A98-31598

ICAS-98-4,6,3

COMPUTATIONAL UNSTEADY AERODYNAMICS IN AEROELASTIC SIMULATION

B.B. Prananta *
Delft University of Technology
Delft, The Netherlands

M.H.L. Hounjet †
National Aerospace Laboratory
Amsterdam, The Netherlands

H.W.M. Hoeijmakers ‡
University of Twente
Enschede, The Netherlands

Abstract: An affordable unsteady aerodynamics method suitable for three-dimensional aeroelastic simulations is presented. The method solves the unsteady Euler/Thin-Layer Navier-Stokes equations on a deforming mesh. The spatial discretization is based on a cell-centered scheme and employs an upwind method using Roe's flux difference splitting (FDS). The temporal integration is carried out efficiently in an implicit manner in which a set of nonlinear equations is solved at each time step using a relaxation method. Further, the method is parallelized using a domain decomposition technique. Discussions concerning the requirements for aeroelastic simulation, the solution method and some applications are presented. The quality of the numerical results and the turn-around time of the method demonstrate the applicability of the current method for routine aeroelastic simulation purposes.

Introduction

In a cooperation between Delft University of Technology (TU Delft) and the National Aerospace Laboratory (NLR) a Computational Aeroelastic Simulation (CAS) method has been developed. The method employs the Euler/Navier-Stokes equations for the unsteady aerodynamic part and a linear modal decomposition method for the structural part. The primary goal of this development is to obtain a CAS method with the potential for future industrial applications.

During the first stage of the research a two-dimensional method for solving the unsteady Euler/Navier-Stokes equations on a arbitrary moving mesh was developed.[21] The method serves as a test-bed for investigating the possible algorithms and modelings. From the many methods which were studied it turned out that an adequate method which allows large time steps to be taken, thereby minimizing the overall turn-around time of the method, should be based on:

- Roe's FDS for the inviscid flux.
- An implicit method employing line-relaxation.
- Thin-Layer assumption for the viscous terms.

*Graduate student/research assistant
†Senior research scientist
‡Professor of Engineering Fluid Dynamics

The extension to three-dimensional configurations was straightforward and the advantages of the two-dimensional method were retained.[19] Recently the development has focussed on improving the coupling procedure between the aerodynamic and the structural part in order to take full advantage of the large time step capability of the aerodynamic part.[20] The three-dimensional method has also been adapted for running concurrently in a workstation cluster as well as on multi processor (SMP or MPP) machines. The speed-up due to parallelization is very encouraging, adding to the possibility of application on a routine basis.

The present paper presents the computational methods employed in CAS (aerodynamic as well as structural) and highlights lessons learned from the parallelization and its validation. Finally the results obtained in applications for isolated wings and for more complex configurations are presented and compared with experimental results and results obtained with other methods.

Governing Equations

The set of governing equations consist of the unsteady aerodynamics part which employs the Navier-Stokes equations and the structural part which utilizes a modal decomposition.

Aerodynamics

For flows at high Reynolds number where the convection phenomena are dominant, it is appropriate to apply the thin-layer assumption. The governing equations are transformed from the Cartesian physical domain to a computational domain by $\xi = \xi(x,y,z,t), \eta = \eta(x,y,z,t), \zeta = \zeta(x,y,z,t)$ and $\tau = t$ to facilitate the discretization in a uniform grid. The conservative form of the Thin-Layer Navier-Stokes equations in a curvilinear coordinate system, with ζ represents the coordinate within the shear layer, reads:

$$\frac{\partial \widehat{Q}}{\partial \tau} + \frac{\partial \widehat{E}}{\partial \xi} + \frac{\partial \widehat{F}}{\partial \eta} + \frac{\partial \widehat{G}}{\partial \zeta} = \frac{\partial \widehat{G}_v}{\partial \zeta}, \qquad (1)$$

where the conservative variable is $\widehat{Q} = hQ$, $Q = [\rho, \rho u, \rho v, \rho w, \rho E]^T$, the inviscid flux vector in ξ direction

is $\widehat{E} = [\rho\widehat{U}, \rho u\widehat{U} + \widehat{\xi}_x p, \rho v\widehat{U} + \widehat{\xi}_y p, \rho w\widehat{U} + \widehat{\xi}_z p, \rho E\widehat{U} + \widehat{U'}p]^T$, and the contravariant velocity is $U = \xi_t + \xi_x u + \xi_y v + \xi_z w = \xi_t + U'$. The other inviscid fluxes, $\widehat{F}$ and $\widehat{G}$ are obtained by replacing (U, ξ) by (V, η) and (W, ζ), respectively. The metric coefficients, $\xi_x, \xi_y, \xi_z, \xi_t, \eta_x$,etc., are obtained from the transformation relation given e.g. in.[18] Quantities with a hat $(\widehat{\bullet})$ denote unscaled quantities with respect to $h = J^{-1}$, the inverse Jacobian of the transformation.

In the Thin-Layer approximation the flow variables can be fully factorized from the metrics and the viscous term $\widehat{G}_v$ can be written in the practical form:

$$\widehat{G}_v = \frac{\mu}{Re}\begin{bmatrix} 0 \\ m_{11}u_\zeta + m_{12}v_\zeta + m_{13}w_\zeta \\ m_{12}u_\zeta + m_{22}v_\zeta + m_{23}w_\zeta \\ m_{13}u_\zeta + m_{23}v_\zeta + m_{33}w_\zeta \\ \widehat{G}_v^4 \end{bmatrix}, \quad (2)$$

where:

$$\widehat{G}_v^4 = \frac{m_{11}}{2}(u^2)_\zeta + \frac{m_{22}}{2}(v^2)_\zeta + \frac{m_{33}}{2}(w^2)_\zeta + m_{12}(uv)_\zeta + m_{23}(vw)_\zeta + m_{13}(wu)_\zeta + m_4\frac{\gamma}{Pr}e_\zeta.$$

In Eq.(2) the Stokes hypothesis for bulk viscosity is used, by which $\lambda + 2\mu/3 = 0$ and only derivatives in ζ direction are retained. The metric functions m are presented in the appendix. The density has been nondimensionalized by ρ_∞, velocity components by a_∞, pressure by $\rho_\infty a_\infty^2$, energy by a_∞^2, the coefficient of viscosity by μ_∞ and the temperature by T_∞. The molecular viscosity is obtained from the Sutherland law. The eddy viscosity is obtained from the Spalart-Allmaras[25] one-equation turbulence model. The Baldwin-Lomax model was used in the previous studies but turned out to be not suited for parallel computing. The laminar and turbulent Prandtl numbers are taken to be constant: $Pr_L = 0.72$ and $Pr_T = 0.90$.

Boundary conditions. On a solid surface the normal velocity vanishes. Assuming that the ζ direction leaves the solid surface this condition is:

$$\widehat{W} = \widehat{\zeta}_x(u - \dot{x}) + \widehat{\zeta}_y(v - \dot{y}) + \widehat{\zeta}_z(w - \dot{z}) = 0. \quad (3)$$

Further for viscous flows the tangential velocity components vanish as well, which can be satisfied simply by setting:

$$u = \dot{x} \quad v = \dot{y} \quad w = \dot{z}. \quad (4)$$

Following the idea of Rizzi,[22] the pressure is extrapolated from the computational domain to the solid surface by using the normal momentum equation. For a moving boundary it reads:

$$(\nabla\widehat{\zeta} \bullet \nabla\widehat{\xi})\frac{\partial p}{\partial \xi} + (\nabla\widehat{\zeta} \bullet \nabla\widehat{\eta})\frac{\partial p}{\partial \eta} + (\nabla\widehat{\zeta} \bullet \nabla\widehat{\zeta})\frac{\partial p}{\partial \zeta} = \rho(h\frac{\partial\widehat{\zeta}_t}{\partial\tau} + u\frac{\partial\widehat{\zeta}_x}{\partial\tau} + v\frac{\partial\widehat{\zeta}_y}{\partial\tau} + w\frac{\partial\widehat{\zeta}_z}{\partial\tau}) + \rho\widehat{U}(\frac{\partial\widehat{\zeta}_t}{\partial\xi} + u\frac{\partial\widehat{\zeta}_x}{\partial\xi} + v\frac{\partial\widehat{\zeta}_y}{\partial\xi} + w\frac{\partial\widehat{\zeta}_z}{\partial\xi}) + \rho\widehat{V}(\frac{\partial\widehat{\zeta}_t}{\partial\eta} + u\frac{\partial\widehat{\zeta}_x}{\partial\eta} + v\frac{\partial\widehat{\zeta}_y}{\partial\eta} + w\frac{\partial\widehat{\zeta}_z}{\partial\eta}). \quad (5)$$

Eq.(5) is valid for inviscid flow and has been obtained after imposing the condition for the normal velocity, i.e. Eq.(3). For viscous flow the condition of zero velocity on the surface and neglecting the normal viscous stresses compared to normal pressure reduces the right-hand side (RHS) of Eq.(5) to simply:

$$\rho h\partial\widehat{\zeta}_t/\partial\tau.$$

The density is calculated using the adiabatic wall condition, $\partial T/\partial \boldsymbol{n} = 0$ or for viscous flow using the stagnation temperature condition on the solid surface. At the outer boundary, far from the airfoil, the flow is practically inviscid so that the boundary conditions for the Euler equations can be applied.

Structural Dynamics

In general the equation of motion of the aeroelastic system can be represented by:

$$M\ddot{\boldsymbol{x}} + C\dot{\boldsymbol{x}} + K\boldsymbol{x} = q_\infty S C_A(M_\infty, Re_\infty, \dot{\boldsymbol{x}}, \boldsymbol{x}), \quad (6)$$

where M, C and K are the mass, structural damping and stiffness matrices, respectively. $\boldsymbol{x}$ is the vector of physical displacement. q_∞ is the freestream dynamic pressure, S is the reference area and C_A is the aerodynamic force coefficient. In aeroelasticity usually the high frequency modes are hardly important and are cut-out from the analysis. To take advantage of this, the motion of the structure is represented by a finite number of modes obtained from solving:

$$M\ddot{\boldsymbol{x}} + K\boldsymbol{x} = 0. \quad (7)$$

By assuming the solution to be $\boldsymbol{x}(t) = \boldsymbol{\phi}e^{i\omega t}$, a standard eigenvalue problem is obtained which can be solved for the mode shapes $\boldsymbol{\phi}_i$ and their eigenfrequencies ω_i. The resulting mode shapes are normalized with respect to the mass matrix:

$$\boldsymbol{\phi}^T M\boldsymbol{\phi} = I, \quad \boldsymbol{\phi}^T K\boldsymbol{\phi} = \omega_i. \quad (8)$$

Substituting $\boldsymbol{x} = \boldsymbol{\phi}\boldsymbol{q}$ into Eq.(6), using Eq.(8) and assuming a diagonal damping (e.g. proportional damping), results in an independent set of equations:

$$\ddot{q}_i + 2\zeta\omega\dot{q}_i + \omega_i^2 q_i = q_\infty S Q_i, \; i = 1..N, \quad (9)$$

where Q_i is the nondimensional generalized aerodynamic force (GAF) defined as:

$$Q_i = \frac{1}{S}\int_S p\phi n dS. \qquad (10)$$

N is the number of modes which are employed in the computation. Most of the time N is much smaller than the number of degree of freedom employed in solving Eq.(7).

Aeroelastic Requirements

The interaction between the fluid and the structure occurs at the solid surface. The structural parameters at the solid surface (displacement, velocity and acceleration) determine the boundary conditions for the fluid flow, while the stresses of the fluid on the solid surface give the loading to the structure. In practice the working parameters for the coupling may differ according to the solution method. For example the loading on the structure can be expressed directly in the form of stresses[5] or in the form of forces.[10] In the present approach, where modal decomposition of the structural part is applied, the working parameter for the loading is the GAF, defined by Eq.(10). GAF represents the work of the fluid performed on the structure. To obtain accurate values of the generalized forces, an accurate prediction of the pressure is essential at the locations with large displacements normal to the surface. This implies also that the computational mesh should be generated according to this criterion. It should be noted that the accuracy requirement for the GAF should also take into account the physical uncertainties involved in structural modeling* and numerical errors due to inadequate resolution.

The requirement for the temporal integration method is dictated by the time scales involved in the simulation. It is assumed, without losing generality, that the number of time steps per cycle is fixed and the time scales can then be expressed in frequency scales. The frequency parameters and the nondimensional time in an aeroelastic simulation are given in the following table:

	aerodynamic	structure
f^*	$k_r = \omega_r c/U_\infty$	ω_r
τ^*	c/a_∞	$1/\omega_r$

where c is a reference length, U_∞ is freestream velocity and ω_r is the highest structural frequency of importance, which is chosen on the basis of experience. From a physical point of view the temporal integration method has to be able to capture the flow behavior only up to the allowed highest frequency of interest, defined as the reference frequency k_r. From the numerical point of view the lowest frequency, defined as k_{stab}, will be of interest. This frequency represents the numerical stability limit of a temporal integration method. If $k_{\text{stab}} > k_r$ then the method is said to be limited by numerical stability. Since large time steps may lead to a better efficiency it is desired that the choice of the time step should only be dictated by the adequate sampling of the physics. Therefore the quantity of interest to determine the requirement to the temporal integration method is the minimum acceptable value of k_r.

When the flow behaves dynamically linear (e.g. small amplitude of oscillation or relatively high frequency) the frequency scale of the flow is k_r. However, in transonic cases involving large amplitude motions where the shock waves travel over a significant traject and/or when flow separation occurs during at least a part of the oscillation and/or due to pilot induced effects, nonlinearities will generate higher harmonics in the response. This suggests the need of a higher frequency scale to prevent aliasing errors. Considering the entire aeroelastic simulation the higher frequency components of the aerodynamic forces would be important only if they coincide with structural frequencies associated with lightly or mildly damped vibration modes and have sufficient support in space and time, otherwise these components will decay anyway. Since this situation hardly appears in practice it may thus be concluded that the most stringent requirement for the time integration method of the aerodynamic part is k_r.

Experience shows that an accurate integration of the structural equations needs time steps of $\mathcal{O}(10)$ per cycle.[12] If a time step of this order is applied to a typical transonic aeroelastic case characterized by relatively low frequencies, medium sized mesh and a high subsonic Mach number, the CFL number in the unsteady aerodynamic calculation can easily exceed $\mathcal{O}(10^4)$. A method which is not sensitive to a CFL restriction is applied.[19]

Solution Method

Following the method of lines the spatial discretization and temporal integration are decoupled and treated separately.

Upwind Cell-Centered Finite-Volume Method

Eq.(1) is spatially discretized using a cell-centered finite-volume method applied to the transformed equations. At time level $(n+1)$ the semi-discretized equations in a uniform computational mesh read ($\Delta\xi = \Delta\eta = \Delta\zeta = 1$):

$$\frac{\partial Q}{\partial \tau} h^{n+1} + \frac{\partial h}{\partial \tau} Q^n + \widehat{E}^{n+1}_{i+\frac{1}{2},j,k} - \widehat{E}^{n+1}_{i-\frac{1}{2},j,k} + \widehat{F}^{n+1}_{i,j+\frac{1}{2},k} - \widehat{F}^{n+1}_{i,j-\frac{1}{2},k} + \bar{\widehat{G}}^{n+1}_{i,j,k+\frac{1}{2}} - \bar{\widehat{G}}^{n+1}_{i,j,k-\frac{1}{2}} = 0, \qquad (11)$$

where $\bar{\widehat{G}} = \widehat{G} - \widehat{G}_v$ and Q represents a cell-averaged value of the conservative variable.

The viscous flux is discretized as usual using central differences which adequately captures its elliptic nature. The inviscid flux is discretized using Flux Difference Splitting (FDS) employing Roe's approximate Riemann solver. From many possibilities this method was chosen because it:

*e.g. **M** might be estimated within 5% accuracy, **C** is hardly to estimate, **K** might be estimated within 10% accuracy and ϕ as a consequence, has an accuracy of about 20%.

- correctly capture shear layers: appropriate for Navier-Stokes equations, less sensitive to mesh stretching in the normal shear layer direction.
- easy applies to moving grid problems: the grid speed affects only the eigenvalues.
- is robust: applicable for implicit method, easy to construct an approximate Jacobian which ensures positivity.

The usually mentioned drawback of Roe's FDS is its expensiveness compared to Flux Vector Splitting (FVS) (e.g. van Leer FVS or AUSM). However, this advantage of FVS methods diminishes when the Jacobian of the flux has to be calculated (to facilitate implicit methods). The inviscid flux calculated using the Roe's FDS is, e.g. in a cell face $(i \pm \frac{1}{2}, j, k)$ in ξ direction:

$$\widehat{E} = \frac{1}{2}[\widehat{E}(Q^+) + \widehat{E}(Q^-)] - \frac{1}{2}|\widehat{A}(\bar{Q})|(Q^+ - Q^-), \quad (12)$$

where the fluxes are evaluated using the metrics data at the cell face, Q^+ and Q^- are the states on the two sides of the cell face, and $\bar{Q} = \bar{Q}(Q^+, Q^-)$ is the state at the cell face calculated using Roe averaging of Q^+ and Q^-, see.[23] $\widehat{A} = \partial\widehat{E}/\partial Q$ is the Jacobian of the flux with respect to the conservative variable. The states on the two sides of the face are obtained using the MUSCL variable extrapolation method due to van Leer (the limited κ-scheme):

$$Q^{\pm}_{i\mp\frac{1}{2}} = Q_i \mp \frac{s}{4}[(1 - \kappa s)dQ^{\pm}_i + (1 + \kappa s)dQ^{\mp}_i], \quad (13)$$

with $dQ^+_i = Q_{i+1} - Q_i$, $dQ^-_i = Q_i - Q_{i-1}$ and $\kappa = -1$ for second-order fully upwind and $\kappa = \frac{1}{3}$ for third-order (in one direction) upwind bias. s is the limiter function needed to keep the scheme monotonic. $|\widehat{A}|$ is calculated as $|\widehat{A}| = \tilde{R}|\widehat{\Lambda}|\tilde{L}$, where $\tilde{R}$ and $\tilde{L}$ are the matrix of right and left eigenvectors of the flux Jacobian $\tilde{A}$, respectively. $\widehat{\Lambda}$ is a diagonal matrix with the eigenvalues $\widehat{\lambda}$ as the entries. $\tilde{R}$ and $\tilde{L}$ are normalized such that $\tilde{R}\tilde{L} = I$. The flux difference in Eq.(12), i.e. the second term, can also be calculated more efficiently using:

$$\tilde{R}|\widehat{\Lambda}|\tilde{L}\delta Q = \tilde{R}|\widehat{\Lambda}|\tilde{l}\delta U = \Sigma^5_{i=1}\lambda_i\alpha_i\tilde{R}_i, \quad (14)$$

where $\tilde{l}$ is the left eigenvector of the inviscid flux of the Euler equations in the primitive variables $U = [\rho, u, v, w, p]^T$. α_i and $\tilde{R}_i$ are the characteristic variable and the eigenvector (i-th column of $\tilde{R}$), respectively, associated with the i-th eigenvalue λ_i.

Implicit Temporal Integration with Relaxation

To meet the requirement of having a large time step capability an implicit method is applied for the temporal integration of Eq.(11). The starting point is a method which is capable of solving the steady Euler/Navier-Stokes ($\Delta\tau = \infty$). Such methods, which employ relaxation techniques, have been applied to solve the Euler equations[7][26] and the Navier-Stokes equations[8].[16] The inclusion of the time derivative in these methods may in general change their stability behavior. If a strictly A-stable scheme is desired, at most a second-order accurate discretization may be applied for $\partial Q/\partial\tau$. Experience shows that the second-order scheme gives satisfactory results in most cases , thus it is adopted as the default scheme of the present method. However, third-order accurate backward differencing for $\partial Q/\partial\tau$, which is defined as stiffly stable by Gear,[11] is also applied.

The residual at a time level is defined by Eq.(11). The time derivative $\partial Q/\partial\tau$ is approximated using backward differences:

$$\frac{\partial Q}{\partial\tau} \approx \frac{1}{\Delta\tau}(c_0\Delta Q^n + c_1\Delta Q^{n-1} + c_2\Delta Q^{n-2}), \quad (15)$$

and $\partial h/\partial\tau$ is calculated from the contravariant grid speed using the geometric conservation law (GCL) relation:[19]

$$\frac{\partial h}{\partial\tau} = -\frac{\partial\widehat{\xi}_t}{\partial\xi} - \frac{\partial\widehat{\eta}_t}{\partial\eta} - \frac{\partial\widehat{\zeta}_t}{\partial\zeta}. \quad (16)$$

ΔQ^n is defined as $(Q^{n+1} - Q^n)$. The coefficients of backward differences for a variety of orders of accuracy are:

Accuracy	c_0	c_1	c_2
1st	1	0	0
2nd	3/2	-1/2	0
3rd	11/6	-7/6	2/6

Inserting Eq.(15) and Eq.(16) into Eq.(11) results in a set of nonlinear algebraic equations which is solved using Newton's method:

$$\frac{\partial R(Q^p)}{\partial Q}\Delta Q^p = -R(Q^p, Q^n, Q^{n-1}, Q^{n-2}), \quad (17)$$

where p is the sub-iteration level, Q^p is the approximation to Q^{n+1} and ΔQ^p is $Q^{p+1} - Q^p$. It should be noted that the accuracy of the sub-iteration scheme is determined only by the right-hand side (RHS), while the left-hand side (LHS) determines the rate of convergence. Quadratic convergence of Newton's method is obtained if the LHS is the exact Jacobian of the RHS and the resulting system of equations is inverted exactly to obtain the correction. In the present method an approximate LHS is employed for the relaxation:

- A first-order accurate Jacobian is always applied, regardless the order of the accuracy of the residual. This results in a block-tridiagonal matrix along a coordinate direction and a diagonally dominant Jacobian, thus facilitates the relaxation approach.
- An approximate inviscid flux Jacobian with respect to the conservative variable is also applied:

$$A^{\pm} = \frac{1}{2}A(Q^{\pm}) \pm \frac{1}{2}|A(\bar{Q})|. \quad (18)$$

This form neglects the dependency of $A(\bar{Q})$ on $Q^{\pm}$. Even simpler forms might also be used[24][29] but turned out to be much less robust or otherwise less efficient than Eq.(18).

It was expected that due to these approximations the convergence would be slower but numerical experiments hardly revealed performance degradation and the method turned out to be very robust.

Relaxation A variety of relaxation methods to solve Eq.(17) have been studied. In the two-dimensional version a line relaxation is applied with an update of Q in each sweep. The sweeping is in ξ-direction while the equations along ζ lines are solved directly. The direct inversion should resolve stiffness due to the mesh stretching and implicitly handles the viscous terms which have an elliptic behavior. In the three-dimensional version the line relaxation is retained which leaves some possibilities for the sweep direction from which the parallelization might benefit:

- Forward-backward sweep in the planes of constant ξ or η while in each plane the two-dimensional method is applied. The variant of this scheme is using a zebra relaxation in each plane to take the advantage of running in a vector computer. For this scheme the update of Q is carried out after each sweep.
- Forward-backward sweep along the planes of constant $(\xi+\eta)$. This scheme is called here Line LU-SGS since equations along the ζ lines are inverted. The original LU-SGS scheme[29] sweeps along the planes of constant $(\xi+\eta+\zeta)$ and only a block diagonal matrix is inverted.

Another method is using a point-relaxation with red-black (RB) ordering. Red and black are used to call computational cells with $(\xi+\eta+\zeta)$ odd and even, respectively.

The sub-iteration is stopped once $|R^{n+1,p}|$ drops below a prescribed value. This criterion is applied instead of $|R^{n+1,p}|/|R^{n+1,0}|$[15] upon observing that sometimes in the beginning of a sub-iteration the value of the unsteady residual was already very small, so that setting the convergence criterion relative to this level would lead to an unnecessary strict condition.

Mesh deformation To follow the motion of the solid surface the mesh has to deform accordingly. The outer boundary is held fixed and the solid boundary is attached to the structure using the volume spline method.[14] To deform the mesh smoothly the spring analogy technique[3] has been adopted, in which the mesh points are connected by springs with stiffness coefficients inversely proportional to the third power of the length of the mesh segments. The original solution method was modified to safely accept large deformation in each time step.[19]

Turbulence model The equation for the Spalart-Allmaras turbulence model is solved in a decoupled manner from the governing flow equations, using a similar discretization method. From the two-dimensional study comparing simultaneous and separate solution methods for the turbulence model no significant difference was observed. The advantage of a separate method is the possibility to apply different relaxation strategies. It is known that the one-equation model of Spalart-Allmaras is stiffer than the flow equations. The solution method should be adapted for this behavior without sacrificing the efficiency of the solution method of the flow equations.

Coupling Procedures

The difficultly in solving Eq.(6) is the dependency of C_A which is a nonlinear function of x. A variety of methods were developed to solve this problem. Bendiksen[5] solved the aeroelastic equations in a fully coupled way using an explicit Runge-Kutta method. This is probably the ideal way according to the physical modeling. The main drawback is the small time step which has to be taken due to stability limits of the explicit method. This also has repercussions for the mesh to be regenerated/deformed and subsequently the calculation of metrics. The turn-around time of this method might be prohibitive for practical problems. An implicit method, which allows larger time steps needs an iterative scheme to solve Eq.(6) and Eq.(1) simultaneously. The scheme may be of explicit type (using dual time stepping, e.g.[1]) or implicit type (e.g.[27] although an explicit fluid-structure coupling is applied during the iteration). These methods certainly need mesh updates during the iterations in each time step. This reduces the advantage of these methods. The so-called loosely-coupled/staggered/partitioned method[9][10] is more widely accepted for solving Eq.(9) due to the clear separation of the solution of aerodynamic and structural equations. The method proceeds by extrapolation of either the fluid or structural state to be used as an input for the other field. This method has the advantage that it is suited for most of the methods developed for the flow and structural parts, and in each time step a limited number of meshes have to be regenerated/deformed. A consideration in introducing mesh updates during the sub-iteration is whether having n updates in the sub-iteration with time step $\Delta\tau$ is more efficient than one update with time step $\Delta\tau/n$. Various strategies were tested for two-dimensional as well as three-dimensional transonic flow cases[20] which concludes that one mesh update in each time step performs quite well, even for large time steps. The methods presented in[20] are employed in the present work.

The second-order differential equation, Eq.(8) can be brought into a standard state-space representation as a system of first-order differential equations:

$$\dot{X} = AX + BQ, \tag{19}$$

where:

$$A = \begin{bmatrix} 0 & 1 \\ -\omega^2 & -2\zeta\omega \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ q_\infty S \end{bmatrix}, \tag{20}$$

and the state variable is defined as $X = [q, \dot{q}]^T$. It should be noted that in general , the q should be considered as a vector of modes q's and an entry of a matrix in Eq.(20) as the entry of a diagonal sub-matrix. When

the aerodynamic forces do not depend on x, a standard method can be use to solve Eq.(19):

$$\begin{aligned} X^{n+1} &= \Phi X^0 + \int_0^{t(n+1)} \Phi(t-\tau)BU(\tau)d\tau \\ &\approx \Phi(\Delta t)X^n + \Theta(\Delta t)\bar{U}, \end{aligned} \tag{21}$$

where $\bar{U}$ is a representative value of $U(t)$ between time level (n) and $(n+1)$. Calculation of Φ and Θ can be found in e.g.[9],[19] In the present method $\bar{U}$ is approximated by $U^{n+\frac{1}{2}}$. Three approaches are employed to define $U^{n+\frac{1}{2}}$:

Aerodynamic extrapolation The aerodynamic force is expressed as: $U = U(Q, X)$ and the extrapolation to time level $(n+\frac{1}{2})$ is:

$$U^{n+\frac{1}{2}} \approx U^n + (\frac{\partial U}{\partial Q}\frac{\partial Q}{\partial t} + \frac{\partial U}{\partial X}\frac{\partial X}{\partial t})\frac{\Delta t}{2}. \tag{22}$$

$\partial Q/\partial t$ and $\partial X/\partial t$ are readily available data while $\partial U/\partial Q$ and $\partial U/\partial X$ have to be calculated.

Structural extrapolation To obtain the aerodynamic force at time level $(n+\frac{1}{2})$, the state of the mesh is first approximated as:

$$\begin{aligned} x^{n+\frac{1}{2}} &\approx x^n + \dot{x}^n \Delta t/2 \\ \dot{x}^{n+\frac{1}{2}} &\approx \dot{x}^n + \ddot{x}^n \Delta t/2 \end{aligned} \tag{23}$$

$\dot{x}^n$ is readily available while $\ddot{x}^n$ is approximated simply as $(\dot{x}^n - \dot{x}^{n-1})/\Delta t$. Using this data a mesh is generated and the surface velocity is used to enforce the boundary condition. Thus the aerodynamic part of the method marches at a time level between the structural states. It should be noted that a similar method was introduced by Farhat[10] for a different purpose.

Prognostic method This method is essentially the refinement/generalization of the two aforementioned methods. In each time step the structural or aerodynamic part is extrapolated to the next time level by:

$$\begin{aligned} X^{n+\frac{1}{2}} &\approx P(X, U, t^{n+\frac{1}{2}}) \text{ or} \\ U^{n+\frac{1}{2}} &\approx P(U, X, t^{n+\frac{1}{2}}). \end{aligned}$$

Here $P(\#, t)$ denotes the approximation of the time trace $\{\#_n, \#_{n-1} \cdots\cdots \#_{n-m}\}$ at t which should be obtained by performing one of the analysis methods as presented in.[13] m denotes the number of retarded time steps in the time domain. In this method an analysis of the time trace, to determine the damping and frequency contents, which is usually done in the postprocessing stage after the simulation, is carried out during the simulation. It is obvious that as soon as the function P is not changing anymore the simulation can be stopped since the following time steps will not present any additional information. This is also an ideal stopping criterion for the simulation.

Parallelization Strategy

The solution method:i.e. the flow equations, the turbulence model and the spring equation used to deform the mesh, are parallelized using a domain decomposition approach. The equation for the structure is not parallelized because the degree of freedom is too small for making the overhead cost of parallelization worthwhile. The adoption of domain decomposition strategy, instead of data parallelization, was dictated by the portability requirement. Data parallelization usually leads to a fine grain parallelization which is not suitable for networked computers. Moreover it needs a special compiler (HPF). The current implementation has been ported to a variety of computers, from PC up to MPP and PVP supercomputers.

To execute the relaxation in each sub-domain in parallel the dependency among the sub-domains has to be frozen. This means that some boundary values are treated explicitly. Consequently, in contrast with an explicit method, the convergence will decrease as the number of sub-domains increases.

Domain decomposition The computational domain is divided into sub-domains where each sub-domain is assigned to a processor/node. The sub-domains are created by dividing the mesh only in ξ and η directions, i.e. not in body surface normal direction. From numerical experimentation it turned out as was expected that dividing the mesh in ζ direction results in a significant penalty in convergence. This probably can be explained from the usually highly stretching of the mesh in ζ direction while also the viscous term behave in an elliptic manner. Thus the ζ lines are kept as a whole.

The decomposition is carried out statically prior to executing the code. The difference in boundary condition for each sub-domain usually does not contribute significantly to the computing time. Thus e.g. for an uniform set of processors the mesh is divided equally among the nodes. For a nonuniform set of processors the mesh is divided according to the processor speed.

Explicit sub-domain coupling At the boundary of each sub-domain the halo cells will have two possible conditions: a physical boundary condition, i.e. solid surface, freestream, etc. or an artificial boundary condition, i.e. the flow variable of its neighboring domain. For maintaining second-order spatial accuracy two layers of halo cells are needed for the artificial boundary, see Figure 1. The flow state at the halo cells of the artificial boundary are transfered from the neighboring sub-domain using communication routines. In this manner the sub-domain coupling is conservative.

For the flow equations the possible coupling variable between the sub-domains are Q and ΔQ. In the first case the interchange occurs after the Q has been updated (using Newton's method), so that the linear step works completely locally in a sub-domain. In the latter case the interchange already occurs during the linear

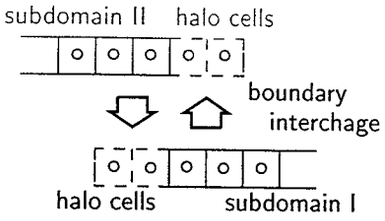

Figure 1: Artificial boundary condition between two subdomains using halo cells

step. Concerning the convergence, the transfer of ΔQ would certainly have a positive effect. Even for the RB point-relaxation there is no penalty at all if in each linear step a transfer of ΔQ is carried out. But certainly this costs some communication time. As will be shown the convergence penalty for a moderate number of processors (up to 32) is not really significant. Thus in most cases, except when it is mentioned otherwise, the results in this paper employ the transfer of Q. The solution method for the one-equation turbulence model works exactly the same as for the flow equation.

For the spring equation the mesh position is transfered to the neighbors. A slight difference exists compared to the decomposition of flow and turbulence model. The unknown for the spring equation is at the vertex, which means that there is an overlap region which is shared by two sub-domains. This overlap region is updated following the update in the field and the transfer to the neighbor. In this manner the neighboring sub-domains will will share a unique value of the mesh position in the overlap region.

Implementations The implementation of the sub-domain coupling method can be carried out conveniently using a communication library (e.g. PVM, MPI, BSP, SHMEM, etc.). There are two possibilities in communicating the interface values between two sub-domains: using message passing (a pair of send-receive routines) or using direct remote memory access (DRMA). The first method already has some sort of synchronization mechanism, while the latter needs a manual synchronization. In the present method the PVM communication library is employed since it is available on most of today computers. In the implementation on the CRAY computer (MPP T3E and PVP J910) the most critical part of the communication routine, i.e. the interchange of the boundary data, is applied using the CRAY native communication library SHMEM which uses the DRMA method.

While waiting for completing the transfer of the data between the processors, some other task can be carried out. For example since the transfer is in η and ξ direction the flux in ζ direction can readily be calculated without waiting for the neighbors. This technique reduces the portion of communication time from the whole computation.

Results

This section presents results of the validation and testing of the parallelization technique and also some results of recent aeroelastic applications.

Parallelization Performance

The purpose of the test cases in this section is to validate the decomposition method and to investigate the scalability of the method. Only a fixed mesh case is presented with a reasonable amount of mesh points.

For this purpose the standard AGARD CT5 test case is used. This case is a transonic flow about the LANN wing[6] at M_∞=0.82, α=0.60 deg oscillating about y-axis located at x=0.621 with reduced frequency k=0.102 based on root semichord and the amplitude of oscillation is 0.25 deg. For the calculation the DLR mesh[17] is employed. The topology of the mesh is CH with a size mesh points.

The calculations were carried out on the SUN MP 1000 machine (8 processors) at the Section Structures and Computational Mechanics and on the CRAY T3E of the TU Delft Center for High Performance Applied Computing (HPαC) with 72 processing elements (PE). The SUN MP is a shared memory system with 1 GB of RAM (4 byte words) while the CRAY T3E is a distributed memory system with 128 MB of RAM (8 byte words) in each PE.

The calculations were carried out with all variables in core. On the CRAY T3E at least 8 PEs have to be employed for memory and load balancing reasons. Thus the comparison for 1–8 processors is done on SUN MP and 8–64 processors on CRAY T3E. The speed-up on the CRAY T3E is calculated relative to the 8-processor performance.

Steady Flow The result of the steady flow calculations are depicted in Figures 2–4. Figure 2 shows the comparison between the result using 8 processors, the one using 64 processors and the experiment. A more comprehensive comparison of the present method with experimental data can be found in.[19] It can be concluded, from Figure 2, that the parallelization does not change the quality of the solution. Figure 3 presents the convergence history of the simulations using a variety of number of processors. The convergence penalty caused by the explicit treatment of the artificial boundary increases with the number of processors. It can be seen from the L2 norm for the mass conservation equations that this penalty is not severe even for a large number of processors. On the other hand, the history of the lift coefficient is hardly influenced by the parallelization. The speedup of running on more processors can be seen from Figure 4. This speedup was calculated from the convergence criteria of the L2 norm of the residual of mass conservation equation. The run using 64 processors took less than 4 minutes.

Unsteady Forced Vibration This section presents the applications for forced vibration cases. The calculations are carried out using 24 time steps per cycle with a varying number of sub-iterations between 12 to 20. Figure 5 and 6 again show the comparison between results using 8 processors, 64 processors and experiment. It can also be concluded that the parallelization does not alter the unsteady solutions. For the 32 and 64 processors runs, on average one additional sub-iteration is required in each time step to achieve the same convergence level as the 8 and 16 processors runs. The speedup is shown in Figure 7. The 64 processors run took about 3 minutes per cycle of oscillation.

Aeroelastic Simulation

The standard AGARD aeroelastic configuration[28] is considered. The weakened model number 3 is selected and its first 4 mode shapes are used. The configuration is simple, an isolated wing, but it requires most of the components building an aeroelastic simulation system: structural model, coupling model and mesh deformer. The mass ratio which is **only** valid at the experimental flutter point for each freestream Mach number is used throughout the simulation using Euler mode.

The simulation proceeds by first calculating the flow condition at the mean steady position. Subsequently aeroelastic responses are calculated by prescribing initial velocity conditions to all four modes. Simulations using a CH mesh of $\approx$ 80000 points have been done at various freestream Mach numbers. For each Mach number at least 3 runs were carried out for different speed indices. Each run took about 20 minutes using 8 processors of the CRAY T3E. An example of the time traces is shown in Figure 8 for M_∞=0.901. The flutter boundary is shown in Figure 9. The results are in a fair agreement with the experiment and with the full potential results of AESIM.[12] The latter method overestimates the flutter speed index at the lower Mach number.

Semi-span straked delta wing

Finally the efficiency and robustness of the EE mode of the present method is demonstrated by comparing calculated data with experimental data of the NLR wind tunnel test described in[2] for the semi-span straked delta wing.

The flow condition is M_∞=0.90, α_{mean}=6.2 deg. Time-accurate calculations have been performed on a mesh of 97×125×30 points for pitching motion about the 73% root chord with amplitude of 2.1 deg and at a reduced frequency of 0.268. †. A time step size of 96 steps per cycle were sufficient to simulate the flow. Figure 10 shows the planform and steady isobar contours at the upper side of the wing with total pressure contours behind the wing. Clearly visible are the vortices initiated by the simple strake and the tip of the outer wing panel. A comparison of calculated and experimental mean and first harmonic distributions of the pressure coefficient at selected span stations is presented in the Figures 11-13 at the same flow conditions.

It is observed that the calculation predicts the shock position more downstream over the entire wing and that the pressure level is consistently over predicted. It's remarkable that the calculated and measured shock positions are almost fixed during the oscillation. Except for the most outboard station a fairly good agreement is shown for the first harmonic data. The most pronouncing (although small) differences are the small peaks at the shock positions.

Conclusions

A method for aeroelastic simulation employing the Euler/Thin-Layer Navier-Stokes equations for three-dimensional aeroelastic applications have been described.

The method has been successfully parallelized and a satisfactory scalability up to a moderate (64) number of processors has been obtained.

The turn-around time of the method for routine aeroelastic simulation purposes has been strongly reduced.

The method has been applied to relatively simple steady and unsteady transonic flows about thick and thin wings and the complex transonic flow about an oscillating straked delta wing.

Results of the method compare fairly well with experimental data and data of other computational methods.

Acknowledgment

The authors would like to thank B.J.G. Eussen for general assistance, Prof. R.J. Zwaan for valuable discussions and encouragement, R. Hordijk and J. Hol for computing support and TU Delft Center for High Performance Applied Computing for granting CPU time on CRAY T3E and CRAY J90. Mr. Prananta's stay in The Netherlands is sponsored by the Dutch-Indonesian cooperation program through the APERT project.

References

[1] J.J. Alonso and A. Jameson. A Fully-implicit time-marching aeroelastic solution. AIAA Paper 94-0056, AIAA, 1994.

[2] A.M.Cunningham Jr., R.G. den Boer, C.S.G. Dogger, E.G.M. Geurts, A.P. Retèl, and R.J. Zwaan. Unsteady transonic wind tunnel test on a semi-span straked delta wing model oscillating in pitch. CR 93570 L parts I through III, NLR, 1993.

[3] J.T. Batina. Unsteady Euler algorithm with unstructured dynamic mesh for complex- aircraft aeroelastic analysis. AIAA Paper 89-1189, AIAA, 1989.

[4] R.M. Beam and R.F. Warming. An implicit factored scheme for the compressible Navier-Stokes equations. *AIAA Journal*, 16:393–402, 1978.

[5] O.O. Bendiksen. A new approach to computational aeroelasticity. AIAA Paper 91-0939-CP, AIAA, 1991.

†based on root chord

[6] S.R. Bland. AGARD Three dimensional aeroelastic configurations. AGARD Advisory Report AR-167, AGARD, 1979.

[7] E. Dick. Multigrid relaxation for steady Euler equations. In J. Mandel et al., editor, *Proceedings of Fourth Coper Mountain Conference on Multigrid Methods*, Copper Mountain, April 1989.

[8] E. Dick and J. Steelant. Multigrid solution of the steady compressible Navier-Stokes equations coupled to the k-ϵ turbulent equations. AIAA Paper 95-1669, AIAA, 1995.

[9] J.W. Edwards, R.W. Bennett, W. Whitlow Jr, and D.A. Seidel. Time marching transonic flutter solutions including angle of attack effects. AIAA Paper 82-0685, AIAA, 1982.

[10] C. Farhat and M. Lesoinne. On the accuracy, stability and performance of the solution of three-dimensional nonlinear transient aeroelastic problems by partitioned procedures. AIAA Paper 96-1388-CP, AIAA, 1996.

[11] C.W. Gear. *Numerical Initial Value Problem in Ordinary Differential Equations*. Series in automatic computation. Prentice-Hall, Englewood Cliffs, 1971.

[12] M.H.L. Hounjet and B.J.G. Eussen. Outline and application of the NLR aeroelastic simulation method. *Proceedings of 19th Congress of ICAS*, number 94-9.4.2, pages 1418–1441, Anaheim, 1994. ICAS.

[13] M.H.L. Hounjet, B.J.G. Eussen, and M. Soijer. Analysis of aeroelastic simulations by fitting time signals. *Proceedings of 1997 CEAS International Forum on Aeroelasticity and Structural Dynamics*, volume 3, pages 131–141, Rome, June 1997. Associazione Italiana di Aeronautica ed Astronautica.

[14] M.H.L. Hounjet and J.J. Meijer. Evaluation of Elastomechanical and Aerodynamic Data Transfer Methods for Non-planar Configurations in Computational Aeroelastic Analysis. *Proceedings of 1995 CEAS International Forum on Aeroelasticity and Structural Dynamics*, pages 11.1–11.24, Manchester, June 1995. Royal Aeronautical Society.

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

[16] B. Koren. *Multigrid and defect correction for the steady Navier-Stokes equations application to aerodynamics*. Dissertation, TU Delft, Delft, May 1989.

[17] D. Neisius. The importance of geometric modelling of elastically oscillating wings in unsteady transonic CFD calculations. *Proceedings of 1995 CEAS International Forum on Aeroelasticity and Structural Dynamics*, pages 11.1–11.24, Manchester, June 1995. Royal Aeronautical Society.

[18] R. Peyret, H. Viviand, and J.J. Smolderen. Computation of viscous compressible flows based on the Navier-Stokes equations. AGARDograph 212, AGARD, 1975.

[19] B.B. Prananta and M.H.L. Hounjet. Aeroelastic Simulation with Advanced CFD Methods in 2-D and 3-D Transonic Flow. *Proceedings of 1996 Unsteady Aerodynamics Conference*, pages 7.1–7.14, London, July 1996. Royal Aeronautical Society.

[20] B.B. Prananta and M.H.L. Hounjet. Large time step aero-structural coupling procedures for aeroelastic simulation. *Proceedings of 1997 CEAS International Forum on Aeroelasticity and Structural Dynamics*, volume 2, pages 63–71, Rome, June 1997. Associazione Italiana di Aeronautica ed Astronautica.

[21] B.B. Prananta, M.H.L. Hounjet, and R.J. Zwaan. A Thin-Layer Navier Stokes solver and its applications for aeroelastic analysis of an airfoil in transonic flows. *Proceedings of 1995 CEAS International Forum on Aeroelasticity and Structural Dynamics*, pages 15.1–15.15, Manchester, June 1995. Royal Aeronautical Society.

[22] A. Rizzi. Numerical implementation of solid body boundary conditions for the Euler equations. *ZAMM*, 58:301–304, 1978.

[23] P.L. Roe. Approximate Riemann solver, parameter vector, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.

[24] C.L. Rumsey and W.K. Anderson. Some numerical and physical aspects of unsteady Navier-Stokes computations over airfoils using dynamic meshes. AIAA Paper 88-0329, AIAA, 1988.

[25] P.R. Spalart and S.R. Allmaras. A one-equation turbulence model for aerodynamic flows. AIAA Paper 92-0439, AIAA, 1992.

[26] S.P. Spekreijse. *Multigrid solution of the steady Euler equations*. Dissertation, TU Delft, Delft, 1987.

[27] S. Weeratunga and E. Pramono. Direct coupled aeroelastic analysis through concurent implicit time integration on a parallel computer. AIAA Paper 94-1550-CP, AIAA, 1994.

[28] E.C. Yates Jr. AGARD standard aeroelastic configurations for dynamic reponse, I-wing 445.6. Report 765, AGARD, 1988.

[29] S. Yoon and A. Jameson. An LU-SSOR scheme for the Euler and Navier-Stokes equations. AIAA Paper 87-0600, AIAA, 1987.

Appendix

The metric functions of Eq.(2) are:

$$m_{11} = h(\frac{4}{3}\zeta_x^2 + \zeta_z^2 + \zeta_z^2) \quad , \quad m_{12} = \frac{h}{3}\zeta_x\zeta_y$$

$$m_{22} = h(\zeta_x^2 + \frac{4}{3}\zeta_y^2 + \zeta_z^2) \quad , \quad m_{23} = \frac{h}{3}\zeta_y\zeta_z$$

$$m_{33} = h(\zeta_x^2 + \zeta_y^2 + \frac{4}{3}\zeta_z^2) \quad , \quad m_{31} = \frac{h}{3}\zeta_z\zeta_x$$

$$m_4 = h(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)$$

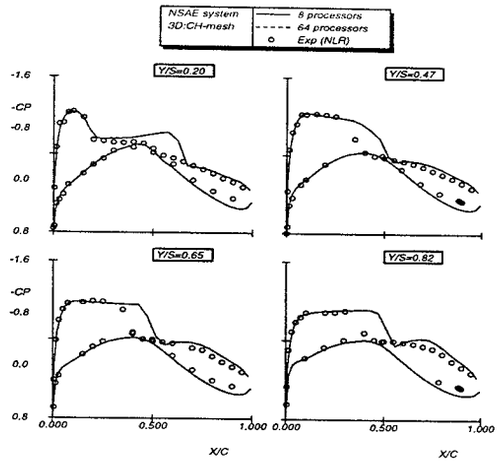

Figure 2: Mean part of pressure coefficient distributions on LANN wing, case CT5

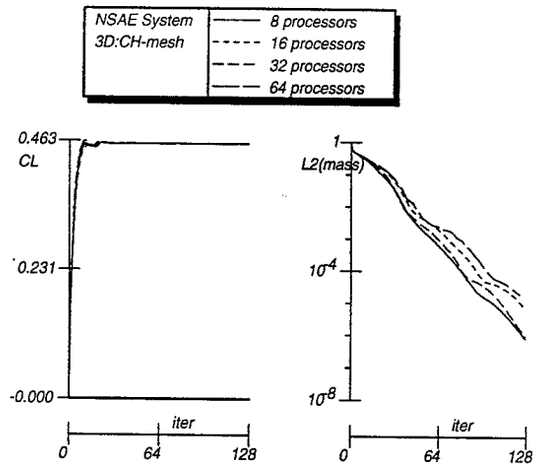

Figure 3: Convergence of lift coefficient and residual in mass equations

Figures

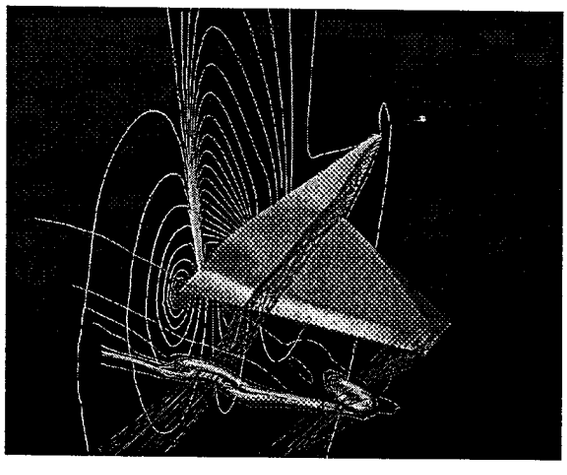

Figure 10: Steady pressure contours (EE) on a semi-span straked delta wing configuration with total pressure contours behind the wing at $M_\infty = 0.90, \alpha_m = 6.2$ deg.

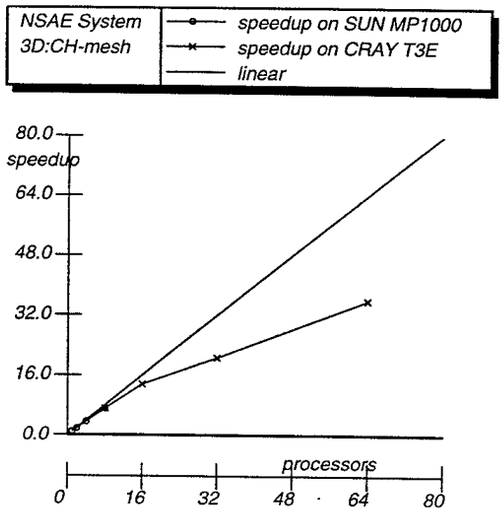

Figure 4: Performance of parallelization for steady flow calculation for LANN wing, case CT5

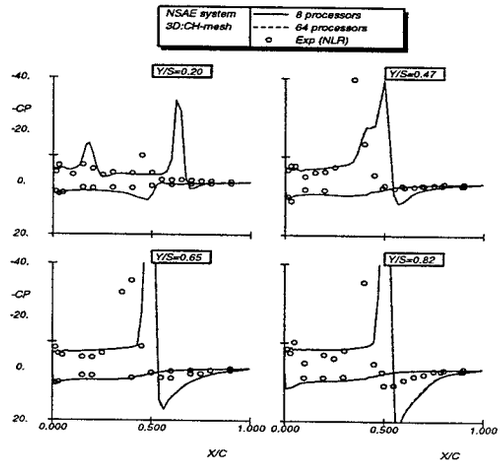

Figure 5: Real part of first harmonic pressure coefficient distributions on LANN wing, case CT5

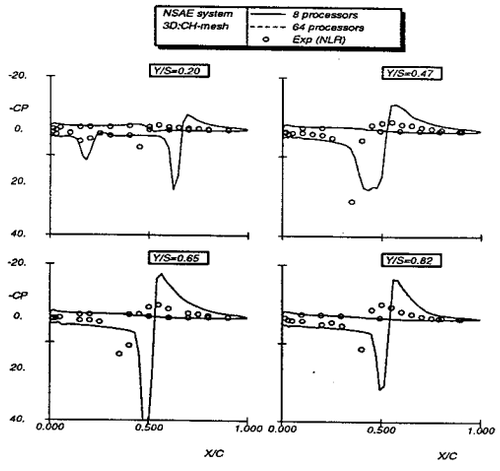

Figure 6: Imaginary part of first harmonic pressure coefficient distributions on LANN wing, case CT5

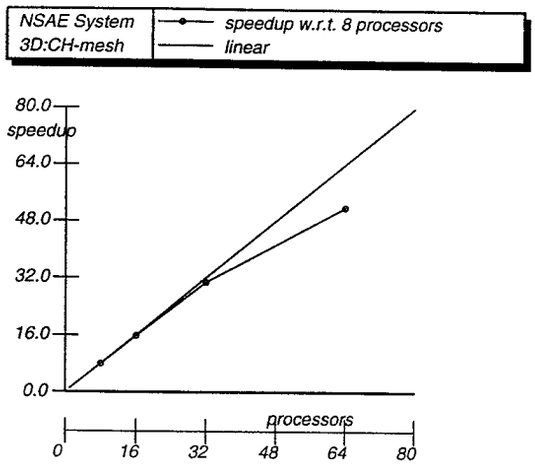

Figure 7: Performance of parallelization for unsteady flow calculation for LANN wing, case CT5

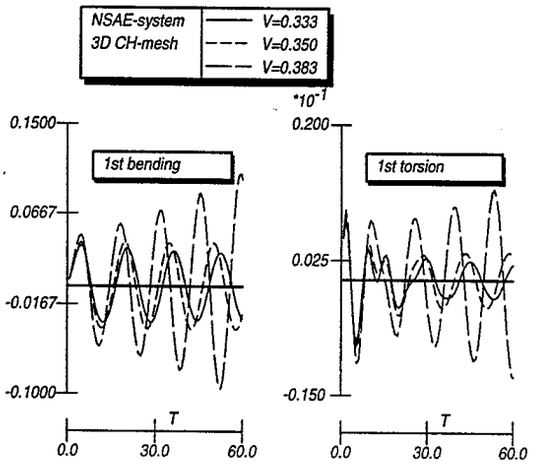

Figure 8: Comparison of time responses between several speed-indices for AGARD I-wing 445.6 at M_∞=0.901 ,$\alpha_m = 0.0$ deg, during aeroelastic simulation

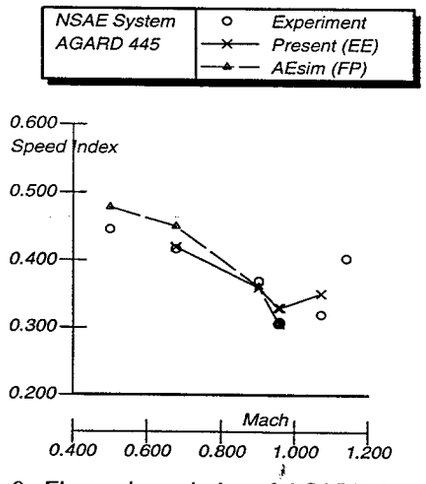

Figure 9: Flutter boundaries of AGARD I-wing 445.6

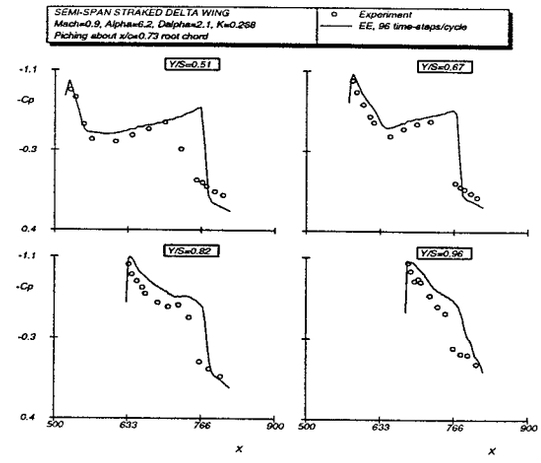

Figure 11: Comparison of mean steady pressure coefficient distributions on a semi-span straked delta wing

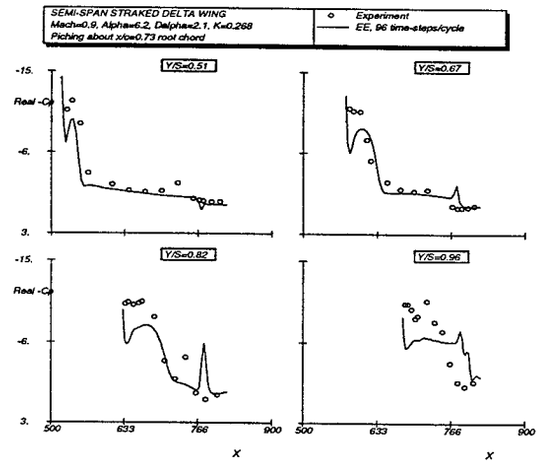

Figure 12: Comparison of real part of first harmonic pressure coefficient distributions on a semi-span straked delta wing

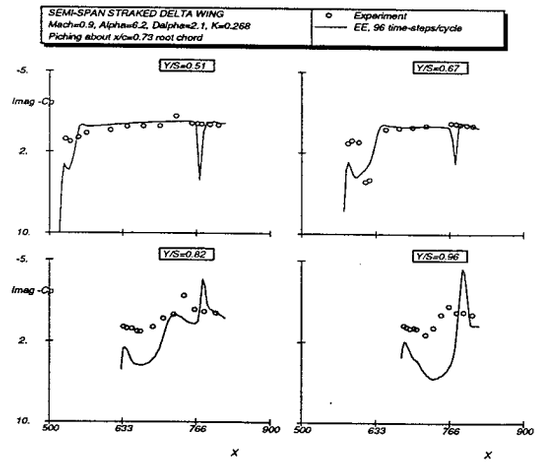

Figure 13: Comparison of imaginary part of first harmonic pressure coefficient distributions on a semi-span straked delta wing