ICAS-90-1.1.4

# DEVELOPMENT OF A THREE DIMENSIONAL UNSTEADY TRANSONIC AERODYNAMICS COMPUTER CODE FOR FLUTTER ANALYSIS

Y. S. WONG*
University of Alberta, Edmonton, Canada T6G 2G1

and

B.H.K. LEE**
National Aeronautical Establishment, Ottawa, Canada K1A OR6

## Abstract

Transonic flutter is an important aeroelastic phenomenon that occurs in the high subsonic Mach number flight regime. This paper presents the development and applications of a three-dimensional unsteady transonic aerodynamics computer code designed primarily for flutter analysis. The mathematical formulation is based on a transonic small disturbance equation, and the numerical technique employs a time-linearization approach in which the flow potentials consist of (nonlinear) steady and (linear) unsteady components. The numerical procedure has been implemented in a computer program called UST3D (**UnSteady Transonic code for a 3D** isolated wing), and the primary features of this code include: (1) An efficient fully implicit Newton-like iterative scheme for the solution of the nonlinear steady transonic equation; (2) A semi-implicit ADI scheme for the linear unsteady transonic equation; (3) New non-reflecting absorbing far-field boundary conditions for the unsteady equation. The UST3D code has been integrated into the National Aeronautical Establishment (NAE) flutter analysis program. Transonic flutter results for a fighter type aircraft are presented.

## 1. Introduction

In the last decade, there has been considerable interest in the development of computational techniques for unsteady transonic aerodynamics. This development was in response to a need to supplement the expensive wind tunnel investigations and flight tests with fast, efficient and inexpensive computer simulation codes to accurately predict flutter boundaries and other fundamental aerodynamic phenomena in the transonic regime. This is especially important for modern supercritical airfoils since the decrease in flutter speeds in the transonic flow regime is considerably more severe than that for conventional airfoils.

This paper describes the development and applications of a three-dimensional unsteady transonic aerodynamics computer simulation code on an isolated wing for aeroelastic flutter calculations.

The mathematical model for the unsteady transonic flow is based on a three-dimensional nonlinear and time-dependent transonic small disturbance (T.S.D.) equation. The assumption of small perturbations in the local flow relative to the free-stream, which is valid for small thickness, incidence and amplitude of oscillation of the body, is made. A finite-difference technique is used to solve the three-dimensional unsteady T.S.D. equation. The solution of the nonlinear, time-dependent partial differential equation can be found either directly using a time-integration technique[1-3] or indirectly by a time- linearization method.[4-5] In the time-linearization approach, the assumption is made that the flow consists of the sum of two velocity potentials representing the steady and unsteady effects. The steady potential satisfies the well-known nonlinear T.S.D. equation, while the unsteady potential is obtained through a linear time-dependent T.S.D. equation with coefficients computed from the solutions of the steady velocity potentials. The time-linearization method works best for small amplitude wing motions which constrain shocks to oscillate about the steady state positions. The time-integration method solves the nonlinear, time-dependent T.S.D. equation directly, it will model shock motions better. However, this approach is more expensive in terms of computer resources.

The numerical technique presented in this paper is based on a time-linearization method, and it is implemented in a computer simulation code called UST3D (**UnSteady Transonic code for 3D** isolated wing). The primary features of UST3D include:

(1) A new fully implicit Newton-like iterative scheme used in conjunction with a preconditioned gradient type method is developed for the nonlinear steady T.S.D. equation. The method is not only faster but also more reliable than the classical successive-linear over-relaxation algorithms.

(2) The linear unsteady T.S.D. equation is solved by a semi-implicit ADI scheme, in which the $x$-direction is treated explicitly while the $y$- and $z$-directions are discretized by an ADI type algorithm.

(3) A new non-reflecting absorbing far field boundary condition is applied for the unsteady equation

*Associate Professor, Department of Mathematics.
**Senior Research Officer, High Speed Aerodynamics Laboratory.

to improve the overall efficiency of the simulation code.

In section 2, the mathematical formulation for the three-dimensional T.S.D. equation is discussed. The computational techniques for the numerical solutions are then described in section 3. In section 4, applications of the UST3D simulation code for transonic flutter calculations are given. Conclusions are presented in section 5.

## 2. Mathematical Formulation

The choice of the T.S.D. equation is not unique. It is obtained from the full potential equation by a process of nondimensionalizing, scaling and eliminating higher order terms as the thickness ratio tends to zero. Often extra terms are included to cater for special requirements such as to handle oblique shocks more exactly. With this in mind, the three-dimensional unsteady modified T.S.D. written in conservation form is given by:

$$\frac{\partial f_0}{\partial t} + \frac{\partial f_1}{\partial x} + \frac{\partial f_2}{\partial y} + \frac{\partial f_3}{\partial z} = 0, \qquad (1)$$

where

$$\begin{aligned} f_0 &= -A\phi_t - B\phi_x, \\ f_1 &= E\phi_x + F\phi_x^2 + G\phi_y^2, \\ f_2 &= \phi_y + H\phi_x\phi_y, \\ f_3 &= \phi_z. \end{aligned}$$

The coefficients $A, B, E, F, G$ and $H$ are defined as

$$\begin{aligned} A &= M_\infty^2\kappa^2, \\ B &= 2M_\infty^2\kappa, \\ E &= 1 - M_\infty^2, \\ F &= -\frac{1}{2}(\gamma+1)M_\infty^2, \\ G &= \frac{1}{2}(\gamma-3)M_\infty^2, \\ H &= -(\gamma-1)M_\infty^2, \end{aligned}$$

where $\kappa$, $M_\infty$, $\gamma$ and $\phi(x,y,z,t)$ are the reduced frequency, free-stream Mach number, specific heat ratio and disturbance velocity potential respectively. This formulation is derived to be consistent to the first order with the full potential equation. In the standard classical T.S.D. equation, the coefficients for $H$ and $G$ are set to zero. However, the additional crossflow terms $\phi_y\phi_{xy}$ and $\phi_x\phi_{yy}$ are retained in the present formulation, to permit a better resolution of shock waves with large sweep angles.

The governing partial differential equation given in Eq.(1) is nonlinear and time-dependent. This makes computation of the flow-field troublesome since the principle of superposition is not applicable. However, under certain conditions, some of the benefits of linearity can be salvaged. In particular, Tijdeman[6] has found experimentally that time-linearization is justified for infinitesimal amplitude disturbances. For such cases the unsteady flow can be treated as a linear perturbation about a mean (nonlinear) steady state flow. This is an attractive approach since the computations can be simplified considerably and it is particularly appropriate in the context of flutter analysis. However, it should be noted that since the time-linearization approach constrains shock waves to oscillate about their steady-state positions, it is generally unsuitable for motions with oscillation amplitudes of the same order as the thickness ratio or mean angle of attack of the airfoil (whichever is larger).

Using the time-linearization approach, the solution of the T.S.D. equation is separated into its (nonlinear) steady $\phi^s(x,y,z)$ and (linear) unsteady $\phi^u(x,y,z,t)$ components, that is

$$\phi(x,y,z,t) \approx \phi^s(x,y,z) + \varepsilon\phi^u(x,y,z,t), \quad \varepsilon \ll 1 \quad (2)$$

where $\varepsilon$ is related to the amplitude of unsteady motion which is assumed to be small.

When Eq. (2) is substituted into Eq. (1), and the $\varepsilon^2$ and higher terms are neglected, the first approximation leads to a system of equations for $\phi^s$ and $\phi^u$:

$$\frac{\partial f_1^s}{\partial x} + \frac{\partial f_2^s}{\partial y} + \frac{\partial f_3^s}{\partial z} = 0, \qquad (3)$$

and

$$\begin{aligned} &A^u\phi_{tt}^u + B^u\phi_{xt}^u \\ &\quad = C^u\phi_{xx}^u + D^u\phi_x^u + E^u\phi_{yy}^u + F^u\phi_y^u + G^u\phi_{xy}^u + \phi_{zz}^u, \end{aligned} \qquad (4)$$

where

$$\begin{aligned} A^u &= M_\infty^2\kappa^2, \\ B^u &= 2M_\infty^2\kappa, \\ C^u &= (1 - M_\infty^2) - (\gamma+1)M_\infty^2\phi_x^s, \\ D^u &= -(\gamma+1)M_\infty^2\phi_{xx}^s - (\gamma-1)M_\infty^2\phi_{yy}^s, \\ E^u &= 1 - (\gamma-1)M_\infty^2\phi_x^s, \\ F^u &= -2M_\infty^2\phi_{xy}^s, \\ G^u &= -2M_\infty^2\phi_y^s. \end{aligned}$$

The steady potential $\phi^s$ satisfies Eq. (3) which is a nonlinear mixed elliptic-hyperbolic partial differential equation. Once $\phi^s$ is obtained from (3), it can be substituted into (4) to give a linear hyperbolic equation for the unsteady component $\phi^u$.

### Wing surface boundary condition

Assume the wing surface is defined by

$$z = g^{\pm}(x,y,t),$$

then the wing surface boundary condition can be described as

$$\phi_z^{\pm} = g_x^{\pm} + \kappa g_t^{\pm},$$

on $z = 0, \quad x_{le} \le x \le x_{te}, \quad y \le y_{tip}$, where $x_{le}$ and $x_{te}$ denote the $x$-coordinate of the wing leading edge and

trailing edge, respectively, and $y_{tip}$ is the $y$-coordinate of the wing tip.

Following the time-linearization procedure, we assume

$$g = g^s + \varepsilon g^u, \quad \varepsilon \ll 1.$$

Thus the wing surface boundary conditions become

$$\phi_z^{s\pm} = g_x^{s\pm}, \tag{5}$$

$$\phi_z^u = g_x^u + \kappa g_t^u. \tag{6}$$

Condition (5) is the same as the wing surface condition for the steady state problem, it includes the effects of thickness and mean angle of attack. Condition (6) represents a wing of vanishing thickness undergoing an unsteady motion with zero mean angle of attack.

Together with the conditions (5) and (6), the equations (3) and (4) are uncoupled. Thus equations (3) and (5) constitute a steady state problem which is independent of unsteady motion of the wing. Equations (3) and (5) are solved by a Newton-like iterative method, which will be discussed in the next section. Having obtained the steady state solutions, the coefficients of equation (4) are computed and the linearized three-dimensional transonic small disturbance equation (4) together with condition (6) is solved for $\phi^u$.

### Far field boundary conditions

The outer boundary conditions used for Eq. (3) are the standard far-field boundary conditions given as follows.

| | |
|---|---|
| Far upstream: | $\phi^s = 0,$ |
| Far downstream: | $\phi^s_{yy} + \phi^s_{zz} = 0,$ |
| Far above and below: | $\phi^s = 0,$ |
| Far spanwise: | $\phi^s = 0,$ |
| For the plane of symmetry: | $\phi^s_y = 0$ and $\phi^s_{xy} = 0.$ |

The standard boundary conditions for the unsteady equation (4) are given by:

| | |
|---|---|
| Far upstream: | $\phi^u = 0,$ |
| Far downstream: | $\phi^u_x + \kappa\phi^u_t = 0,$ |
| Far above and below: | $\phi^u_z = 0,$ |
| Far spanwise and for the plane of symmetry: | $\phi^u_y = 0.$ |

These boundary conditions for the unsteady T.S.D. equation presented above are, however, perfectly reflecting boundary conditions, i.e., the energy of outgoing waves will be totally reflected by the boundaries. When such boundary conditions are imposed, it is necessary to place the boundaries far away from the wing in order to prevent spurious reflections from contaminating the solution. A common technique is to use the coordinate transformation method, in which an infinite physical region is mapped into a finite computational domain. While this approach is effective for the steady state equation (3), it is unsatisfactory for the unsteady equation (4).

In this paper, a new non-reflecting absorbing boundary condition is applied for the solution of the unsteady T.S.D. equation. The construction of absorbing boundary conditions for second-order hyperbolic equations is discussed in detail by Jiang and Wong.[7] The new conditions have been developed so that wave-like solutions are permitted to propagate through the artificial computational boundaries as if there are no boundaries presented. The boundary conditions are derived from the theory of wave propagation, and it is obtained through the dispersion relation of the differential equation by requiring that the initial-boundary value problem admits wave solutions travelling in one direction only. It has been shown that local approximations of the global boundary conditions yields an $n^{\text{th}}$-order differential operators, and the optimal boundary condition satisfies a canonical factoriztion form with each factor annihilating a wave packet travelling at a given group velocity.

An important feature in the application of a non-reflecting absorbing boundary condition is that not only does it increase the accuracy of the numerical solutions, but it also reduces the computing costs since a smaller computational domain could be used. Using the approach presented in [7], the first-order absorbing far-field boundary conditions for the unsteady equation (4) are given as follows.

Far upstream and downstream:

$$\alpha\phi^u_t + \beta\phi^u_x + \gamma\phi^u_y = 0,$$

where

$$\alpha = \sqrt{\frac{B^2}{4} + AC} - \frac{B}{2} + \left(\frac{1}{v} - 1\right)\frac{AC}{\sqrt{\frac{B^2}{4} + AC}},$$

$$\beta = A\left[\left(1 - \frac{1}{v}\right)\frac{B}{2v\sqrt{\frac{B^2}{4} + AC}} - 1\right], \qquad -1 \le v < 0$$

$$\gamma = \frac{AG}{2v\sqrt{\frac{B^2}{4} + AC}},$$

for upstream and

$$\alpha = \sqrt{\frac{B^2}{4} + AC} + \frac{B}{2} + \left(\frac{1}{v} - 1\right)\frac{AC}{\sqrt{\frac{B^2}{4} + AC}},$$

$$\beta = A\left[\left(1 - \frac{1}{v}\right)\frac{B}{2\sqrt{\frac{B^2}{4} + AC}} + 1\right], \qquad 0 < v \le 1$$

$$\gamma = \frac{AG}{2v\sqrt{\frac{B^2}{4} + AC}},$$

for downstream.

Far above and below:

$$\phi^u_z + v\sqrt{A}\,\phi^u_t + \frac{Bv}{2\sqrt{A}}\,\phi^u_x = 0, \qquad (7)$$

where $0 < v \leq 1$ for above and $-1 \leq v < 0$ for below.

Far spanwise:

$$E\phi^u_y + v\sqrt{AC}\,\phi^u_t + \frac{1}{2}\Big(G + v\sqrt{\frac{E}{A}}\,B\Big)\phi^u_x = 0, \qquad (8)$$

where $0 < v \leq 1$.
For the plane of symmetry:

$$\phi^u_y = 0.$$

In the above conditions $v$ is a parameter which can be tuned to improve the absorptivity of the boundary conditions. Typical choices are $v = 1$ or $v = -1$.

## 3. Computational Methods

In this section, numerical techniques based on finite-difference approximations are applied for the solution of the T.S.D. equation (1).

The computational space and grid systems used in this report is the same as that adopted by Boppe.[8] The computational space is divided by a Cartesian grid, in which the original $(x, y, z)$ coordinates are mapped into the new $(\xi, \eta, \zeta)$ coordinates. A stretching transformation is applied, so that the original physical domain in the $x, y, z$ region with far-field boundaries corresponding to infinity becomes a finite computational domain in the $\xi, \eta, \zeta$ region. Notice that the computational space in the $x$-direction is broken up in three regions. The wing planform is located in region II, in which a uniform grid is employed. In region I and III, stretched grids are applied in the upstream and downstream directions.

### 3.1. Solution Procedure for Steady Potentials

The solution of the steady potential $\phi^s(x, y, z)$ is determined by solving the following equation:

$$C\phi_{xx} + G\phi_{xy} + E\phi_{yy} + \phi_{zz} = 0, \qquad (9)$$

where

$$C = (1 - M^2_\infty) - (\gamma + 1)M^2_\infty\phi_x,$$
$$G = -2M^2_\infty\phi_y,$$
$$E = 1 - (\gamma - 1)M^2_\infty\phi_x.$$

Notice that Eq. (9) is essentially the same as Eq. (3) but written in non-conservation form. For convenience, the superscript $s$ will be omitted.

Eq. (9) is nonlinear and of mixed elliptic-hyperbolic type, and the type of equation is determined by:

$$D = G^2 - 4CE.$$

Eq. (9) is of elliptic type (i.e., for subsonic region) if

$$D < 0,$$

and of hyperbolic type (i.e., for supersonic region) if

$$D > 0.$$

**Finite-difference approximations**

A finite-difference approximation is used to discretize Eq. (9), and the Murman-Cole type-dependent difference sheme is used for the $\phi_{xx}$ term, that is

$$\phi_{xx} = \frac{\phi_{i+1,j,k} - 2\phi_{i,j,k} + \phi_{i-1,j,k}}{\Delta x^2},$$

if the grid point $(i, j, k)$ is in subsonic region (i.e., $D < 0$), and

$$\phi_{xx} = \frac{\phi_{i,j,k} - 2\phi_{i-1,j,k} + \phi_{i-2,j,k}}{\Delta x^2},$$

if the grid point $(i, j, k)$ is in supersonic region (i.e., $D > 0$).

Central differencing is used for all other derivatives regardless of the local flow velocity, that is,

$$\phi_x = \frac{\phi_{i+1,j,k} - \phi_{i-1,j,k}}{2\Delta x},$$

$$\phi_y = \frac{\phi_{i,j+1,k} - \phi_{i,j-1,k}}{2\Delta y},$$

$$\phi_{yy} = \frac{\phi_{i,j-1,k} - 2\phi_{i,j,k} + \phi_{i,j+1,k}}{\Delta y^2},$$

$$\phi_{zz} = \frac{\phi_{i,j,k-1} - 2\phi_{i,j,k} + \phi_{i,j,k+1}}{\Delta z^2},$$

$$\phi_{xy} = \frac{\phi_{i-1,j-1,k} - \phi_{i-1,j+1,k} - \phi_{i+1,j-1,k} + \phi_{i+1,j+1,k}}{4\Delta x\Delta y}.$$

The finite-difference approximations need to be modified to take account of the far-field and wing boundary conditions, and this has been discussed in detail by Boppe.[8]

**Newton-like iterative scheme**

By the application of the finite-difference approximations, the solution of the steady transonic equation (9) is transformed to the solution of a large system of difference equations

$$\mathcal{L}(\phi) = 0, \qquad (10)$$

where $\phi$ is a vector with elements $\phi_{i,j,k}$ corresponding to the steady potential at the grid point $(i, j, k)$. Since Eq. (9) is nonlinear, the resulting system of difference equation is also nonlinear.

A Newton-like iterative scheme, which was first propsoed by Wong[9] for the nonlinear transonic full potential equation, is applied for the solution of Eq. (10). The iterative procedure is described as follows:

Let $\phi^0$ be an initial approximation for the steady potential vector, then for $n = 0, 1, 2, \ldots,$

Step 1. Compute the residual vector:

$$r^{(n)} = \mathcal{L}(\phi^{(n)}),$$

Step 2. Solve the linear system:

$$M_n \delta\phi^{(n)} = -r^{(n)}, \tag{11}$$

Step 3. Update the steady potential:

$$\phi^{(n+1)} = \phi^{(n)} + \delta\phi^{(n)}.$$

where $n$ is an iteration number, $\delta\phi$ is the correction vector and $M_n$ is a matrix operator varying from iteration to iteration. The iterative process is repeated until the residual norm $\|r^{(n)}\|$ is less than $\varepsilon$.

$M_n$ is a linear operator, which is chosen to be an approximation to the linearized transonic operator. Suppose, at the $n^{\text{th}}$ iterate, the coefficients $C, E$ and $G$ in Eq. (9) have been calculated from the value of the potential at the $(n-1)^{\text{th}}$ iteration. The result of the application of a finite-difference approximation to Eq. (9) then leads to a 12-point formula, where

$$\begin{aligned}(\mathcal{L}\phi)_{i,j,k} = {} & AC_{i,j,k}\phi_{i,j,k} + AW_{i,j,k}\phi_{i-1,j,k} \\ & + AWW_{i,j,k}\phi_{i-2,j,k} + AE_{i,j,k}\phi_{i+1,j,k} \\ & + AO_{i,j,k}\phi_{i,j+1,k} + AI_{i,j,k}\phi_{i,j-1,k} \\ & + AN_{i,j,k}\phi_{i,j,k+1} + AS_{i,j,k}\phi_{i,j,k-1} \\ & + NW_{i,j,k}\phi_{i-1,j+1,k} + NE_{i,j,k}\phi_{i+1,j+1,k} \\ & + SW_{i,j,k}\phi_{i-1,j-1,k} + SE_{i,j,k}\phi_{i+1,j-1,k}.\end{aligned} \tag{12}$$

Notice that the coefficient $AWW_{i,j,k}$ is equal to zero if the grid point $(i,j,k)$ is in subsonic region. Moreover, the coefficients $NW_{i,j,k}$, $NE_{i,j,k}$, $SW_{i,j,k}$ and $SE_{i,j,k}$ are due to the cross-term $G\phi_{xy}$ in Eq. (9), and their values are usually smaller than other coefficients for the transonic small disturbance formulation. In the present work, the operator $M_n$ is chosen by setting the coefficients $AWW_{i,j,k}$, $NW_{i,j,k}$, $NE_{i,j,k}$, $SW_{i,j,k}$ and $SE_{i,j,k}$ to zero. Hence, $M_n$ corresponds to a seven-point formula:

$$\begin{aligned}(M_n\phi)_{i,j,k} = {} & AC_{i,j,k}\phi_{i,j,k} + AW_{i,j,k}\phi_{i-1,j,k} \\ & + AE_{i+1,j,k}\phi_{i,j,k+1} + AO_{i,j,k}\phi_{i,j+1,k} \\ & + AI_{i,j,k}\phi_{i,j-1,k} + AN_{i,j,k}\phi_{i,j,k+1} \\ & + AS_{i,j,k}\phi_{i,j,k-1}.\end{aligned} \tag{13}$$

### Preconditioned minimal residual algorithm

The main computational work in the Newton-like iterative procedure given in (11) is in step 2 for the solution of the correction vector,

$$M\delta\phi = -r, \tag{14}$$

where $M$ is a large sparse matrix consisting of seven non-zero diagonals. For convenience, the subscript and superscript $n$ are omitted. It is important to have an efficient solution technique for Eq. (14), since the large system of equations has to be solved for each step in the Newon-like procedure. A direct method is not possible because it requires a large amount of storage and arithmetic operations. A classical iterative technique such as the successive line over-relaxation method is also not efficient because of the slow rate of convergence. In this paper an iterative scheme based on a minimal residual algorithm is applied. Moreover, the algorithm is used in conjunction with a preconditioning technique in order to yield a rapid rate of convergence.

The purpose of using a preconditioning technique is to accelerate the convergence rate of an iterative process. Suppose $C$ is a non-singular matrix, Eq. (14) can be rewritten as

$$MC^{-1}C\delta\phi = -r, \tag{15}$$

where $C$ is generally known as a preconditioning matrix, and Eq. (15) is called the preconditioned system for Eq. (14). If the operator $C$ is chosen so that $C^{-1}$ is a good approximation to $M^{-1}$, then the condition number of $MC^{-1}$ would be much smaller than that for $M$ itself. Since the rate of convergence for an iterative method is depended upon the condition number of the linear system, solving the preconditioned system Eq. (15) could yield a faster convergence rate than that of the original system Eq. (14). A detailed account of the construction for the $C$ matrix will be discussed shortly.

The preconditioned minimal residual algorithm is given as follows.

Let $\delta\phi^0$ be an initial guess correction vector, compute the initial residual vector for the system, i.e.

$$p^0 = -r^0 - M\delta\phi^0.$$

and solve the preconditioning system, i.e.,

$$Cz^0 = p^0.$$

Then for $k = 0, 1, 2, \ldots, \tilde{k}$, do:

Step 1. Compute the parameter:

$$\alpha_k = \frac{(p^k, Mz^k)}{(Mz^k, Mz^n)},$$

Step 2. Update the new correction vector:

$$\delta\phi^{k+1} = \delta\phi^k + \alpha_k z^k,$$

Step 3. Update the new residual vector:

$$p^{k+1} = p^k - \alpha_k M z^k,$$

Step 4. Solve the preconditioning matrix for $z^{k+1}$:

$$Cz^{k+1} = p^{k+1}.$$

In step 1, $(x,y)$ denotes the usual inner product, i.e., $(x,y) = x^T y$. Notice that, if $C = I$ (the identity matrix), then $z^k = p^k$ for all $k$ and the method becomes the basic minimal residual algorithm with no preconditioning. The main computational work per iteration in the preconditioned minimal residual algorithm is one matrix by vector multiplication for $Mz$, and solving the preconditioning system $Cz = p$.

The iterative procedure for the solution of the steady potential thus consists of outer and inner iterations: the outer is based on a Newton-like algorithm for solving the nonlinear T.S.D. equation (9), and a preconditioned minimal residual technique is applied to find the solution for the linear system Eq. (11). Since we are interested in the overall convergence for the nonlinear problem (i.e., Eq. (9)), it may not be necessary to solve the linear system in Eq. (11) to excessively accuracy for

each Newton-like iteration. In the present implementation, only an approximate solution is sought for the linear system. Hence, a small fixed number of iteration (such as $\bar{k} = 4$) will be sufficed in the preconditioned minimal residual algorithm.

### Preconditioning matrix

The preconditioning matrix $C$ plays an important role in the convergence rate of the minimal residual algorithm. For an effective preconditioning, the matrix $C$ should have the following two properties:

(1) $C$ should be chosen so that $C^{-1}$ is a good approximation to $M^{-1}$ in some sense. Consequently, the condition number of $MC^{-1}$ is much smaller than that of $M$ itself.

(2) The preconditioning step $Cz = p$ should be easily computed; otherwise, the preconditioning step will not be efficient.

A popular preconditioning algorithm which satisfies the conditions mentioned above is based on an incomplete $LU$ factorization. In this approach, $C$ is taken to be an approximation to the matrix $M$, and it is also a product of sparse triangular matrices:

$$C = LU = M + E, \tag{16}$$

where $L$ and $U$ are sparse lower and upper triangular matrices. The non-zero elements of $L$ and $U$ appear in the same locations as those in the matrix $M$, where

$$\begin{aligned}(L\phi)_{i,j,k} =& D_{i,j,k}\phi_{i,j,k} + A_{i,j,k}\phi_{i,j,k-1} \\ &+ C_{i,j,k}\phi_{i-1,j,k} + B_{i,j,k}\phi_{i,j-1,k},\end{aligned} \tag{17}$$

and

$$\begin{aligned}(U\phi)_{i,j,k} = \phi_{i,j,k} + E_{i,j,k}\phi_{i+1,j,k} + F_{i,j,k}\phi_{i,j+1,k} \\ + G_{i,j,k}\phi_{i,j,k+1}.\end{aligned} \tag{18}$$

The matrix $E$ in Eq. (16) is known as the error matrix which measures how good the approximation between $C$ and $M$ is. Since $L$ and $U$ are constructed so that their non-zero diagonals are in the same locations as those in the lower and upper triangular part of $M$, this approach is generally referred to as an incomplete $LU$ factorization preconditioning. The preconditioning techniques have been proven to be very successful for large sparse symmetric and nonsymmetric matrix computations.[10-11]

The coefficients of $L$ and $U$ are computed from the coefficients of $M$ according to the relations:

$$\begin{aligned}A_{i,j,k} &= AS_{i,j,k}, \\ B_{i,j,k} &= AI_{i,j,k}, \\ C_{i,j,k} &= AW_{i,j,k},\end{aligned}$$

$$\begin{aligned}D_{i,j,k} = (1+\alpha)(AC_{i,j,k} + AWW_{i,j,k}) - A_{i,j,k}G_{i,j,k-1} \\ + C_{i,j,k}E_{i-1,j,k} + B_{i,j,k}F_{i,j-1,k} \\ - A_{i,j,k}(E_{i,j,k-1} + F_{i,j,k-1}) \\ - B_{i,j,k}(E_{i,j-1,k} + G_{i,j-1,k}) \\ - C_{i,j,k}(F_{i-1,j,k} + G_{i-1,j,k}),\end{aligned} \tag{19}$$

$$\begin{aligned}E_{i,j,k} &= AE_{i,j,k}/D_{i,j,k}, \\ F_{i,j,k} &= AO_{i,j,k}/D_{i,j,k}, \\ G_{i,j,k} &= AN_{i,j,k}/D_{i,j,k}.\end{aligned}$$

A small value $\alpha$ is added to the main diagonal elements of $M$ to ensure the stability of the iterative scheme. However, the convergence rate is not sensitive to $\alpha$ and $\alpha = 0.05$ is used in our computer implementation. Notice that, $AWW_{i,j,k} = 0$ if the point $(i,j,k)$ is in the subsonic region, and $AWW_{i,j,k} \neq 0$ if it is in the supersonic region. However, if $AWW_{i,j,k} \neq 0$. Then the SIGN of $AWW_{i,j,k}$ is the same as that for the $AC_{i,j,k}$. Consequently, adding the coefficient $AWW_{i,j,k}$ to the $AC_{i,j,k}$ will improve the stability of the scheme.

Since $C = LU$, the preconditioning step

$$Cz = p,$$

can be computed efficiently via solving a forward substitution $Ls = p$, and then a back substitution $Uz = s$. It is worthwhile to mention that the computational work required for the preconditioning step (i.e., solve $Cz = p$) is about one matrix by vector multiplication $Mz$.

## 3.2. Solution Procedure for Unsteady Potentials

After the steady potential is obtained, the coefficients of equation (4) can be computed. The unsteady potential $\phi^u(x,y,z,t)$ can then be computed by solving the linear hyperbolic equation. In the following, we describe the algorithm and the solution procedure for solving (4). For convenience, the superscript $u$ will be omitted in the following discussion.

### Finite-difference approximations

Equation (4) is discretized by a semi-implicit finite-difference scheme, where the $x$-direction is treated explicitly while the $y-$ and $z$-direction are discretized by an ADI algorithm analogous to that of Borland and Rizzetta.[2]

**$y$-sweep:**

$$\begin{aligned}B\overleftarrow{\delta}_x \frac{\tilde{\phi} - \phi^n}{\Delta t} =& CD_x\overleftarrow{\delta}_x\phi^n + D\delta_x\phi^n + \frac{1}{2}E\delta_{yy}(\phi^n + \tilde{\phi}) \\ &+ F\delta_y\phi^n + \frac{1}{2}G\overleftarrow{\delta}_x D_y(\phi^n + \tilde{\phi}) + \delta_{zz}\phi^n,\end{aligned} \tag{20}$$

**z-sweep:**

$$A\frac{\phi^{n+1}-2\phi^n+\phi^{n-1}}{(\Delta t)^2}+B\overleftarrow{\delta}_x\frac{\phi^{n+1}-\tilde{\phi}}{\Delta t}=\frac{1}{2}\delta_{zz}(\phi^{n+1}-\phi^n), \tag{21}$$

where

$$\overleftarrow{\delta}_x\phi_{i,j,k}=\frac{\phi_{i,j,k}-\phi_{i-1,j,k}}{x_i-x_{i-1}},$$

$$\delta_x\phi_{i,j,k}=\frac{1}{2}\left(\overleftarrow{\delta}_x\phi_{i+1,j,k}+\overleftarrow{\delta}_x\phi_{i,j,k}\right),$$

$$D_x\phi_{i,j,k}=\begin{cases}\frac{\phi_{i,j,k}-\phi_{i-1,j,k}}{x_i-x_{i-1}}, & \text{if } C<0\\ \frac{\phi_{i+1,j,k}-\phi_{i,j,k}}{x_{i+1}-x_i}, & \text{if } C\geq 0\end{cases}$$

$$D_y\phi_{i,j,k}=\begin{cases}\frac{\phi_{i,j,k}-\phi_{i,j-1,k}}{y_j-y_{j-1}}, & \text{if } G<0\\ \frac{\phi_{i,j+1,k}-\phi_{i,j,k}}{y_{j+1}-y_j}, & \text{if } G\geq 0\end{cases}$$

$$\overleftarrow{\delta}_y\phi_{i,j,k}=\frac{\phi_{i,j,k}-\phi_{i,j-1,k}}{y_j-y_{j-1}},$$

$$\delta_y\phi_{i,j,k}=\frac{1}{2}\left(\overleftarrow{\delta}_y\phi_{i,j+1,k}+\overleftarrow{\delta}_y\phi_{i,j,k}\right),$$

$$\delta_{yy}\phi_{i,j,k}=\frac{2}{y_{j+1}-y_{j-1}}\left(\overleftarrow{\delta}_y\phi_{i,j+1,k}-\overleftarrow{\delta}_y\phi_{i,j,k}\right),$$

$$\overleftarrow{\delta}_z\phi_{i,j,k}=\frac{\phi_{i,j,k}-\phi_{i,j,k-1}}{z_k-z_{k-1}},$$

$$\delta_{zz}\phi_{i,j,k}=\frac{2}{z_{k+1}-z_{k-1}}\left(\overleftarrow{\delta}_z\phi_{i,j,k+1}-\overleftarrow{\delta}_z\phi_{i,j,k}\right).$$

**Semi-implicit ADI (SADI) algorithm**

The SADI algorithm is used to advance the potential $\phi$ from the time level $n$ to $n+1$ until some prescribed time is reached. At each time level, the solution is updated from $n$ to $n+1$ on each successive $y-z$ plane, marching from upstream to downstream. The value of $\phi$ on the first $y-z$ plane upstream is obtained by using upstream boundary conditions. On each subsequent $y-z$ plane, the two sweeps of the SADI algorithm are performed and the solution is obtained on this plane by solving two tridiagonal matrix equations. On the last $y-z$ plane downstream, the downstream boundary condition is applied.

**y-sweep:**

Eq. (20) leads to a system of equations of the form

$$a_j^y\tilde{\phi}_{j-1}+b_j^y\tilde{\phi}_j+c_j^y\tilde{\phi}_{j+1}=d_j^y,\quad j=2,\ldots,J_{\text{max-1}} \tag{22}$$

for each fixed $z$. The wing root and the far spanwise boundary conditions are used to provide two additional equations. Thus a completed system can be solved for $\tilde{\phi}_1,\ldots,\tilde{\phi}_{J\max}$ for each fixed $z$. To evaluate the right-hand-side $d_j^y$, the value of $\delta_{zz}\phi^n$ is needed. $\delta_{zz}\phi^n$ is the standard central differencing for all values of $z$ except $z=0$, the plane of wing surface, where a special formula must be derived because of the discontinuity of $\phi$ across $z=0$.

On the plane $z=0$ (i.e., $k=KB$), the potential on the upper and the lower surface of the wing are denoted by $\phi^U$ and $\phi^L$, respectively. In region I, which includes the region upstream of the wing and the region beyond the wing tip, the potential is continuous across $z=0$, thus

$$\phi^L=\phi^U,$$

and

$$\delta_{zz}\phi=\left(\frac{\phi_{k+1}-\phi_k}{z_{k+1}-z_k}-\frac{\phi_k-\phi_{k-1}}{z_k-z_{k-1}}\right)\frac{2}{z_{k+1}-z_{k-1}},\qquad k=KB,\quad z=0.$$

In region II which covers the wing planform, the wing surface boundary conditions are applied to evaluate $\delta_{zz}\phi$.

$$\delta_{zz}\phi=\left(\frac{\phi_{k+1}-\phi^U}{z_{k+1}-z_k}-(g_x+\kappa g_t)\right)\frac{2}{z_{k-1}-z_k},\qquad k=KB,\quad z=0^+, \tag{23}$$

$$\delta_{zz}\phi=\left((g_x+\kappa g_t)-\frac{\phi^L-\phi_{k-1}}{z_k-z_{k-1}}\right)\frac{2}{z_k-z_{k-1}},\qquad k=KB,\quad z=0^-, \tag{24}$$

where $z=0^+$ and $z=0^-$ denote the upper and lower surface of the wing, respectively. Finally in region III of wake, the wake condition

$$[\phi_z]=0,\qquad \text{across } z=0$$

is used to obtain

$$\delta_{zz}\phi=\left(\frac{\phi_{k+1}-\phi^U}{z_{k+1}-z_k}-\frac{\phi^L-\phi_{k-1}}{z_k-z_{k-1}}\right)\frac{2}{z_{k+1}-z_{k-1}},\qquad k=KB,\quad z=0. \tag{25}$$

Thus on $z=0$, two systems of equations of form (22) need to be solved, one for $\tilde{\phi}^L$ and one for $\tilde{\phi}^U$.

**z-sweep:**

After $\tilde{\phi}$ is obtained from the $y$-sweep, Eq. (21) is used to compute $\phi^{n+1}$. Because of the term $\delta_{zz}\phi^{n+1}$, Eq.(22) will take three different forms in the three different regions as described in $y$-sweep above.

In the region I, the potential is continuous, $\phi=\phi^L=\phi^U$. Eq. (21) plus the above and below boundary conditions leads to

$$a_k^z\phi_{k-1}^{n+1}+b_k^z\phi_k^{n+1}+c_k^z\phi_{k+1}^{n+1}=d_k^z,\quad k=1,\ldots,k_{\max} \tag{26}$$

where $\phi_{KB}^{n+1} = (\phi^U)^{n+1}$.

In the region II, conditions (23) and (24) are used to decouple Eq. (21) into two systems of equations for $\phi^{n+1}$ above and below the wing surface. That is, two tridiagonal matrix equations, one for $\phi_1^{n+1}, \ldots, (\phi_{KB}^L)^{n+1}$, the other for $(\phi_{KB}^U)^{n+1}, \phi_{KB+1}^{n+1} \cdots \phi_{k_{\max}}^{n+1}$ are solved.

Finally in the region III, an intermediate variable $\phi^I$ is introduced

$$\phi^I = \mu\phi^U + \nu\phi^L, \tag{27}$$

where

$$\mu = \frac{z_k - z_{k-1}}{z_{k+1} - z_{k-1}},$$

and

$$\nu = \frac{z_{k+1} - z_k}{z_{k+1} - z_{k-1}}, \quad k = KB.$$

With this new variable, Eq. (25) becomes

$$\delta_{zz}\phi = \left(\frac{\phi_{k+1} - \phi^I}{z_{k+1} - z_k} - \frac{\phi^I - \phi_{k-1}}{z_k - z_{k-1}}\right) \frac{2}{z_{k+1} - z_{k-1}}.$$

Thus Eq. (21) leads to the same system as (26) with $\phi_{KB}$ replaced by $\phi^I$, i.e., a tridiagonal system for $\phi_1, \ldots, \phi^I, \ldots, \phi_{k_{\max}}$. After $\phi^I$ is computed, $\phi^U$ and $\phi^L$ can be obtained by using the Kutta condition

$$\phi^U - \phi^L = \Gamma. \tag{28}$$

Combining (27) and (28), we have $\phi^U = \phi^I + \nu\Gamma$, and $\phi^L = \phi^I - \mu\Gamma$.

**Time-Step Consideration**

The solution is advanced from the time level $n$ to $n+1$ by SADI algorithm. Since the algorithm is semi-implicit, a CFL condition must be satisfied for numerical stability. An estimate of the CFL condition may be obtained by performing a stability analysis for a semi-implicit algorithm applied to Eq. (4) with constant coefficients, and it is found to be

$$\frac{\Delta t}{\Delta x} \leq \frac{B}{\max C}. \tag{29}$$

The allowable time-step size for stability from (29) is compatible with the time-accuracy requirement. The restriction for stability is based on the receeding-wave propagation time scale; such a time scale is necessary to resolve receeding waves accurately. In the present formulation, a large time-step size is not an important issue because Eq. (4) is not used to calculate the steady state solution; the steady state solution is obtained from equation (3).

The following tables show the number of time-steps per cycle required for stability. The theoretical value is determined from Eq. (29) and the practical value is the actual number of time-steps used in the computer code without losing the stability. The results presented in Tables 1 and 2 are for the reduced frequency $k = 0.09$.

**Table 1.** ONERA M6 Wing

| $M_\infty$ | theoretical time steps | practical time steps |
|---|---|---|
| .85 | 480 | 300 |
| .90 | 440 | 300 |

**Table 2.** Fighter type Wing

| $M_\infty$ | theoretical time steps | practical time steps |
|---|---|---|
| .85 | 640 | 350 |
| .90 | 560 | 300 |

### 3.3. Computational Results

The numerical algorithms reported in sections 3.1 and 3.2 have been implemented in a computer simulation program called UST3D. The UST3D code has been tested for the ONERA M6 wing and a fighter type wing. This paper presents some results for the fighter type wing. The computations are carried out for Mach number 0.9 at zero angle of attack, the reduced frequency $k$ for the unsteady T.S.D. equation is 0.45. The grid system is 51 by 26 by 31 in the $x-$, $y-$ and $z$-directions.

Figures 1 and 2 compare the performance of the Newton-like iterative scheme and the classical successive line over-relaxation (SLOR) method used in Ref. [8] for the steady transonic equation. It is clear from the development of the supersonic points (SUP) in the flow fields and the lift coefficient $C_L$ that the Newton-like algorithm yields a very rapid convergence rate. The improvement of the present algorithm increases for the T.S.D. equation with high Mach number and non-zero angle of attack. A typical computing time for the steady-state solution using the Newton-like scheme is about 120 seconds on Amdahl 5860 computer. It is worth mentioning that the iterative scheme given in section 3.1 can be easily vectorized. A vector version of the Newton-like algorithm has been developed, in which the incomplete $LU$ factorization preconditioning is replaced by a matrix polynomial preconditioning technique.[12] The computing time for the steady-state solution reduces to about 35 seconds on CDC Cyber 205 vector computer.

Figure 3 illustrates the effects on the application of absorbing far-field boundary conditions for the unsteady T.S.D. equation. The solid lines are the results obtained using a large grid compuation (i.e., $51 \times 26 \times 31$). It is evident that good agreements are achieved when absorbing boundary conditions are used on a smaller grid (i.e., $51 \times 26 \times 21$). In contrast, considerable differences in the real part of $C_L$ coefficients are observed when the standard reflecting boundary conditions are imposed for a small grid calculation. An attractive feature on applying absorbing boundary conditions to a smaller grid calculation is that the computing time is significantly reduced. About 30 – 35% saving on the computing time is achieved by the application of the new absorbing boundary conditions. For the wing under consideration, Figure 4 shows the spanwise unsteady $\Delta C_P$ distributions.

## 4. Application to Flutter Analysis

The UST3D code has recently been implemented into the NAE flutter analysis program. Some preliminary results are presented herein to illustrate the validity of the code by comparison with results using the doublet lattice method.[13] Some fine tunning of the code is in progress. The results given are for a fighter type aircraft and for illustration purposes in the flutter calculations, only the two flutter modes are considered.

Figure 5 shows the two mode shapes computed from NASTRAN. The bending mode at 8.32 Hz has a significant plunging component. Similarly, the torsion mode at 9.15 Hz has a large pitching component.

Each mode is separated into its plunge and pitch components along the wing elastic axis and a separate unsteady aerodynamic calculation is carried out for each of these components. This modelling of the wing assumes no chordwise bending. A virtual work calculation using the original modes is performed to obtain the generalized aerodynamic forces for flutter analysis.

In the present model the aircraft wing is divided into 18 spanwise sections. The spanwise distributions of $C_L$ and $C_M$ for the pitch component of the torsion mode are shown in Figure 6 at $M = 0.85$ and reduced frequency $k$ based on mean chord of 0.18. The doublet lattice results are also included for comparison purposes. The $\Delta C_P$ distributions for the same mode and pitch component are shown in Figure 7 for two spanwise locations. At this Mach number no shocks are formed on the airfoil. For this configuration the agreement with the doublet lattice results is fairly good.

As an example of using the UST3D code in flutter analysis, the damping and frequency plots with airspeed in knots are shown in Figure 8. The version of RAE method, without lining up values of frequency and speed, is used and the value of $k$ is 0.18. The flutter speed determined when damping goes to zero is about 700 knots. The flutter speed using doublet lattice aerodynamics is very close to this value.

## 5. Conclusions

In this paper, the development of a three-dimensional unsteady transonic aerodynamics computer program for an isolated wing is described. The simulation code is designed primarily for flutter calculations. The performance of the computer code has been illustrated by the computational results reported in the paper. The computer code is efficient and requires a modest computing time.


## Acknowledgement

The present research has been supported in part by the Department of National Defence.


## References


1 Ballhaus, W.F. and Goorjian, P.M., "Implicit Finite-Difference Computations of Unsteady Transonic Flow About Airfoils", AIAA Journal, Vol. 15, No.12, 1977, pp. 1728–1735.

2 Borland, C.J. and Rizzetta, D.P., "Nonlinear Transonic Flutter Analysis", AIAA Journal, Vol. 20, No. 11, 1982, pp. 1606–1615.

3 Goorjian, P.M., "Computations of Unsteady Transonic Flows", in "Advances in Computational Transonics", Ed. W.G. Habashi, Pineridge Press, 1985, pp. 215–256.

4 Weatherill, W.H., Sebastin, J.D. and Ehlers, F.E., "The Practical Application of a Finite Difference Method for Analyzing Transonic Flow Over Oscillating Airfoils and Wings", NASA CR–2933, 1978.

5 Ehlers, F.E. and Weatherill, W.H., "A Harmonic Analysis Method for Unsteady Transonic Flow and Its Application to the Flutter of Airfoils", NASA CR–3537, 1982.

6 Tijdeman, H., "High Subsonic and Transonic Effects in Unsteady Aerodynamics", NLR report TR–75079U, 1975.

7 Jiang, H. and Wong, Y.S., "Absorbing Boundary Conditions for Second-Order Hyperbolic Equations", NASA TM 102009, ICOMP–89–7, 1989.

8 Boppe, C.W., "Transonic Flow Field Analysis for Wing-Fuselage Configurations", NASA CR–3243, 1980.

9 Wong, Y.S., "Newton-like Minimal Residual Methods Applied to Transonic Flow Calculations", AIAA Journal, Vol. 23, No. 4, 1985, pp. 515–521.

10 Wong, Y.S., "Preconditioned Conjugate Gradient Methods Applied to Certain Symmetric Linear Systems", Intern. J. Computer Math., Vol. 19, 1986, pp. 177–200.

11 Wong, Y.S., "Preconditioned Gradient Type Methods Applied to Nonsymmetric Linear Systems", Intern. J. Computer Math., Vol. 23, 1988, pp. 141–165.

12 Wong, Y.S., "Solving Large Elliptic Difference Equations on CYBER 205", Parallel Computing, Vol. 6, 1988, pp. 195–207.

13 Giesing, J.P., Kalman, T.P. and Rodden, W.P., "Subsonic Unsteady Aerodynamcis for General Configurations. Part II, Volume 1. Application of the Doublet Lattice Method and the Method of Images to Lifting-Surface/Body Interference", AFFDL–TR–71–5 Part 1, Vol. 1, November, 1971.

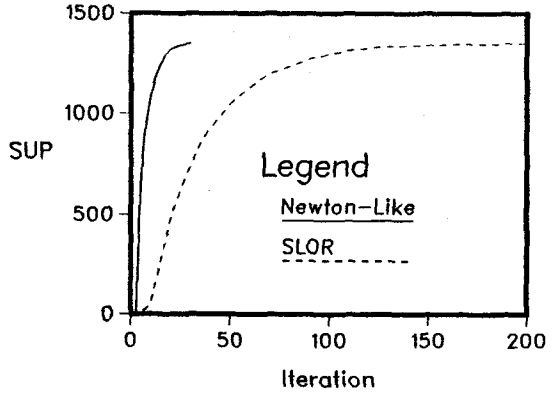


**Figure 1.** Supersonic Points versus Iterations at $M = 0.90$

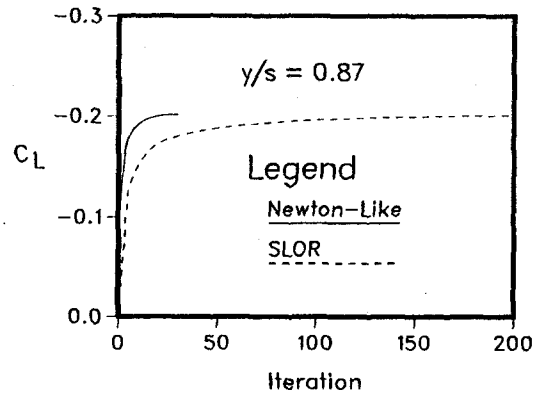


**Figure 2.** $C_L$ versus Number of Iterations at $M = 0.90$

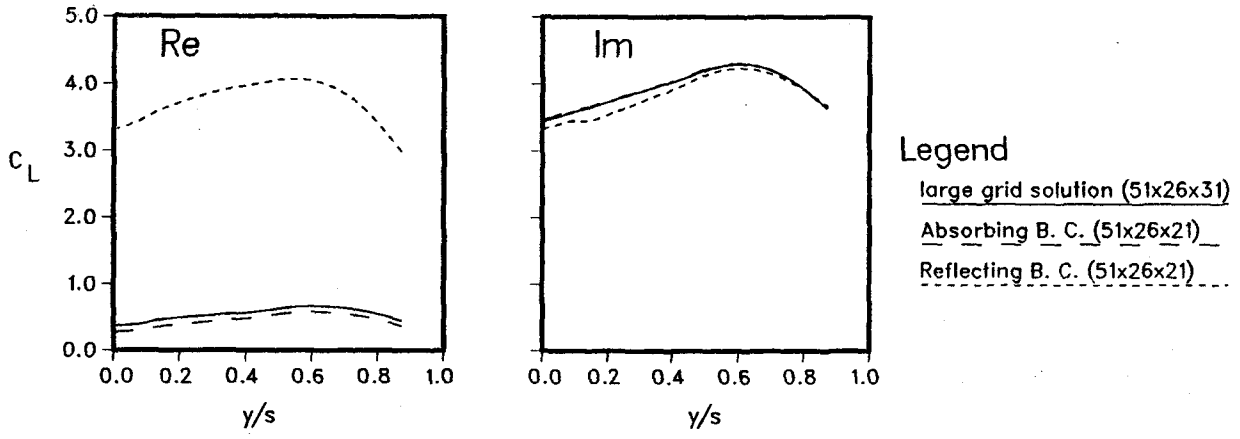


**Figure 3.** Spanwise $C_L$ Distributions at $M = 0.90$ and $k = 0.45$ (Plunging Mode)

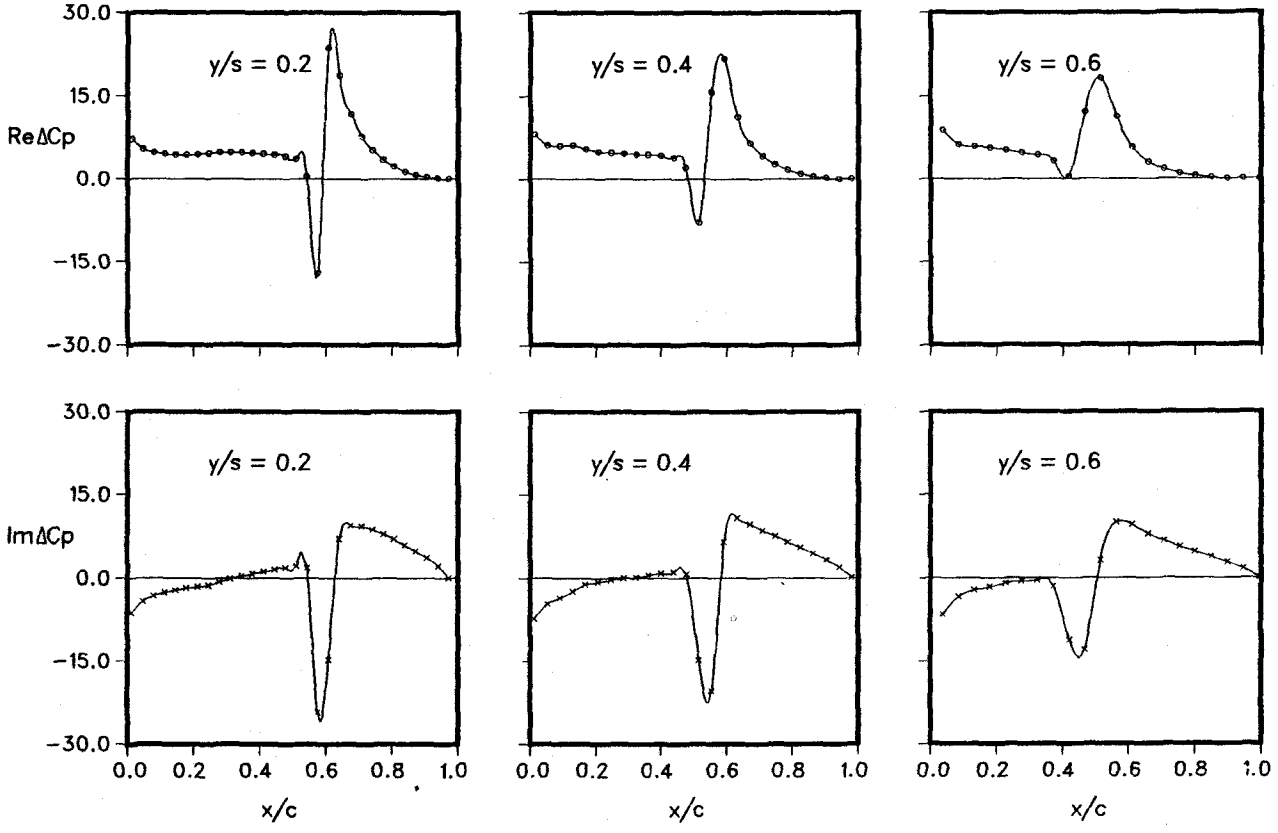


**Figure 4.** Spanwise $\Delta C_P$ Distributions at $M = 0.90$ and $k = 0.45$ (Pitching Mode)

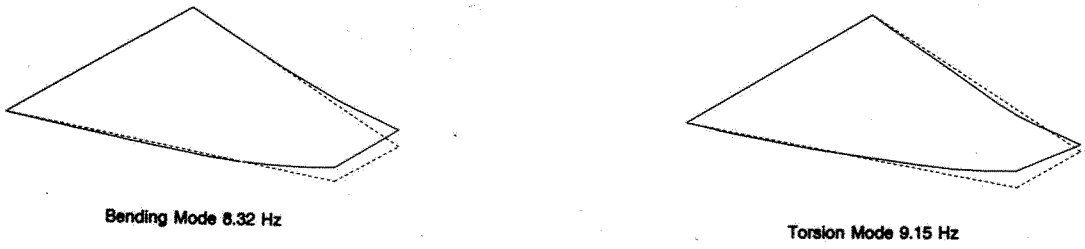


**Figure 5.** Mode Shapes

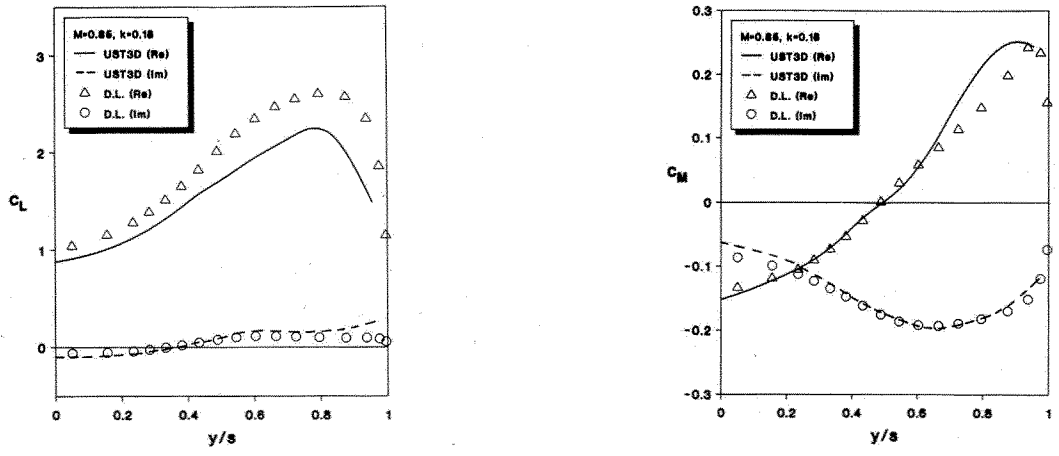


**Figure 6.** Spanwise $C_L$ and $C_M$ Distributions at $M = 0.85$ and $k = 0.18$

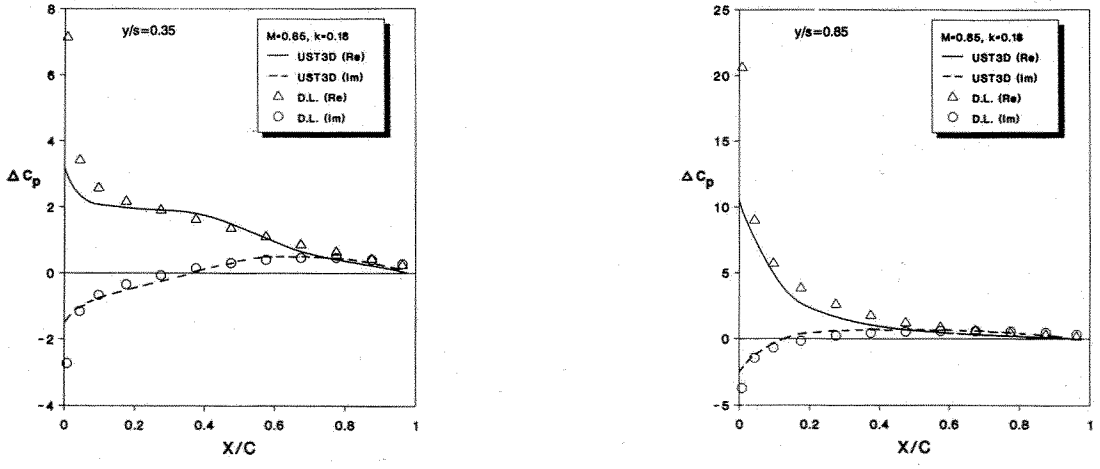


**Figure 7.** $\Delta C_P$ Distributions at $M = 0.85$ and $k = 0.18$

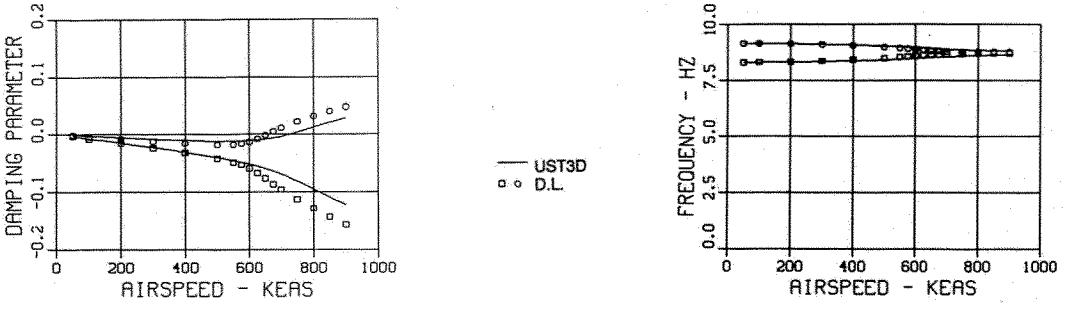


**Figure 8.** Damping and Frequency versus Airspeed