity," *AIAA 32nd Structures, Structural Dynamics, and Materials Conference*, AIAA, Washington, DC, 1991, pp. 1712–1727 (AIAA Paper 91-0930).

[6]Felker, F. F., "A New Method for Transonic Static Aeroelastic Problems," *AIAA 32nd Structures, Structural Dynamics, and Materials Conference*, AIAA, Washington, DC, 1992, pp. 415–425 (AIAA Paper 92-2123).

[7]Borland, C. J., and Rizzetta, D., "XTRAN3S—Transonic Steady and Unsteady Aerodynamics for Aeroelastic Applications, Volume I—Theoretical Manual," Air Force Wright Aeronautical Labs., AFWAL-TR-80-3107, Dec. 1985.

[8]Guruswamy, G. P., Goorjian, P. M., and Merritt, F. J., "ATRAN2S—An Unsteady Transonic Code for Clean Wings," NASA TM 86783, Dec. 1985.

[9]Batina, J. T., Bennett, R. M., Seidal, D. A., Cunningham, S. R., and Bland, S. R., "Recent Advances in Transonic Computational Aeroelasticity," NASA TM 100663, Sept. 1988.

[10]Guruswamy, P., "Unsteady Aerodynamics and Aeroelastic Calculations of Wings Using Euler Equations," *AIAA Journal*, Vol. 28, No. 3, 1990, pp. 461–469.

[11]Guruswamy, G. P., "Vortical Flow Computations on a Flexible Blended Wing-Body Configuration," *AIAA Journal*, Vol. 30, No. 10, 1992, pp. 2497–2503.

[12]Guruswamy, G. P., "ENSAERO—A Multidisciplinary Program for Fluid/Structural Interaction Studies of Aerospace Vehicles," *Computing System Engineering*, Vol. 1, Nos. 2–4, 1990, pp. 237–256.

[13]Lee-Rausch, E. M., and Batina, J. T., "Calculation of AGARD Wing 445.6 Flutter Using Navier-Stokes Aerodynamics," AIAA Paper 93-3476, Aug. 1993.

[14]Newmark, N. M., "A Method of Computation for Structural Dynamics," *Journal of Engineering Mechanics Division*, Vol. 85, 1959, pp. 67–94.

[15]Beam, R., and Warming, R. F., "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form," *Journal of Computational Physics*, Vol. 22, No. 9, 1976, pp. 87–110.

[16]Pulliam, T. H., and Chaussee, D. S., "A Diagonal Form of an Implicit Approximate Factorization Algorithm," *Journal of Computational Physics*, Vol. 39, No. 2, 1981, pp. 347–363.

[17]Harder, R. L., and Desmarais, R. N., "Interpolation Using Surface Spline," *Journal of Aircraft*, Vol. 9, No. 2, 1972, pp. 189–191.

[18]Appa, K., Yankulich, M., and Cowan, D. L., "The Determination of Load and Slope Transformation Matrices for Aeroelastic Analysis," *Journal of Aircraft*, Vol. 22, No. 8, 1985, pp. 734–736.

[19]Appa, K., "Finite-Surface Spline," *Journal of Aircraft*, Vol. 26, No. 5, 1989, pp. 495, 496.

[20]Pidaparti, R. M. V., "Structural and Aerodynamic Data Transformation Using Inverse Isoparametric Mapping," *Journal of Aircraft*, Vol. 29, No. 3, 1992, pp. 507–509.

[21]Murti, V., and Valliappan, S., "Numerical Inverse Isoparametric Mapping in Remeshing and Nodal Quantity Contouring," *Computers and Structures*, Vol. 22, No. 6, 1986, pp. 1011–1021.

[22]Barszcz, E., "Intercube Communication on the iPSC/860," *Proceedings of the Scalable High Performance Computing Conference* (Williamsburg, VA), IEEE Computer Society Press, Los Alamitos,

CA, 1992, pp. 307–313.

[23]Ryan, J. S., and Weeratunga, S. K., "Parallel Computation of 3-D Navier-Stokes Flowfields for Supersonic Vehicles," AIAA Paper 93-0064, Jan. 1993.

[24]Bhardwaj, M., Kapania, R. K., Byun, C., and Guruswamy, G. P., "Parallel Aeroelastic Computations by Using Coupled Euler Flow and Wing-Box Structural Models," AIAA Paper 95-1291, April 1995.

[25]Law, K. H., "A Parallel Finite Element Solution Method," *Computers and Structures*, Vol. 23, No. 6, 1986, pp. 845–858.

[26]Holst, T. L., Flores, J., Kaynak, U., and Chaderjian, N., "Navier-Stokes Computations, Including a Complete F-16 Aircraft," *Applied Computational Aerodynamics*, edited by P. A. Henne, Vol. 125, Progress in Astronautics and Aeronautics, AIAA, Washington, DC, 1990, Chap. 21.

[27]Luh, R. C., Pierce, L., and Yip, D., "Interactive Surface Grid Generation," AIAA Paper 91-0796, Jan. 1991.

[28]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, Nov. 1959.

[29]Park, K. C., Pramono, E., Stanley, G. M., and Cabiness, H. A., "The ANS Shell Elements: Earlier Developments and Recent Improvements," *Analytical and Computational Models of Shells*, edited by A. K. Noor, CED-Vol. 3, American Society of Mechanical Engineers, New York, 1989, pp. 217–240.