NASACR-166,251

NASA CONTRACTOR REPORT 166251

NASA-CR-166251
19820013285

Computation of Wing-Vortex Interaction in Transonic Flow Using Implicit Finite Difference Algorithm

G. Srinivasan
J. L. Steger

CONTRACT NAS2-10474
March 1981

NASA

NASA CONTRACTOR REPORT 166251

Computation of Wing-Vortex Interaction in Transonic Flow Using Implicit Finite Difference Algorithm

G. Srinivasan
J. L. Steger
Flow Simulations, Inc.
298 S. Sunnyvale Ave., Suite 204
Sunnyvale, California 94086

Prepared for
Ames Research Center
under Contract NAS2-10474

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field California 94035

ABSTRACT

An implicit delta form finite-difference algorithm for Euler equations in conservation law form has been used in preliminary calculations of three-dimensional wing-vortex interaction. Both steady and unsteady transonic flow wing-vortex interactions are computed. The computations themselves are meant to guide upcoming wind tunnel experiments of the same flow field. Various modifications to the numerical method that are intended to improve computational efficiency are also described and tested in both two- and three-dimensions. Combination of these methods can reduce the overall computational time by a factor of 4.

FOREWORD

The work reported in this document was performed for NASA-Ames Research Center under Contract NAS2-10474. During the period of performance of this contract, Mr. W. J. Chyu was the Technical Monitor.

TABLE OF CONTENTS

LIST OF SYMBOLS

Symbol	Definition
A,B,C	Jacobian matrices, see Eq. (13)
a	vortex core radius
c	characteristic length scale, chord of the airfoil
C_D	wave drag
C_p	coefficient of pressure
E	flux vector, see Eq. (2)
e	total energy per unit volume
F,G	flux vector, see Eq. (2)
H_t	total enthalpy
I	identity matrix
J	transformation Jacobian, see Eq. (9)
M_∞	free stream Mach number
p	pressure
q	unknown vector, see Eq. (1)
r	radial distance from the vortex center
U,V,W	contravariant velocity, see Eq. (8)
U_∞	free-stream velocity
u,v,w	velocity components in physical plane
v_θ	cylindrical velocity
x,y,z,t	physical plane coordinates
α	angle of attack
γ	ratio of specific heats

LIST OF SYMBOLS (concluded)

Symbol	Definition
ϵ	smoothing coefficient
ξ,η,ζ,τ	transformed plane coordinates
ξ_x, ξ_y, ..	metrics
ρ	density
σ	dimensionless frequency, $\frac{\omega c}{2U_\infty}$
ω	scaling factor, see Eq. (31); also frequency

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (concluded)

I. INTRODUCTION

Wing-vortex interaction is encountered in many practical applications in aerodynamics, and yet it is not a well understood mechanism. Helicopter rotor flow fields are a potentially rich source of vortical flow phenomenon. The interaction of a trailing vortex wake with the oncoming blades can induce unsteady blade loading and aerodynamic noise. Interest in improving helicopter flight performance has provided impetus to develop computer simulations to understand the physics of this problem.

Current numerical algorithms to compute unsteady transonic inviscid vortical flow about complex configurations are frequently either inadequate or too costly to use for routine design analysis of a large class of two- and three-dimensional flow fields. Unsteady potential theory cannot be satisfactorialy used for such analyses unless rotational flow effects are sufficiently weak and can be modeled with various circulation sheets. Numerical algorithms based on the Euler equations are suitable for any inviscid flow field simulation, but current numerical algorithms for the Euler equations have large computer time and computer storage requirements.

The present study has a two-fold objective. The first of these is to apply an existing computer code for computing rotational compressible flow field of simple wing-vortex interactions so as to better understand the flow phenomenon. These computations are expected to guide follow-on wind tunnel experiments and more efficient approximate methods. The second objective is to further the methodology and efficiency of current

numerical procedures. Then, advanced simulations of helicopter flow fields can begin as more powerful computers become available.

With the above objectives in mind, an implicit finite-difference procedure for solving the unsteady conservation law equations of inviscid flow was upgraded and applied to a simple three-dimensional wing-vortex interaction problem shown in Figure 1. Experimental data will be taken at a later date for a rectangular wing spanning the wind tunnel walls. Interacting with this wing will be an upstream vortex generated from a lifting half span wing. The preliminary computations presented here model this configuration, although the upstream vortex is analytically specified at this time.

As noted earlier, computations of three-dimensional flow fields using Euler equations generally require too much computer time. As a consequence, methods of saving computer time and distributing the grid points efficiently were studied simultaneously with the wing-vortex interaction calculations. Several methods of improving the computational efficiency of the present implicit code are demonstrated for two-dimensional flows and some are implemented in the three-dimensional code as well. In this report the governing equations are reviewed in Section 2 and the boundary conditions are discussed in Section 3. The numerical algorithm is described in Section 4, with the relevant grid system in Section 5. Results are discussed in Section 6 with concluding remarks in Section 7.

2. GOVERNING EQUATIONS

For a perfect gas neglecting viscous effects, the governing partial differential equations are the unsteady Euler equations. These can be written in a Cartesian coordinates frame of reference and in strong conservation law form for three-dimensions as follows:

$$q_t + E_x + F_y + G_z = 0 \tag{1}$$

where

$$q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \qquad E(q) = \begin{bmatrix} \rho u \\ \rho u^2+p \\ \rho uv \\ \rho uw \\ u(e+p) \end{bmatrix}$$

$$F(q) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2+p \\ \rho vw \\ v(e+p) \end{bmatrix}, \text{ and } G(q) = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2+p \\ w(e+p) \end{bmatrix} \tag{2}$$

The primitive variables of Equation (1) are the density ρ, the three mass fluxes ρu, ρv, ρw in the three coordinate directions x, y, and z, and the total energy per unit volume e. In Equation (2), p represents the pressure and is nondimensionlized by ρ_∞; density ρ by ρ_∞;

u, v, and w the velocity components in x, y, and z directions by a_∞, where a_∞ is the free stream sound speed and γ is the ratio of specific heats; and e by $\rho_\infty a_\infty^2$.

The pressure, density, and velocity components are related to the energy by the equation of state which is written for a perfect gas as

$$e = \frac{p}{\gamma-1} + \rho \left(\frac{u^2 + v^2 + w^2}{2}\right) . \tag{3}$$

In order to transform the physical flow domain into a cubical computational domain, the following independent variable transformation is employed:

$$\begin{aligned} \tau &= t \\ \xi &= \xi(x,y,z,t) \\ \eta &= \eta(x,y,z,t) \\ \zeta &= \zeta(x,y,z,t) \end{aligned} \tag{4}$$

This transformation maps the body and outer boundary surfaces onto constant coordinate planes as shown in Figure 2 and thus facilitiates the application of boundary condition procedure. The above transformation also permits the clustering of grid points in the vicinity of the body and in the regions of steep gradients in the flow.

Application of coordinate transformation, Equation (4), into Equation (1) yields the following governing partial differential equations:

$$\frac{\partial \hat{q}}{\partial \tau} + \frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} + \frac{\partial \hat{G}}{\partial \zeta} = 0 \tag{5}$$

where

$$\hat{q} = q/J$$

$$\hat{E} = (\xi_t q + \xi_x E + \xi_y F + \xi_z G)/J$$

$$\hat{F} = (\eta_t q + \eta_x E + \eta_y F + \eta_z G)/J \tag{6}$$

$$\hat{G} = (\zeta_t q + \zeta_x E + \zeta_y F + \zeta_z G)/J$$

The flux vectors $\hat{E}$, $\hat{F}$, and $\hat{G}$ all can also be written as

$$\hat{E} = J^{-1}\begin{bmatrix} \rho U \\ \rho uU + \xi_x p \\ \rho vU + \xi_y p \\ \rho wU + \xi_z p \\ (e+p)U - \xi_t p \end{bmatrix}, \quad \hat{F} = J^{-1}\begin{bmatrix} \rho V \\ \rho uV + \eta_x p \\ \rho vV + \eta_y p \\ \rho wV + \eta_z p \\ (e+p)V - \eta_t p \end{bmatrix}$$

$$\text{and} \quad \hat{G} = J^{-1}\begin{bmatrix} \rho W \\ \rho uW + \zeta_x p \\ \rho vW + \zeta_y p \\ \rho wW + \zeta_z p \\ (e+p)W - \zeta_t p \end{bmatrix} \tag{7}$$

where U, V, and W are the contravariant velocities (without metric normalization) given by

$$U = \xi_t + \xi_x u + \xi_y v + \xi_z w$$

$$V = \eta_t + \eta_x u + \eta_y v + \eta_z w \tag{8}$$

$$W = \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w$$

The metrics required by Equation (7) are obtained from a chain rule expansion of x_ξ, y_ξ, etc. and solved for ξ_x, ξ_y, etc., to yield

$$\begin{aligned}
\xi_x &= J(y_\eta z_\zeta - y_\zeta z_\eta) & \eta_x &= J(z_\xi y_\zeta - y_\xi z_\zeta) \\
\xi_y &= J(z_\eta x_\zeta - x_\eta z_\zeta) & \eta_y &= J(x_\xi z_\zeta - x_\zeta z_\xi) \\
\xi_z &= J(x_\eta y_\zeta - y_\eta x_\zeta) & \eta_z &= J(y_\xi x_\zeta - x_\xi y_\zeta) \\
\zeta_x &= J(y_\xi z_\eta - z_\xi y_\eta) & \xi_t &= -x_\tau \xi_x - y_\tau \xi_y - z_\tau \xi_z \\
\zeta_y &= J(x_\eta z_\xi - x_\xi z_\eta) & \eta_t &= -x_\tau \eta_x - y_\tau \eta_y - z_\tau \eta_z \\
\zeta_z &= J(x_\xi y_\eta - y_\xi x_\eta) & \zeta_t &= -x_\tau \zeta_x - y_\tau \zeta_y - z_\tau \zeta_z
\end{aligned} \tag{9}$$

where $J^{-1} = x_\xi y_\eta z_\zeta + y_\zeta z_\xi x_\eta + z_\eta x_\zeta y_\xi - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi$ is the transformation Jacobian.

The transformed equations given by Equation (5) are not much more complex than the original set cast in Cartesian coordinates (Eq. (1)). Strong conservation-law form of the equations is maintained for shock-capturing purposes.

3. BOUNDARY CONDITIONS

The tangency condition along the surface $\zeta(x,y,z,t)$ = constant is that $W = 0$ and is used in

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = J^{-1} \begin{bmatrix} (\eta_y\zeta_z - \eta_z\zeta_y) & -(\xi_y\zeta_z - \xi_z\zeta_y) & (\xi_y\eta_z - \xi_z\eta_y) \\ -(\eta_x\zeta_z - \eta_z\zeta_x) & (\xi_x\zeta_z - \xi_z\zeta_x) & -(\xi_x\eta_z - \xi_z\eta_x) \\ (\eta_x\zeta_y - \eta_y\zeta_x) & -(\xi_x\zeta_y - \xi_y\zeta_x) & (\xi_x\eta_y - \xi_y\eta_x) \end{bmatrix} \begin{bmatrix} U - \xi_t \\ V - \eta_t \\ W - \zeta_t \end{bmatrix} \tag{10}$$

to obtain u, v, and w.

The pressure along the body surface can be obtained from a normal momentum relation, found by combining the three transformed momentum equations

$$\begin{aligned} p_n(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)^{\frac{1}{2}} &= (\xi_x\zeta_x + \xi_y\zeta_y + \xi_z\zeta_z)p_\xi \\ &\quad + (\eta_x\zeta_x + \eta_y\zeta_y + \eta_z\zeta_z)p_\eta \\ &\quad + (\zeta_x^2 + \zeta_y^2 + \zeta_z^2)p_\zeta \\ &= \rho(\partial_\tau\zeta_t + u\partial_\tau\zeta_x + v\partial_\tau\zeta_y + w\partial_\tau\zeta_z) \\ &\quad - \rho U(\zeta_x u_\xi + \zeta_y v_\xi + \zeta_z w_\xi) \\ &\quad - \rho V(\zeta_x u_\eta + \zeta_y v_\eta + \zeta_z w_\eta) \end{aligned} \tag{11}$$

where n is the normal direction to the body surface.

At the far field boundaries free-stream values are specified except on the upstream and downstream grid boundaries. At the upstream grid boundary, an analytical vortex is initialized on the wing axis and at the downstream boundary a simple linear extrapolation is used for all the flow quantities.

To simulate wind tunnel test conditions at the ends of the wing (wing tips), spanwise velocity v is set to zero there and values of other flow quantities are set equal to the corresponding values at a previous spanwise station.

4. NUMERICAL ALGORITHM

An implicit numerical algorithm is used to solve the equations. This method is chosen to avoid restrictive stability conditions when a fine grid spacing is used to resolve flow quantities in a region of large spacial flow gradient.

4.1. Conventional Form

The basic numerical algorithm used was developed by Beam and Warming[1-3] and has been used extensively and successfully for a large class of problems.[4-11] It is second-order accurate in space and time, is noniterative, and is in a spatially factored form referred to as the "delta-form." For either trapezoidal or Euler temporal implicit differencing, the delta form scheme is given by[6]

$$
\begin{aligned}
&(I + h\delta_\xi \hat{A}^n - \varepsilon_I J^{-1}\nabla_\xi\Delta_\xi J)(I + h\delta_\eta \hat{B}^n - \varepsilon_I J^{-1}\nabla_\eta\Delta_\eta J)\times \\
&(I + h\delta_\zeta \hat{C}^n - \varepsilon_I J^{-1}\nabla_\zeta\Delta_\zeta J)(\hat{q}^{n+1} - \hat{q}^n) = -\Delta t(\delta_\xi \hat{E}^n + \delta_\eta \hat{F}^n + \delta_\zeta \hat{G}^n) \\
&- \varepsilon_E J^{-1}[(\nabla_\xi\Delta_\xi)^2 + (\nabla_\eta\Delta_\eta)^2 + (\nabla_\zeta\Delta_\zeta)^2]J\hat{q}^n
\end{aligned}
\qquad (12)
$$

where the δ's are central difference operators, Δ and ∇ are forward or backward difference operators, e.g., $\Delta_\zeta q = q(\xi, \eta, \zeta + \Delta\zeta) - q(\xi, \eta, \zeta)$. Indices denoting spatial location are suppressed for convenience and $h = \Delta t$ corresponds to Euler implicit first-order and $h = \Delta t/2$ to trapezoidal second-order time accuracy. Also, I is the identity matrix and $J\hat{q}^n = q(n\Delta t)$ where n is the number of time steps.

The Jacobian matrices $\hat{A}^n$, $\hat{B}^n$, and $\hat{C}^n$ are obtained in the time linearization of $\hat{E}^n$, $\hat{F}^n$, and $\hat{G}^n$ and can be written as

$$\hat{A},\ \hat{B},\ \text{or}\ \hat{C} = \begin{bmatrix} K_0 & K_1 & K_2 & K_3 & 0 \\ K_1\phi^2-u\theta & K_0+\theta-K_1(\gamma-2)u & K_2u-(\gamma-1)K_1v & K_3u-(\gamma-1)K_1w & K_1(\gamma-1) \\ K_2\phi^2-v\theta & K_1v-K_2(\gamma-1)u & K_0+\theta-K_2(\gamma-2)v & K_3v-(\gamma-1)K_2w & K_2(\gamma-1) \\ K_3\phi^2-w\theta & K_1w-K_3(\gamma-1)u & K_2w-K_3(\gamma-1)v & K_0+\theta-K_3(\gamma-2)w & K_3(\gamma-1) \\ \theta[2\phi^2-\gamma(e/\rho)] & [K_1\{\gamma(e/\rho)-\phi^2\} -(\gamma-1)u\theta] & [K_2\{\gamma(e/\rho)-\phi^2\} -(\gamma-1)v\theta] & [K_3\{\gamma(e/\rho)-\phi^2\} -(\gamma-1)w\theta] & K_0+\gamma\theta \end{bmatrix} \tag{13}$$

where

$$\phi^2 = 0.5(\gamma-1)(u^2 + v^2 + w^2),$$

$$\theta = K_1u + K_2v + K_3w$$

and for example, to obtain $\hat{A}$,

$$K_0 = \xi_t\ , \quad K_1 = \xi_x\ , \quad K_2 = \xi_y\ , \quad K_3 = \xi_z\ .$$

Fourth-order dissipation terms such as $\varepsilon_E J^{-1}(\nabla_\xi \Delta_\xi)^2 J \hat{q}$ in Equation (12) are added explicitly and these help to control possible numerical instabilities. The addition of the implicit second difference terms, with coefficient ε_I, operating on $(\hat{q}^{n+1} - \hat{q}^n)$ extends the linear stability bound of the fourth-order terms. For more details see Reference 6.

The solution algorithm for the approximately factored form of the implicit scheme of Equation (12) takes the following form:

$$\left.\begin{aligned} -(I + h\delta_\xi \hat{A}^n - \varepsilon_I J^{-1}\nabla_\xi \Delta_\xi J)\bar{q} &= \text{RHS} \\ (I + h\delta_\eta \hat{B}^n - \varepsilon_I J^{-1}\nabla_\eta \Delta_\eta J)\bar{\bar{q}} &= \bar{q} \\ (I + h\delta_\zeta \hat{C}^n - \varepsilon_I J^{-1}\nabla_\zeta \Delta_\zeta J)(\hat{q}^{n+1} - \hat{q}^n) &= \bar{\bar{q}} \end{aligned}\right\} \qquad (14)$$

and finally

$$\hat{q}^{n+1} = \hat{q}^n + \Delta\hat{q}^n \qquad \text{for updating the solution}$$

The central difference operators on the implicit side of Equation (12) or (14) produce block-tridiagonal matrix operators which must be inverted sequentially to obtain $\Delta\hat{q}^n = (\hat{q}^{n+1} - \hat{q}^n)$.

In implementing boundary conditions, the unknown values of $\hat{q}$ on the boundaries are updated explicitly, and $\Delta\hat{q}$ is set to zero leading to first-order error in time at the boundaries. Explicit treatment of the boundaries leads to a simple and flexible scheme, where boundary conditions become a modular element which can be put-in or pulled-out of a computer code without disturbing the implicit algorithm.

The values of ρ, U, and V along the body surface are found by linear extrapolation from above and values of u, v, and w are obtained from Equation (10). The surface pressure is obtained by integrating Equation (11). In Equation (11), the right-hand side is known from the above extrapolation process, and the basic approximate factorization algorithm is applied along the body using backward differencing in ζ and central differencing in ξ and η. Scalar tridiagonals are thus inverted in the ξ and η directions.

4.2. Diagonal Form

The solution process of implicit algorithm of Equation (12) involves the block-tridiagonal matrix solutions and this constitutes the major part of the numerical work. Equations (12) are a coupled set and thereby produce a 5 × 5 block structure for the implicit operators of Equation (12). If the operators are factored into five scalar operators, the resulting system would be more efficiently solved. This idea is used in the diagonal form of the implicit algorithm developed in Reference 12. Here, we shall briefly outline the development of this algorithm. More details can be found in Reference 12.

The Jacobian matrices $\hat{A}$, $\hat{B}$, and $\hat{C}$ (Equation (13)) have a set of eigenvalues and a complete distinct set of eigenvectors. Similarity transformations (see Warming, Beam, and Hyett, Reference 14) can be used to diagonalize $\hat{A}$, $\hat{B}$, and $\hat{C}$ where

$$\hat{A} = T_\xi \hat{\Lambda}_\xi T_\xi^{-1} \quad , \quad \hat{B} = T_\eta \hat{\Lambda}_\eta T_\eta^{-1} \quad , \text{ and } \quad \hat{C} = T_\zeta \hat{\Lambda}_\zeta T_\zeta^{-1} \tag{15a}$$

with

$$\hat{\Lambda}_\xi = D[U,U,U,U + c(\xi_x^2 + \xi_y^2 + \xi_z^2)^{\frac{1}{2}} \ , \ U - c(\xi_x^2 + \xi_y^2 + \xi_z^2)^{\frac{1}{2}}]$$

$$= \begin{bmatrix} U & 0 & 0 & 0 & 0 \\ 0 & U & 0 & 0 & 0 \\ 0 & 0 & U & 0 & 0 \\ 0 & 0 & 0 & U+c(\xi_x^2+\xi_y^2+\xi_z^2)^{\frac{1}{2}} & 0 \\ 0 & 0 & 0 & 0 & U-c(\xi_x^2+\xi_y^2+\xi_z^2)^{\frac{1}{2}} \end{bmatrix} \tag{15b}$$

$$\hat{\Lambda}_\eta = D[V,V,V,V + c(\eta_x^2 + \eta_y^2 + \eta_z^2)^{\frac{1}{2}} \ , \ V - c(\eta_x^2 + \eta_y^2 + \eta_z^2)^{\frac{1}{2}}]$$

$$\hat{\Lambda}_\zeta = D[W,W,W,W + c(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)^{\frac{1}{2}} \ , \ W - c(\zeta_x^2 + \zeta_y^2 + \zeta_z^2)^{\frac{1}{2}}]$$

where $\tilde{\theta} = \tilde{k}_x u + \tilde{k}_y v + \tilde{k}_z w$ and, for example

$$\tilde{k}_x = k_x \Big/ (k_x^2 + k_y^2 + k_z^2)^{\frac{1}{2}} \ , \quad \text{etc.} \tag{16}$$

with $k = \xi$ for $\hat{A}$, $k = \eta$ for $\hat{B}$, and $k = \zeta$ for $\hat{C}$, and $\alpha = \rho/(\sqrt{2}c)$, $\beta = 1/(\sqrt{2}\rho c)$.

Using the similarity transformations, Equation (15a), for the Jacobian matrices in Equation (12), the diagonal form of the finite-difference algorithm is written in approximate factored form as

$$T_\xi(I + h\delta_\xi \hat{\Lambda}_\xi^n - \varepsilon_I J^{-1}\nabla_\xi\Delta_\xi J)\hat{N}(I + h\delta_\eta \hat{\Lambda}_\eta^n - \varepsilon_I J^{-1}\nabla_\eta\Delta_\eta J)\hat{P}$$

$$\times (I + h\delta_\zeta \hat{\Lambda}_\zeta^n - \varepsilon_I J^{-1}\nabla_\zeta\Delta_\zeta J)(T_\zeta^{-1})\Delta\hat{q}^n = \hat{R}^n$$

$$= -\Delta t\,(\delta_\xi \hat{E}^n + \delta_\eta \hat{F}^n + \delta_\zeta \hat{G}^n)$$

$$- \varepsilon_E J^{-1}[(\nabla_\xi\Delta_\xi)^2 + (\nabla_\eta\Delta_\eta)^2 + (\nabla_\zeta\Delta_\zeta)^2]\,J\hat{q}^n \qquad (17)$$

with

$$\hat{N} = T_\xi^{-1}T_\eta\ ,\quad \hat{N}^{-1} = T_\eta^{-1}T_\xi\ ,\quad \hat{P} = T_\eta^{-1}T_\zeta,\quad \hat{P}^{-1} = T_\zeta^{-1}T_\eta \qquad (18a)$$

where

$$T_k^{-1}T_\ell = \begin{bmatrix} m_1 & m_2 & m_3 & -\mu m_4 & \mu m_4 \\ -m_2 & m_1 & m_4 & \mu m_3 & -\mu m_3 \\ -m_3 & -m_4 & m_1 & -\mu m_2 & \mu m_2 \\ \mu m_4 & -\mu m_3 & \mu m_2 & \mu^2(1+m_1) & \mu^2(1+m_1) \\ -\mu m_4 & \mu m_3 & -\mu m_2 & \mu^2(1-m_1) & \mu^2(1+m_1) \end{bmatrix} \qquad (18b)$$

with

$$m_1 = \tilde{k}_x\tilde{\ell}_x + \tilde{k}_y\tilde{\ell}_y + \tilde{k}_z\tilde{\ell}_z\ ,\qquad m_3 = \tilde{k}_x\tilde{\ell}_z - \tilde{k}_z\tilde{\ell}_x\ ,$$

$$m_2 = \tilde{k}_x\tilde{\ell}_y - \tilde{k}_y\tilde{\ell}_z\ ,\qquad m_4 = \tilde{k}_y\tilde{\ell}_z - \tilde{k}_z\tilde{\ell}_y\ ,$$

$$\mu = 1/\sqrt{2}$$

$$T_k = \begin{bmatrix} \tilde{k}_x & \tilde{k}_y & \tilde{k}_z & \alpha & \alpha \\ \tilde{k}_x u & \tilde{k}_y u - \tilde{k}_x \rho & \tilde{k}_z u + \tilde{k}_y \rho & \alpha(u + \tilde{k}_x c) & \alpha(u - \tilde{k}_x c) \\ \tilde{k}_x v + \tilde{k}_z \rho & \tilde{k}_y v & \tilde{k}_z v - \tilde{k}_x \rho & \alpha(v + \tilde{k}_y c) & \alpha(v - \tilde{k}_y c) \\ \tilde{k}_x w - \tilde{k}_y \rho & \tilde{k}_y w + \tilde{k}_x \rho & \tilde{k}_z w & \alpha(w + \tilde{k}_z c) & \alpha(w - \tilde{k}_z c) \\ \left[\tilde{k}_x \frac{\phi^2}{(\gamma-1)} + \rho\left(\tilde{k}_z v - \tilde{k}_y w\right)\right] & \left[\tilde{k}_y \frac{\phi^2}{(\gamma-1)} + \rho\left(\tilde{k}_x w - \tilde{k}_z u\right)\right] & \left[\tilde{k}_z \frac{\phi^2}{(\gamma-1)} + \rho\left(\tilde{k}_y u - \tilde{k}_x v\right)\right] & \left\{\alpha\left[\frac{\phi^2 + c^2}{(\gamma-1)} + c\tilde{\theta}\right]\right\} & \left\{\alpha\left[\frac{\phi^2 + c^2}{(\gamma-1)} - c\tilde{\theta}\right]\right\} \end{bmatrix} \tag{18c}$$

and

$$T_k^{-1} = \begin{bmatrix} \left[\tilde{k}_x\left(1 - \frac{\phi^2}{c^2}\right) + \rho^{-1}\left(\tilde{k}_z v - \tilde{k}_y w\right)\right] & \tilde{k}_x(\gamma-1)uc^{-2} & \tilde{k}_z\rho^{-1} + \tilde{k}_x(\gamma-1)vc^{-2} & -\tilde{k}_y\rho^{-1} + \tilde{k}_x(\gamma-1)wc^{-2} & -\tilde{k}_x(\gamma-1)c^{-2} \\ \left[\tilde{k}_y\left(1 - \frac{\phi^2}{c^2}\right) - \rho^{-1}\left(\tilde{k}_x w - \tilde{k}_z u\right)\right] & \left[-\tilde{k}_z\rho^{-1} + \tilde{k}_y(\gamma-1)uc^{-2}\right] & \tilde{k}_y(\gamma-1)vc^{-2} & \left[\tilde{k}_x\rho^{-1} + \tilde{k}_y(\gamma-1)wc^{-2}\right] & -\tilde{k}_y(\gamma-1)c^{-2} \\ \left[\tilde{k}_z\left(1 - \frac{\phi^2}{c^2}\right) - \rho^{-1}\left(\tilde{k}_y u - \tilde{k}_x v\right)\right] & \left[\tilde{k}_y\rho^{-1} + \tilde{k}_z(\gamma-1)uc^{-2}\right] & \left[-\tilde{k}_x\rho^{-1} + \tilde{k}_z(\gamma-1)vc^{-2}\right] & \tilde{k}_z(\gamma-1)wc^{-2} & -\tilde{k}_z(\gamma-1)c^{-2} \\ \beta(\phi^2 - c\tilde{\theta}) & \beta\left[\tilde{k}_x c - (\gamma-1)u\right] & \beta\left[\tilde{k}_y c - (\gamma-1)v\right] & \beta\left[\tilde{k}_z c - (\gamma-1)w\right] & \beta(\gamma-1) \\ \beta(\phi^2 + c\tilde{\theta}) & -\beta\left[\tilde{k}_x c + (\gamma-1)u\right] & -\beta\left[\tilde{k}_y c + (\gamma-1)v\right] & -\beta\left[\tilde{k}_z c + (\gamma-1)w\right] & \beta(\gamma-1) \end{bmatrix} \tag{18d}$$

The new implicit operators $(I + h\delta_\xi \hat{\Lambda}_\xi^n - \varepsilon_I J^{-1}\nabla_\xi \Delta_\xi J)$, etc., of the Equation (17) are still block-tridiagonal, but now the blocks are simply in diagonal form so that the operators reduce to five independent scalar tridiagonal operators. This has a large positive effect on saving the overall computational time.

The solution process for the implicit part of Equation (17) then consists of the following steps:

1. $\bar{S}_1 = (T_\xi^{-1})^n \hat{R}^n$, a matrix-vector multiplication at each grid point, since T_ξ^{-1} is known analytically;
2. $\bar{S}_2 = [I + h\delta_\xi \hat{\Lambda}_\xi - \varepsilon_I J^{-1}\nabla_\xi \Delta_\xi J]^{-1} \bar{S}_1$, five scalar tridiagonal inversions for the operators;
3. $\bar{S}_3 = \hat{N}^{-1} \bar{S}_2$, a matrix-vector multiply at each point;
4. $\bar{S}_4 = [I + h\delta_\eta \hat{\Lambda}_\eta - \varepsilon_I J^{-1}\nabla_\eta \Delta_\eta J]^{-1} \bar{S}_3$, five more scalar tridiagonal inversions;
5. $\bar{S}_5 = P^{-1} \bar{S}_4$, a matrix-vector multiply at each grid point;
6. $\bar{S}_6 = [I + h\delta_\zeta \hat{\Lambda}_\zeta - \varepsilon_I J^{-1}\nabla_\zeta \Delta_\zeta J]^{-1} \bar{S}_5$, five more scalar tridiagonal inversions;
7. $\Delta\hat{q}^n = T_\zeta^n \bar{S}_6$, another set of matrix-vector multiply; and finally,
8. $\hat{q}^{n+1} = \hat{q}^n + \Delta\hat{q}^n$ to update the solution.

This solution procedure constracts with the three block-tridiagonal inversions required in Equation (12).

In both conventional and diagonal schemes discussed above, fourth-order central difference operators are used to compute convective derivatives. Metrics are computed using second-order central difference

formulas at interior points and three-point one sided formulas are used at the boundaries. For differencing metrics in three-dimensions a free-stream error can be introduced if this differencing is not done with special weighted averages.[6] Perfect maintenance of the free-stream can be achieved by simply subtracting the free-stream fluxes from the governing equations; that is

$$\partial_\tau \hat{q} + \partial_\xi(\hat{E} - \hat{E}_\infty) + \partial_\eta(\hat{F} - \hat{F}_\infty) + \partial_\zeta(\hat{G} - \hat{G}_\infty) = 0 .$$

The metrics are not stored in the present computations because of computer storage limitations and hence have to be computed at each time step.

Equations (12) and (17) can be used for either time-accurate (unsteady) or steady state computations. For steady state calculations, $\Delta\hat{q}^n$ approaches zero asymptotically with the solution satisfying the explicit side of the equation, which is the exact steady state difference equation.

5. GRID GENERATION

Although the transformed equations, Equations (5), are somewhat more complicated than the original Cartesian form, Equations (1), they offer the following significant advantages:

(1) The boundary surfaces in the physical plane can be mapped on to rectangular surfaces in the transformed plane.

(2) The grid points can be clustered in regions that experience rapid change in the flow field gradients. This is particularly important in the present problem because of the presence of the interacting vortex and shock waves.

A surface conforming grid structure is used to simplify application of surface boundary conditions and to improve the overall accuracy of the numerical scheme. The grid is generated in two phases. Firstly, an O-type, two-dimensional grid is generated. To take advantage of the generality of the transformed equations, one needs a fairly automatic method of generating a smoothly varying grid that conforms to arbitrary bodies and allows grid point clustering. The elliptic grid generation procedure popularized by Thompson, Thames, and Mastin[15] with modifications by Steger and Sorenson[16] and Sorenson[17] was used. Source terms[15] are implemented in the elliptic grid generation equation to cluster points both at the body surface and at the center of the grid so as to resolve the approaching vortex. Additional source terms[16,17] are added in the present case to pull the grid lines towards the wing axis. The method also enforces orthogonality of mesh lines at the boundaries. Figure 3 shows several views of a two-dimensional O-type grid generated by the

above method. This grid is then distributed along the rectangular wing span using spanwise exponential clustering functions to generate a three-dimensional mesh. The grid so generated has 48 points in the periodic direction, 20 points in the radial direction, and 21 points along the span of the wing. The grid structure is coarse and its dimensions are matched to the internal computer storage capability of Control Data Corporation 7600 machine.

6. RESULTS

Preliminary numerical experiments using the conventional algorithm with the above 48 × 20 × 21 grid for a simple flow past three-dimensional wing at $M_\infty = 0.72$ and $\alpha = 0$,deg. revealed that a single solution took as much as 8 hours of computer time on CDC 7600 computer. By using the diagonal form of the algorithm, this time was reduced by a factor of 2.5 per time step without the loss of accuracy or numerical stability. The magnitude of savings in computational time reported here is much better than the previously reported studies with the diagonal algorithm.[12,13] This improvement stems from the fact that in the present application the size of the block periodic tridiagonal operator is much more significant. The diagonal algorithm is twice as efficient for periodic block tridiagonal matrices as it is for a regular block tridiagonal. Because of this big savings in computational time, all further computations were undertaken using the diagonal algorithm.

6.1. Two-Dimensional Calculations

In initial tests of the computer code several two-dimensional flow cases were computed. Instead of generating a solution at one spanwise station (for a 2-D case), a five plane solution (like a 3-D case) was computed with the spanwise index remaining active in the code. The two spanwise end planes are updated with simple symmetry conditions. The results of such a calculation are shown in Figure 4 in which a plot of computed C_p distribution for a steady nonlifting case using the present

code is compared with a reference solution.[18] This solution uses the two-dimensional grid shown in Figure 3. Figure 5 shows a C_p plot for a lifting case.

In initial calculations, two-dimensional solutions of inviscid free shear layers (jets and wakes) with their velocity field superimposed on the main stream at the grid boundary of the wing were computed. The excess velocity for the jet and the defect for the wake is assumed to be $0.2U_\infty$ on the wing axis. Figures 6 and 7 show the computed steady solution in the form of C_p plots. While the shear flows of Figures 6 and 7 are symmetric with respect to wing axis, Figure 8 shows a C_p plot for the asymmetric case. Asymmetry in the flow is created by locating the axis of the symmetric jet of Figure 8 0.01C above the wing axis.

6.2. Three-Dimensional Calculations

As noted earlier, the objective of the present study is to provide computational guidance for an upcoming wing-vortex interaction wind tunnel experiment. The aerodynamic performance of rotory wing aircraft is strongly influenced by the vortex wake shed by the rotating blades. The vortex generated by the tip of the blade is time-variant and has large strength. The interaction of this vortex with the following rotor blade will affect the flow characteristics around the blade and thus its aerodynamic loads. In the transonic flow regime this effect is more pronounced because of the shock-wave motion. Investigation of the wing-vortex interaction is thus important not only for improving the understanding of the flow around the wing but also for the optimum design of the rotor wing.

An analytical vortex aligned in the center of the wind tunnel and located initially at the grid boundary was made to impinge on the rectangular wing as shown in Figure 1. The wing was set at zero angle of attack to a free-stream of $M_\infty = 0.72$. The specified analytical vortex chosen is a steady Lamb or spreading vortex whose cylindrical velocity distribution given by[19]

$$\frac{v_\theta(r)}{U_\infty} = \frac{\Gamma}{2\pi U_\infty r}\left(1 - e^{-r^2/a^2}\right) \tag{19}$$

resembles that of an experimentally generated vortex.[20] In Equation (19), $\frac{\Gamma}{2\pi}$ is the strength of vortex; $a = (4\nu t)^{\frac{1}{2}}$ is the core radius; ν is the kinematic viscosity; t is the time; and r is the radial distance measured from the core center. The pressure field induced by this vortex is determined using the radial momentum equation

$$\frac{dp}{dr} = \frac{\rho v_\theta^2}{r} \tag{20}$$

Using the energy equation for constant enthalpy flow, Equation (20) can be written as

$$\frac{dp}{dr} = \frac{\gamma p}{\gamma - 1}\left(\frac{2}{2H_t - v_\theta^2 - q_\infty^2}\right)\frac{v_\theta^2}{r} \tag{21}$$

where H_t is the total enthalpy and $q_\infty^2 = u^2 + v^2 + w^2$. The pressure field from this and the velocity distribution from Equation (19) are superimposed on the main stream at the upstream grid boundary of the wing for initializing the vortex. Figure 9 shows a plot of typical radial velocity and pressure distribution for a Lamb vortex of strength $\frac{\Gamma}{2\pi U_\infty c} = 0.03$ at the grid

boundary. The three-dimensional computation of wing-vortex interaction is carried out using grid dimensions of 48 × 20 × 21 with 21 spanwise points. Figure 10 shows the computed steady pressure distribution on the wing plotted in the form of C_p distribution at various spanwise stations.

Because of the coarse grid, it is necessary to consider the vortex recovered slightly upstream of the wing leading edge as the input vortex for interaction purpose. Typically, the size of this vortex is of the order of the span of the wing although the vortex initialized at the grid boundary is much smaller. Figure 11 shows a spreading vortex at the grid boundary whose radial velocity distribution is given by

$$\frac{v_\theta(r)}{U_\infty} = \frac{\Gamma}{2\pi r U_\infty} \left(1 - e^{-r^2/a^2}\right) e^{-r^2/a^2}. \tag{22}$$

Steady state solution of the interaction of this vortex with the wing of Figure 1 is computed and a plot of C_p distribution at various spanwise stations is shown in Figure 12. Figure 13 shows a detailed C_p distribution at a spanwise location of 0.42 chords to the right of the centerline. These plots show that the effect of vortex interaction is to superimpose a swirl velocity on the free-stream which imparts lift and slight supercritical effects.

Preliminary unsteady computations of interaction is done by choosing a vortex with time-dependent strength whose radial velocity distribution is given by

$$\frac{\bar{v}_\theta(r,t)}{U_\infty} = \frac{\Gamma}{2\pi r U_\infty} \left(1 - e^{-r^2/a^2}\right) e^{-r^2/a^2} \left(1 - \frac{1}{2} \sin \omega t\right). \tag{23}$$

where ω is the frequency of the "breathing" motion of the vortex. Figure 14 shows a plot of this velocity distribution as a function of ωt.

Interaction of this vortex with the rectangular wing of Figure 1 is computed for two periods of "breathing" motion at a dimensionless frequency (σ) of 0.4 and a vortex strength of 0.2. This time period of two cycles is considered long enough to depict the unsteady nature of the flow field. Figure 15 shows the computed C_p distribution on the wing surface at various instants of breathing cycle for this flow field.

6.3. Further Experiments to Reduce Computer Time

As indicated previously, computational efficiency is being improved by carefully distributing grid points over the body and by using the diagonalization scheme. Two additional techniques have been demonstrated to save overall computational time further and are discussed below.

(a) Special Variation of Numerical Dissipation Coefficients

The numerical algorithm, Equation (17), uses added numerical dissipation terms to maintain stability for nonlinear equations. So as not to degrade solution accuracy, fourth-order dissipation terms are added to the explicit side of the equations and second-order terms working on $(\hat{q}^{n+1} - \hat{q}^{n})$ are added to the left hand or implicit side of the equations. These terms work well, but analysis of new upwind split schemes,[21,22] which are naturally dissipative, suggest that the dissipation terms should be weighted by the "absolute value" of the Jacobian matrices A, B, and C where $A = \frac{\partial E}{\partial q}$ etc. The advantage of using such weightings is that dissipation is automatically added as needed for variation of the flow parameters and, more crucially, the mesh size spacing.

Consider the 2-D system in transformed coordinates

$$\frac{\partial \hat{q}}{\partial \tau} + \frac{\partial \hat{E}}{\partial \xi} + \frac{\partial \hat{F}}{\partial \eta} = 0 \ . \tag{24}$$

Using the homogeneous property of flux vectors $\hat{E}(q)$ and $\hat{F}(q)$, they can be split into two parts via similarity transforms as[21]

$$\hat{E} = \hat{E}^{+} + \hat{E}^{-}$$

where

$$\hat{E}^{+} = A^{+}\hat{q} \ , \quad \hat{E}^{-} = A^{-}\hat{q} \ ,$$

and

$$A = \frac{\partial \hat{E}}{\partial \hat{q}} = X\Lambda X^{-1}, \ A^{\pm} = X \ \frac{\Lambda \pm |\Lambda|}{2} \ X^{-1}$$

Here Λ is a diagonal matrix whose elements are the eigenvalues of A. The eigenvalues of A^{+} are nonnegative and those of A^{-} are nonpositive. (In subsonic flow the eigenvalues are of mixed sign since $|u| < c$.)

With the use of split flux vectors, one-sided spatial difference approximations are possible. For example, Equation (24) can be written as[21]

$$\begin{aligned} &[I + \Delta t(\nabla_{\xi} A^{+}_{j,k}\big|^{n} + \Delta_{\xi} A^{-}_{j,k}\big|^{n} + \Delta_{\eta} B^{+}_{j,k}\big|^{n} + \Delta_{\eta} B^{-}_{j,k}\big|^{n})]\, \Delta \hat{q}^{n}_{j,k} \\ &= -\Delta t(\delta^{b}_{\xi}\hat{E}^{+}_{j,k}\big|^{n} + \delta^{f}_{\xi}\hat{E}^{-}_{j,k}\big|^{n} + \delta^{b}_{\eta}\hat{F}^{+}_{j,k}\big|^{n} + \delta^{f}_{\eta}\hat{F}^{-}_{j,k}\big|^{n}) \end{aligned} \tag{25}$$

where

$$\begin{aligned} \delta^{b}_{\xi}\hat{E} &= \frac{3\hat{E}_{j} - 4\hat{E}_{j-1} + \hat{E}_{j-2}}{2\Delta\xi} \\ \delta^{f}_{\xi}\hat{E} &= \frac{-3\hat{E}_{j} + 4\hat{E}_{j+1} - \hat{E}_{j+2}}{2\Delta\xi} \end{aligned} \tag{26}$$

are backward and forward second-order accurate one-sided difference operators, respectively.

Further, $A^{\pm} \equiv (A \pm |A|)/2$

$$|A| \equiv X|\Lambda|X^{-1}$$

$$E^{\pm} = (E \pm |A|q)/2 \text{ , etc.} \tag{27}$$

Substituting the relations (27) in (25), we can write Equation (25) as[22]

$$[I + \Delta t(\delta_\xi A^n + \delta_\eta B^n - (2\Delta\xi)^{-1}(\nabla\Delta)|A| - (2\Delta\eta)^{-1}(\nabla\Delta)|B|]\,\Delta\hat{q}^n$$

$$= -\Delta t(\bar{\delta}_\xi \hat{E}^n + \bar{\delta}_\eta \hat{F}^n + (4\Delta\xi)^{-1}(\nabla\Delta)^2|\hat{A}|q^n + (4\Delta\eta)^{-1}(\nabla\Delta)^2|B|q^n) \tag{28}$$

where $\delta_\xi = \dfrac{\nabla_\xi + \Delta_\xi}{2}$, $\bar{\delta}_\xi = \dfrac{\delta_\xi^f + \delta_\xi^b}{2}$, etc.

Calculating the absolute values of the coefficients matrices $|A|$ and $|B|$ is very costly. However, the mesh variation enters into A and B at a point as a simple scaling. This important variation can be accounted for, along with the main flow parameter, by using not $|A|$ but its spectral radius (i.e. largest eigenvalue in absolute value).

Replace $|A|$ with ρ_a and $|B|$ with ρ_b where ρ_a and ρ_b are the spectral radii. For A, the maximum eigenvalue $\rho_a = |U| + c(\xi_x^2 + \xi_y^2)^{\frac{1}{2}}$, and similarly for B, $\rho_b = |V| + c(\eta_x^2 + \eta_y^2)^{\frac{1}{2}}$. We can further approximate these, for computational simplicity, as

$$\rho_a \simeq (1 + M_\infty)a_\infty\,(|\xi_x| + |\xi_y|)$$

and (29)

$$\rho_b \simeq (1 + M_\infty)a_\infty\,(|\eta_x| + |\eta_y|)$$

Note that in the standard algorithm, the spectral radius is set to unity.

With the above approximations, the new smoothing factor, say in the ξ-direction, can be written as

$$\epsilon_{old} \frac{(1+M_\infty)a_\infty}{4} (|\xi_x| + |\xi_y|) \tag{30}$$

where ϵ_{old} is the old smoothing coefficient. Scaling the dissipation terms by the appropriate spectral radius is relatively inexpensive (the order of 5% increase in computational work).

For a three-dimensional test case of a rectangular wing with NACA0012 airfoil at $\alpha = 0°$ and $M_\infty = 0.72$, a steady solution previously obtained in 750 iterations is now obtained with this scaling in 400 iterations. This is a savings of over 40% in computer time. The new dissipation terms allow use of a larger time step without loss of accuracy, as seen in the C_p plot of Figure 16, and thus account for the improved efficiency.

(b) Special Variation of the Time Step

Keeping in mind that the grid variation accounts for significant metric variation, various researchers have long suggested that a variable Δt be used at each point in the computational field. The rationale is given that a large Δt should be used as the mesh cell increases in size, or alternatively, the equations should be scaled to reduce the condition number. In either event, Δt should vary as the spectral radius of $(A\delta_\xi + B\delta_\eta + C\delta_\zeta)$, which is an unknown magnitude.

In our two-dimensional experiments, we have scaled Δt based on simple estimates of the spectral radius of $(A\delta_\xi + B\delta_\eta)$, but we have

found that a geometric scaling inversly proportional to the Jacobian is adequate and far easier to program. The precise algorithm is now

$$[(I + \omega h\delta_\xi \hat{A} - \omega\varepsilon_I J^{-1}(\nabla_\xi \Delta_\xi)J)(I + \omega h\delta_\eta \hat{B} - \omega\varepsilon_I J^{-1}(\nabla_\eta \Delta_\eta)J](\hat{q}^{n+1} - \hat{q}^n)$$

$$= -\omega h(\delta_\xi \hat{E}^n + \delta_\eta \hat{F}^n) - \omega\varepsilon_E J^{-1}[(\nabla_\xi \Delta_\xi)^2 + (\nabla_\eta \Delta_\eta)^2] J \hat{q}^n \qquad (31)$$

where $\omega = \frac{1}{1 + \sqrt{J}}$ and $J = (\xi_x \eta_y - \xi_\eta \eta_x)$. The results shown in Figures 17 to 19 indicate the effectiveness of this scaling. For a circular cylinder at $M_\infty = 0.45$, wave drag can be computed to 4 significant figure accuracy within 100 iterations on a 50 × 20 grid using a variable scale factor ω for the time step Δt. Using the best constant Δt (i.e. $\omega = 1$), slightly over 250 iterations are needed as shown in Figures 17 and 18.

The circular cylinder calculation uses a stretched polar grid. To access the effect of grid distortion, similar runs were made for the NACA0012 airfoil at $\alpha = 0°$ and $M_\infty = 0.72$ and the grid shown in Figure 3. Residual decay using constant Δt ($\omega = 1$) and variation of Δt is indicated in Figure 19 and Figure 20 shows the corresponding C_p distribution. More iterations are required for this case (even though it is shock free), than the circular cylinder case, but the variable Δt proves to be at least twice as efficient.

The scaling given above in Equation (31) works effectively when the far field grid structure is very coarse, but a scaling such as $\omega = J^{-1}$ is unstable as ω becomes too large. Figure 21 shows the C_p distribution on a circular cylinder at $M_\infty = 0.45$ with a 78 × 26 grid structure computed using variable Δt scheme.

7. CONCLUDING REMARKS

The present study addressed the simulation of transonic three-dimensional wing-vortex interaction. The objective was to guide the design of an upcoming wind tunnel experiment and to support the interpretation and analysis of the data obtained.

A calculation procedure was adopted which employs an implicit finite-difference scheme for solving the Euler equations describing the flow about arbitrary geometric shapes. To check the accuracy and resolution of the procedure, calculations were carried out for three-dimensional wing-vortex interaction, for two-dimensional solutions of simple flow past lifting and non-lifting wing as well as for simple shear flows superimposed on the main stream. The key to the accuracy of the computations was found to be the grid structure.

The calculations were carried out within the memory of the Control Data Corporation 7600 computer. This machine tends to restrict the size of the grid which can be used. Consequently, a great deal of effort had to be exerted in clustering grid points to flow field gradient regions, especially near the vortex core. An elliptic grid generation routine with boundary and interior clustering functions was used in this effort and greatly aided in the generation of the finite-difference grid. Nevertheless, vortices with sharply contained swirl velocity could not be adequately resolved with the grid points available. The adequate resolution of such regions will likely require the development of more sophisticated software such as the overlay of one grid onto another. Alternatively, a more advanced computer than the CDC 7600 could be employed.

Initially computer time also proved to be quite excessive. Typically about 10 hours of CDC 7600 time was required to obtain one three-dimensional solution. By using a variety of techniques, the overall computational time was reduced by a factor of 5. As described in the main text, computer time was reduced by using diagonalization of the implicit operators and smoothing terms. Other methods including use of a variable time step were tested in two-dimensions but these have not been successfully implemented in three-dimensions. It is expected that with further work these changes could offer significant improved efficiencies in the current and in related computer codes.

The computations of the wing-vortex interaction considered a non-lifting rectangular wing of NACA0012 airfoil and an analytically represented spreading vortex in a free stream of Mach number 0.72. The solutions obtained showed that for the vortex considered here, the swirl velocity of the vortex imparted lift (rolling moment) and slight supercritical effect on the wing. Also, one of the calculations showed that when the imposed vortex is large (of the order of the span of the wing), a particularly important interaction between the vortex and the tunnel wall takes place. This is to be avoided in the design of the experiment. This result suggests that if one wants to avoid such an interaction and its effect on the flow on the wing, one has to consider a smaller vortex of the size of the chord of the wing in the experiment.

The results reported here are considered as initial in nature and are viewed as being qualitatively correct. Any quantitative improvement has to come by way of using a much refined grid.

8. REFERENCES

1. Beam, R. and Warming, R. F., An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law Form, Journal of Computational Physics, Vol. 22, Sept. 1976, pp. 87-110.

2. Beam, R. and Warming, R. F., An Implicit Factored Scheme for the Compressible Navier-Stokes Equations, AIAA Paper 77-645, 1977.

3. Warming, R. F. and Beam, R., On the Construction and Application of Implicit Factored Schemes for Conservation Laws, SIAM-AMS Proceedings, Vol. 11, Proceedings of the Symposium on Computational Fluid Mechanics, New York, 1977.

4. Steger, J. L., Implicit Finite-Difference Simulation of Flow About Arbitrary Geometries with Application to Airfoils, AIAA Paper 77-665, Presented at AIAA 10th Fluid and Plasma Dynamics Conference, Albuquerque, N. Mex., June 1977.

5. Kutler, P., Chakravarthy, S. R., and Lombard, C. K., Supersonic Flow Over Ablated Nosetips Using an Unsteady Implicit Numerical Procedure, AIAA Paper 78-213, 1978.

6. Pulliam, T. H. and Steger, J. L., On Implicit Finite-Difference Simulations of Three-Dimensional Flow, AIAA Paper 78-10, 1978.

7. Chaussee, D., On the Transonic Flow Field Surrounding Tangent Ogive Bodies with Emphasis on Nose Drag Calculations, AIAA Paper 78-212, 1978.

8. Nietubicz, C. J., Pulliam, T. H., and Steger, J. L., Numerical Solution of the Azimuthal-Invariant Thin-Layer Navier-Stokes Equations, AIAA Paper No. 79-10, 1979.

9. Steger, J. L. and Baily, H. E., Calculation of Transonic Aileron Buzz, AIAA Paper 79-134, 1979.

10. Chyu, W. J. and Davis, S. S., Calculation of Unsteady Transonic Flow over an Arbitrary Airfoil, AIAA Paper 79-1554, 1979.

11. Steger, J. L. and Kutler, P., Implicit Finite-Difference Procedures for the Computation of Vortex Wakes, AIAA Journal, Vol. 15, No. 4, April 1977, pp. 581-590.

12. Pulliam, T. H. and Chaussee, D. S., A Diagonal Form of an Implicit Approximation-Factorization Algorithm, Journal of Computational Physics, to be published, 1981.

13. Chaussee, D. S. and Pulliam, T. H., Two-Dimensional Inlet Simulation Using a Diagonal Implicit Algorithm, AIAA Journal, Vol. 19, No. 2, Feb. 1981, pp. 153-159.

14. Warming, R. F., Beam, R. M., and Hyett, B. J., Diagonalization and Simulataneous Symmetrization of the Gas-Dynamic Matrices, Math Comp., Vol. 29, 1975, p. 1037.

15. Thompson, J. F., Thames, F. C., and Mastin, C. W., Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies, Journal of Computational Physics, Vol. 15, No. 3, July 1974, pp. 299-319.

16. Steger, J. L. and Sorenson, R. L., Automatic Mesh-Point Clustering Near a Boundary in Grid Generation with Elliptic Partial Differential Equations, Journal of Computational Physics, Vol. 33, No. 3, Dec. 1979, pp. 405-410.

17. Sorenson, R. L., A Computer Program to Generate Two-Dimensional Grids About Airfoils and Other Shapes by the Use of Poisson's Equation, NASA TM 81198, May 1980.

18. Lock, R. C., Test Cases for Numerical Methods in Two-Dimensional Transonic Flows, AGARD-R-575-70.

19. Lamb, H., Hydrodynamics, 6th ed., Dover, New York, p. 592.

20. McMillan, O. J., Schwind, R. G., Nielsen, J. N., and Dillenius, M. F. E., Rolling Moments in a Trailing Vortex Flow Field, NASA CR-151961, Feb. 1977.

21. Steger, J. L. and Warming, R. F., Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite-Difference Methods, NASA TM-78605, July 1979.

22. Steger, J. L., A Preliminary Study of Relaxation Methods for the Inviscid Conservative Gas Dynamics Equations Using Flux Splitting, Flow Simulations FSI 80-4, Nov. 1980. Also to be published as a NASA Contractor Report.

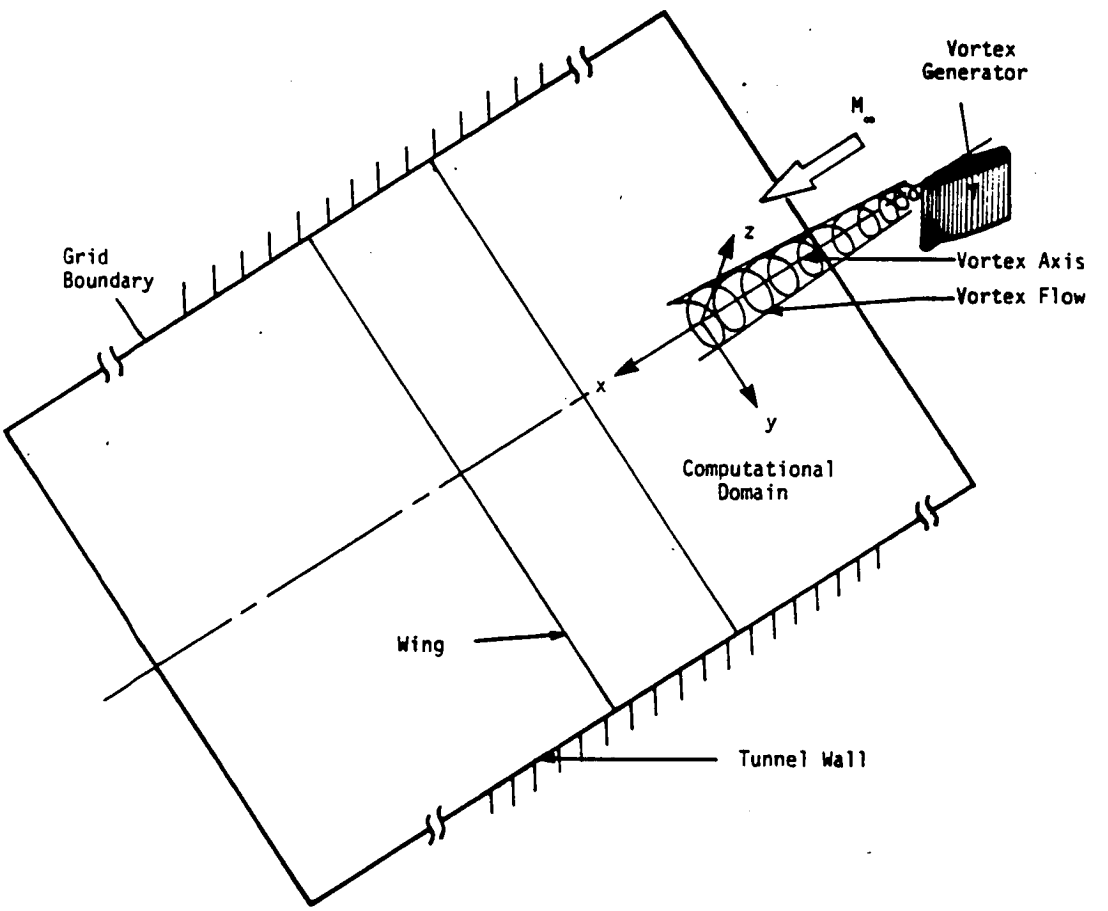
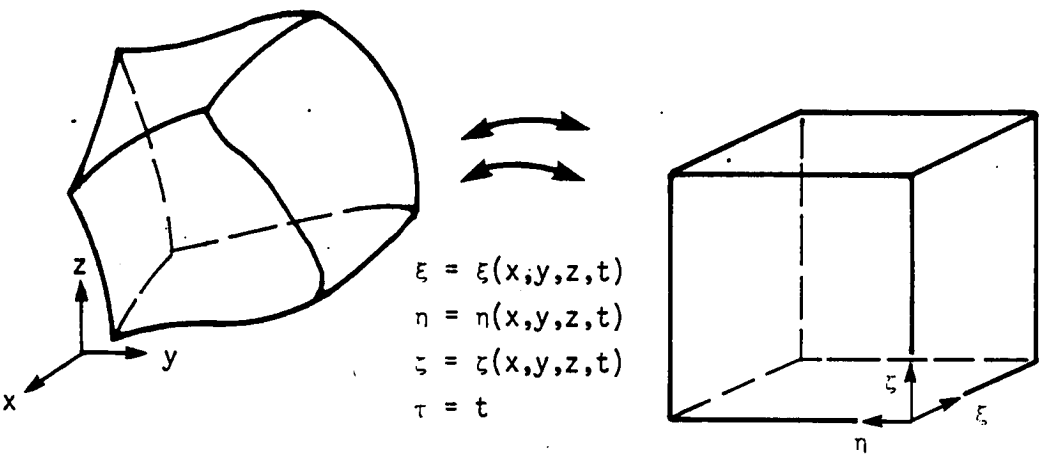
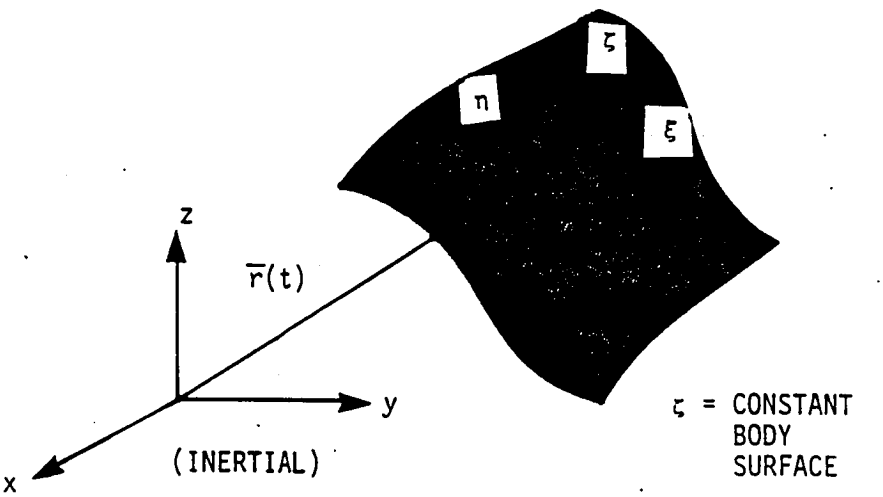


Fig. 1. Schematic of experimental arrangement showing inertial coordinate system.

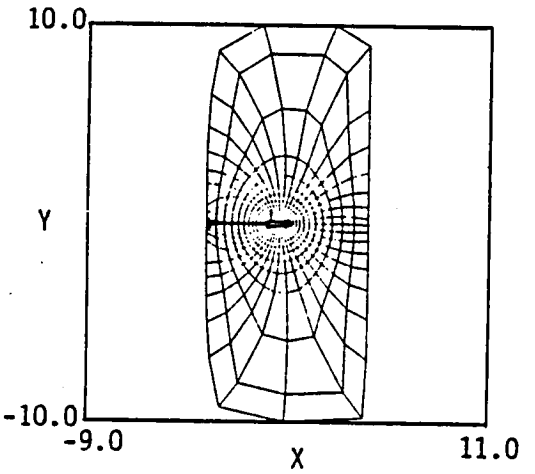


(a)

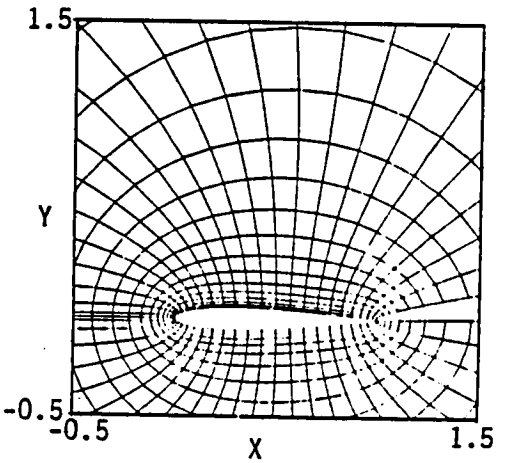


(b)

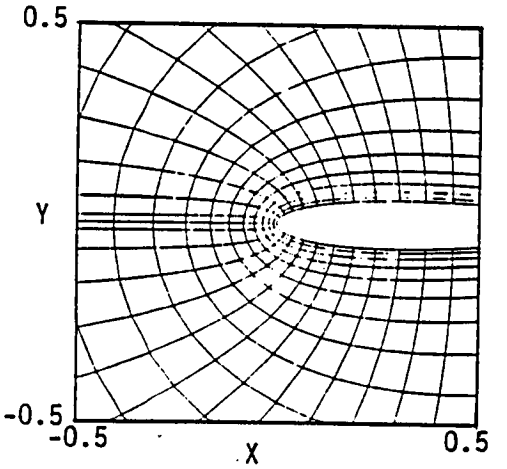
Fig. 2. Coordinates transformation and body coordinates.



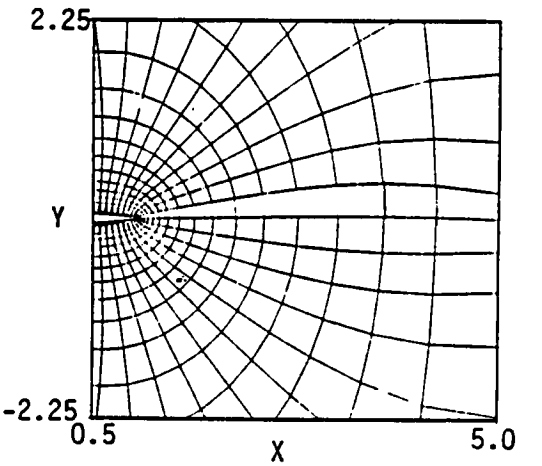
a) Full view of the grid



b) Details at the body surface



c) Details at leading edge



d) Details at trailing edge

Fig. 3. Two dimensional computational grid

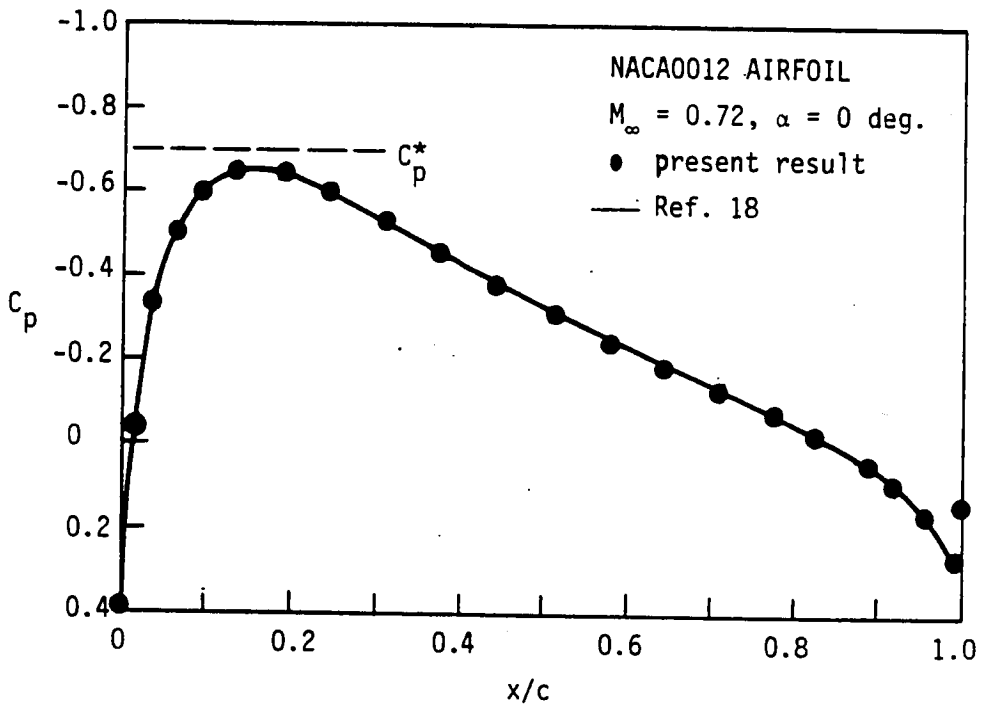


Fig. 4. C_p distribution in a two dimensional flow on a non-lifting airfoil.

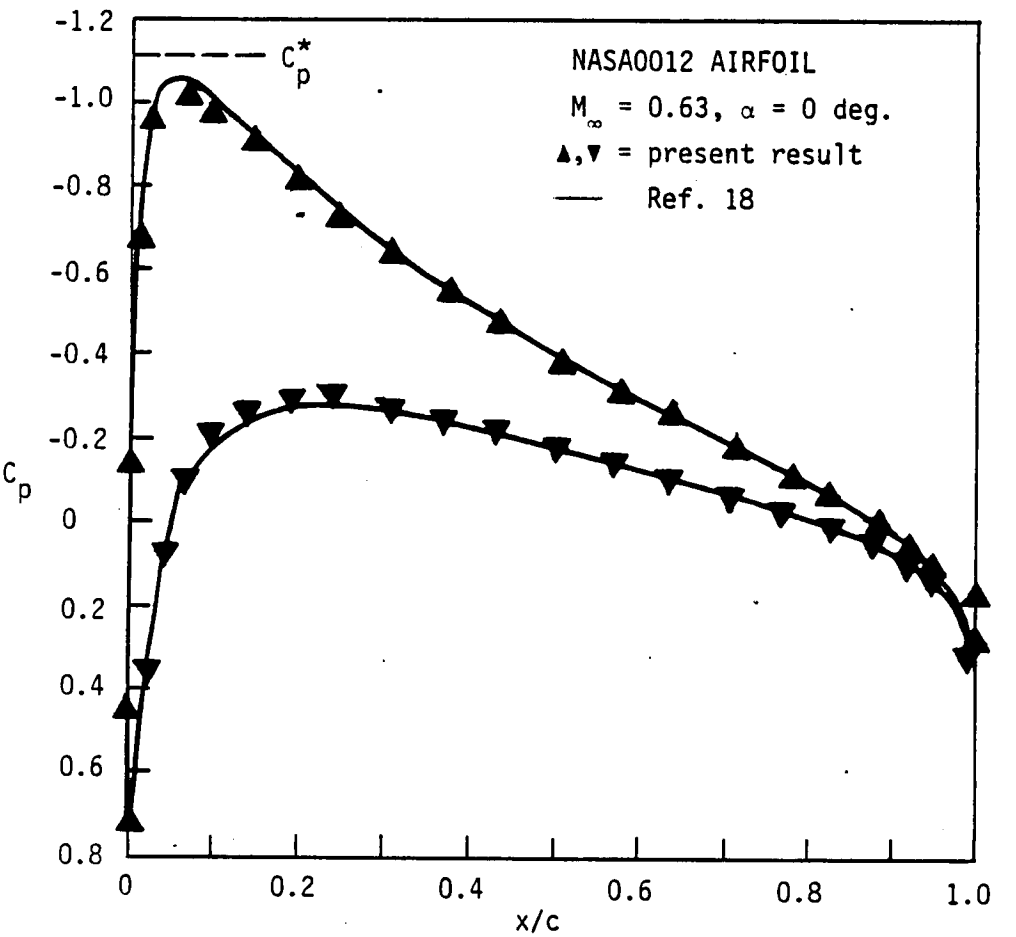


Fig. 5. C_p distribution in a two dimensional flow on a lifting airfoil.

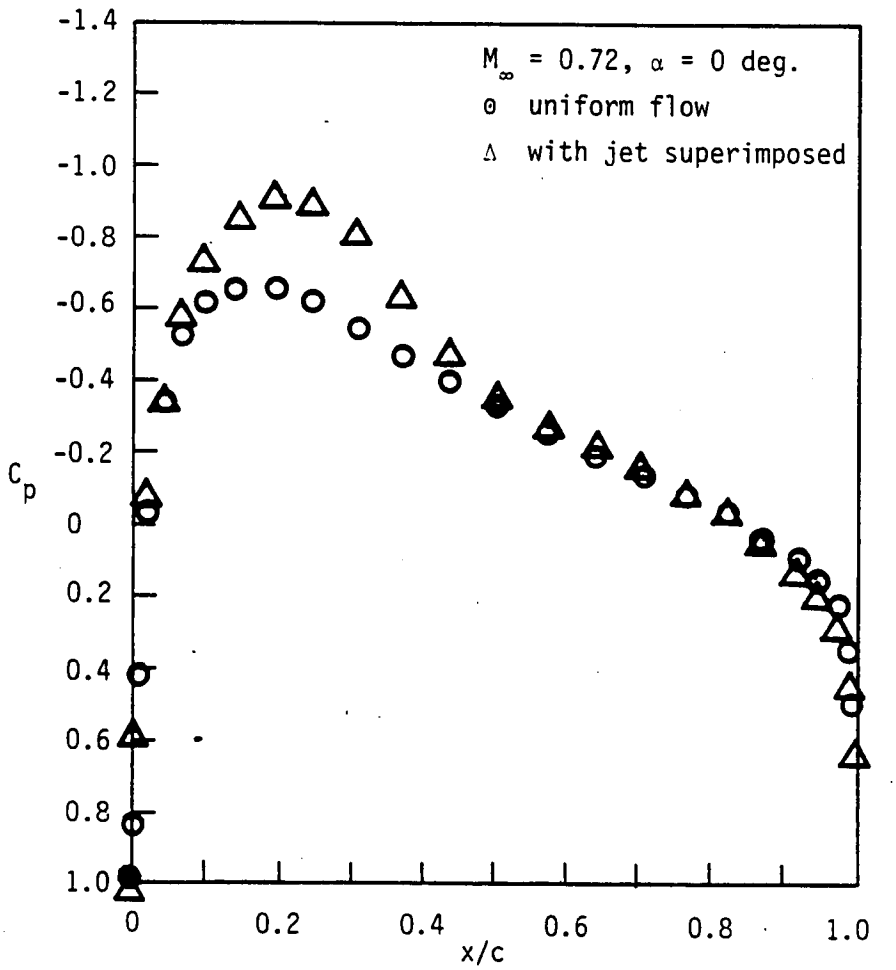


Fig. 6. Two dimensional jet flow with velocity distribution $u(z) = U_\infty\left(1 + 0.2e^{-(6z)^2}\right)$ superimposed on the free stream.

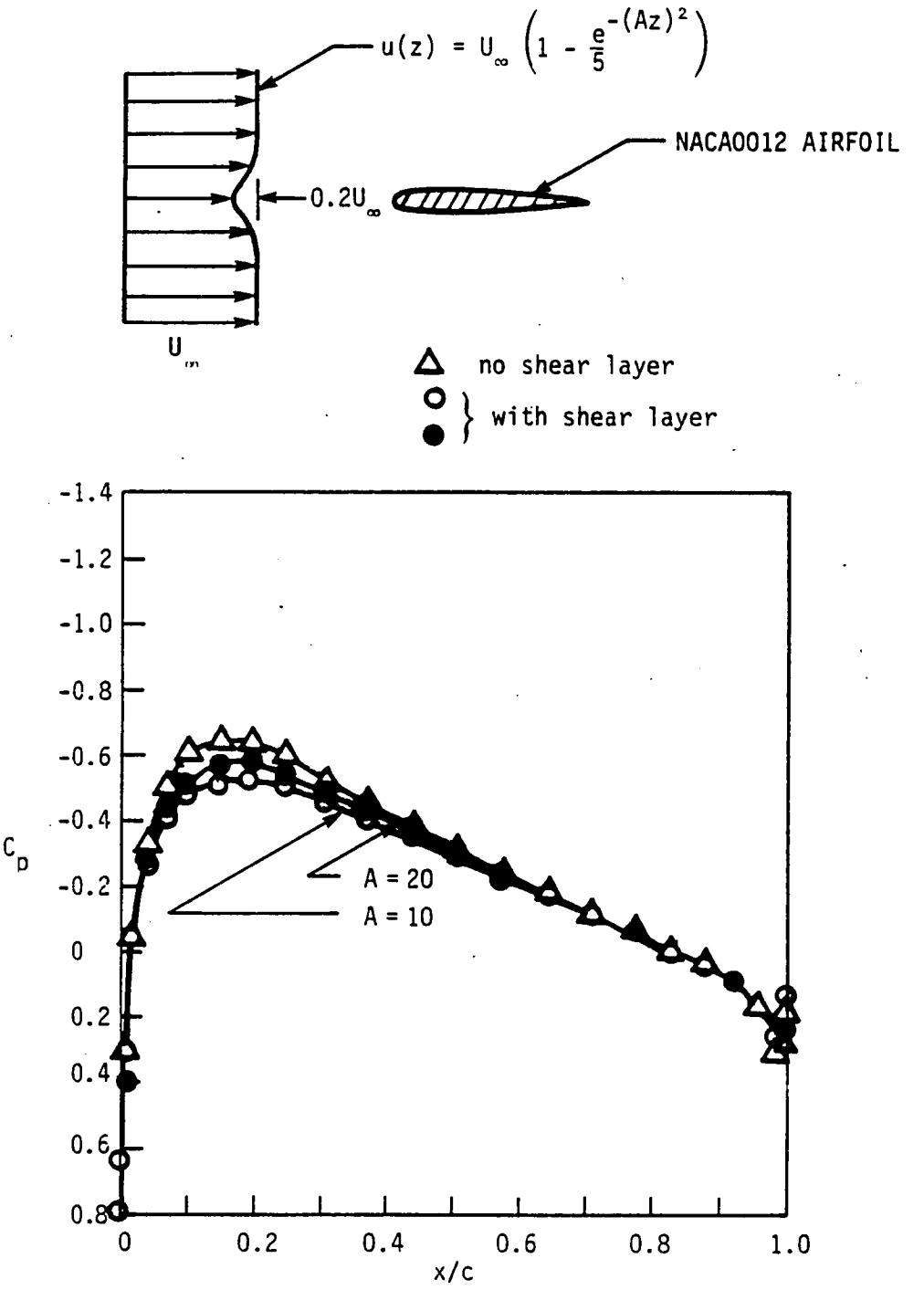


Fig. 7. Two dimensional wake flow superimposed on the free stream.

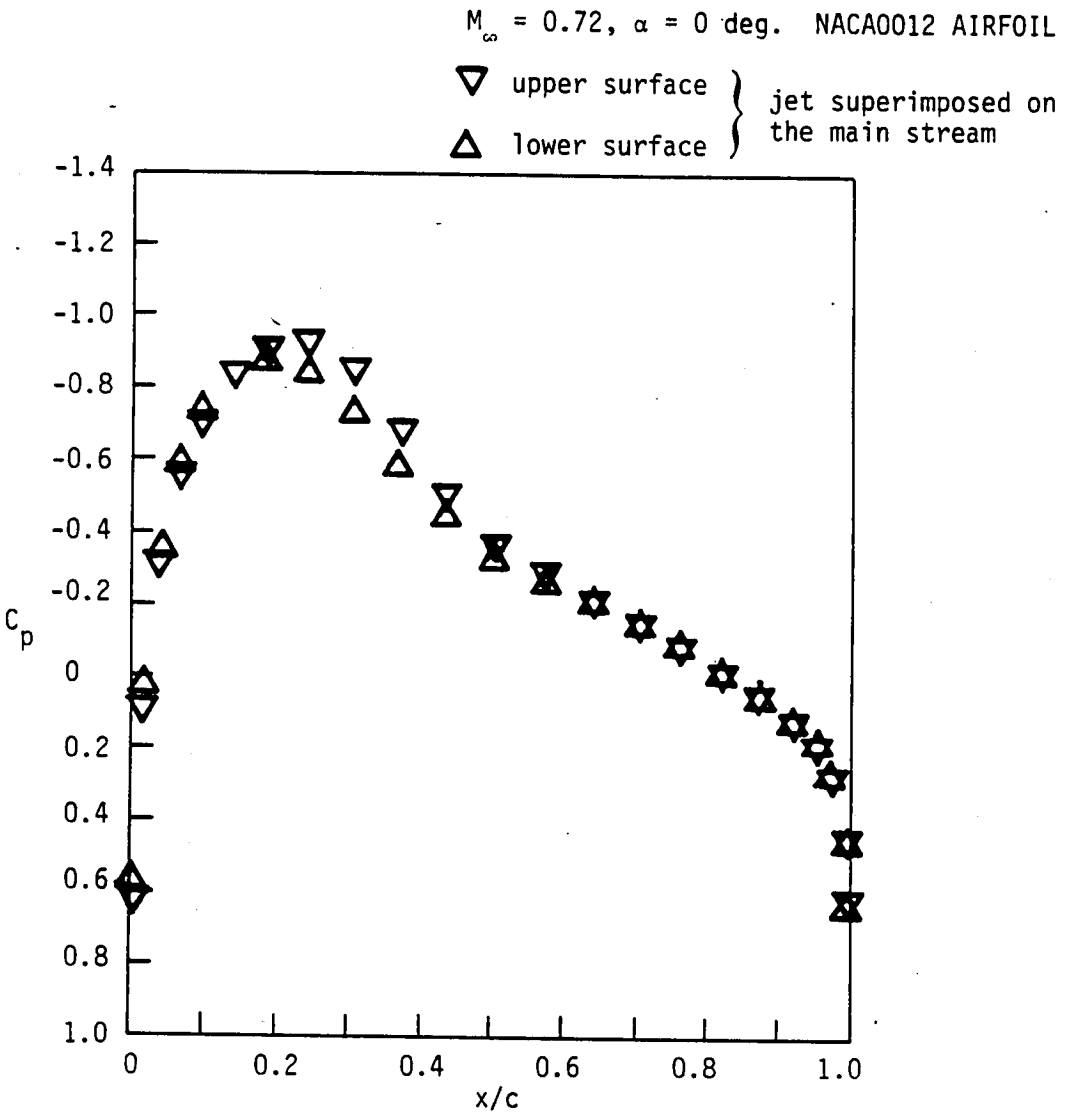


Fig. 8. Two dimensional asymmetric jet with velocity distribution $u(\hat{z}) = U_\infty\left(1 + 0.2e^{-(6\hat{z})^2}\right)$ with $\hat{z}$ = z-0.05 superimposed on free stream..

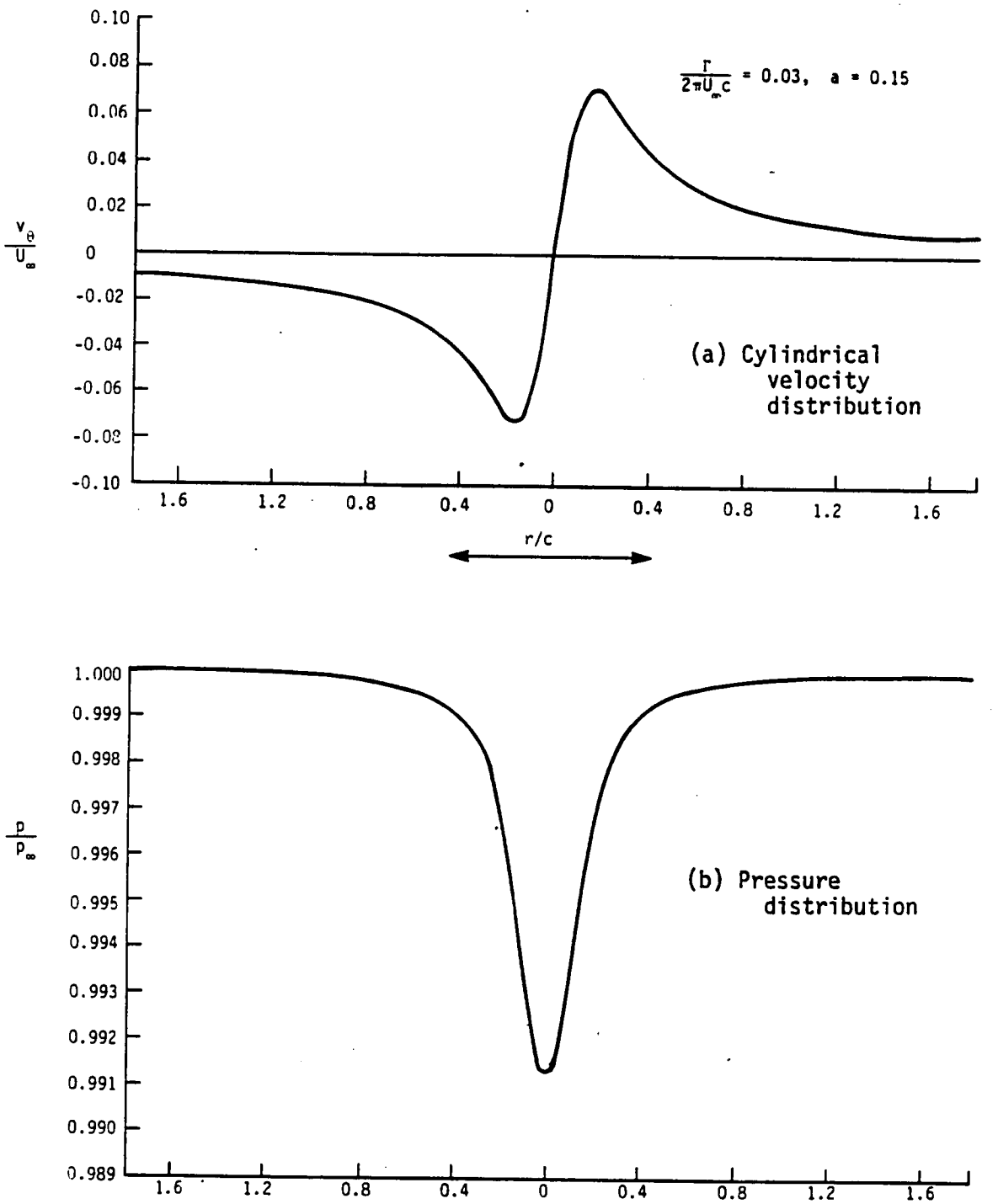
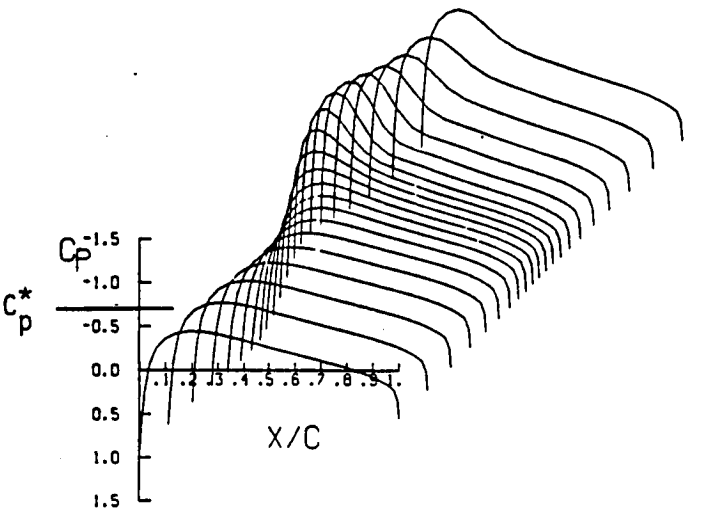
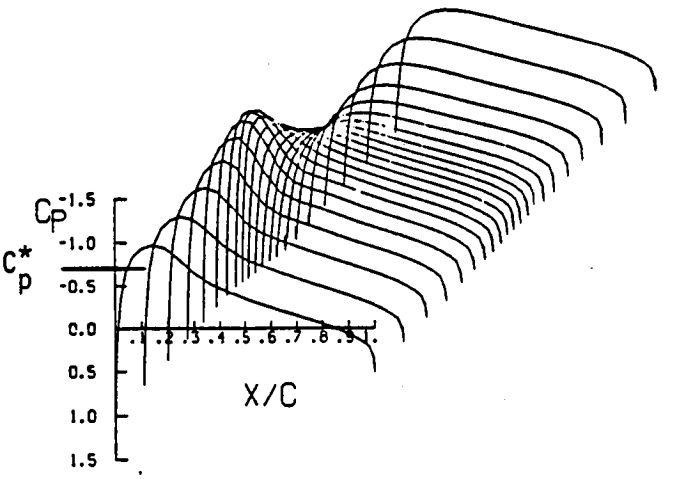


Fig. 9. Velocity and pressure distribution of a Lamb vortex



(a) upper surface



(b) lower surface

Fig. 10. C_p distribution at various spanwise stations on the wing.

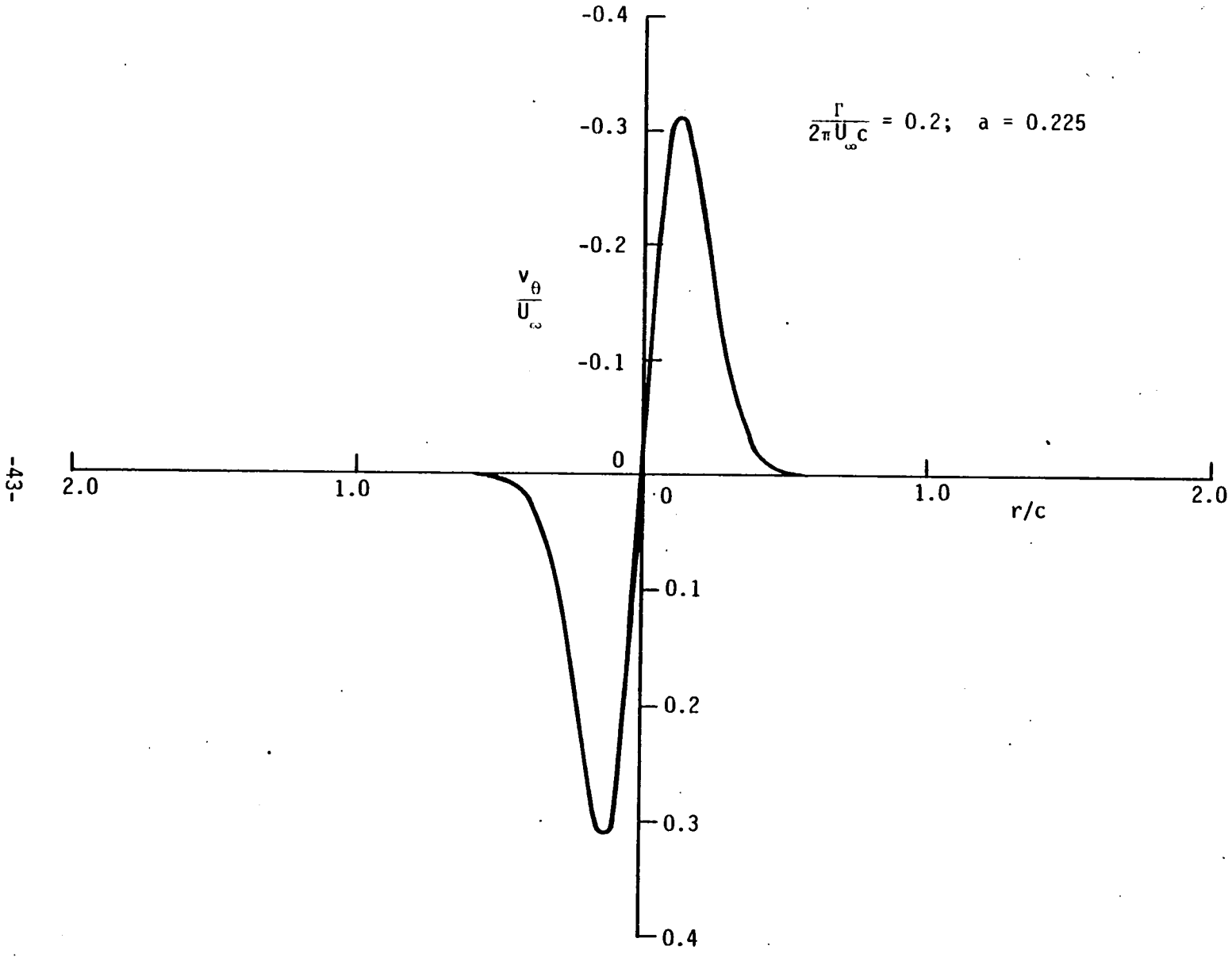


Fig. 11. Velocity distribution of a spreading vortex.

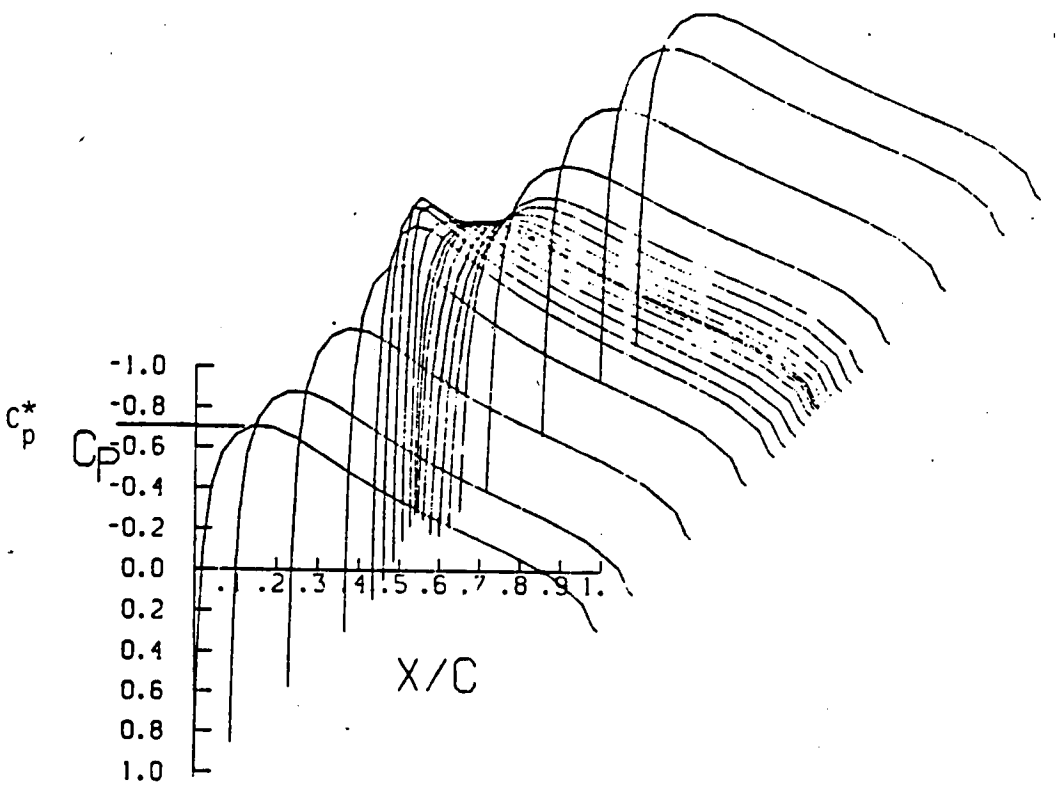


Fig. 12. C_p distribution on the wing.

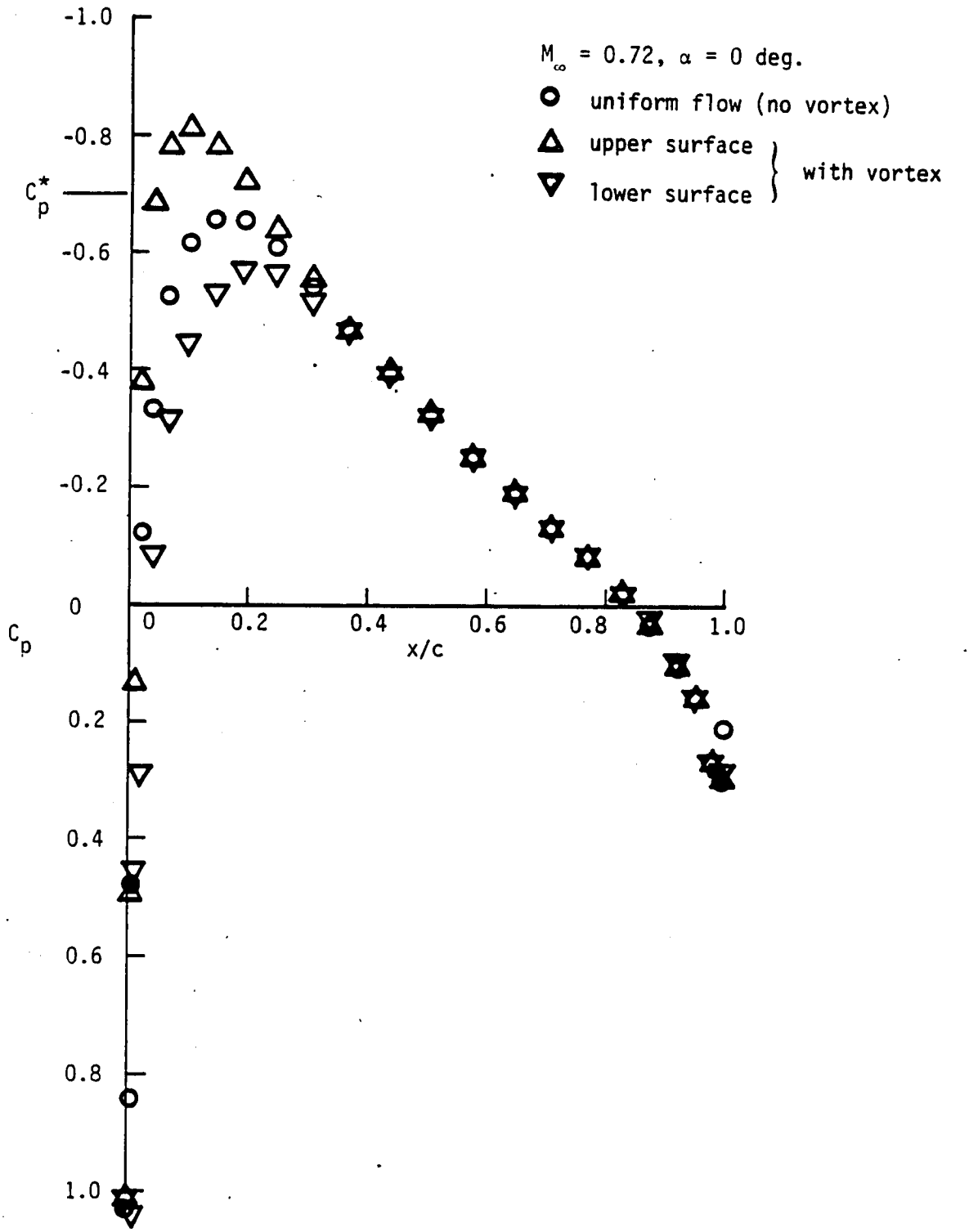


Fig. 13. C_p distribution at one spanwise station.

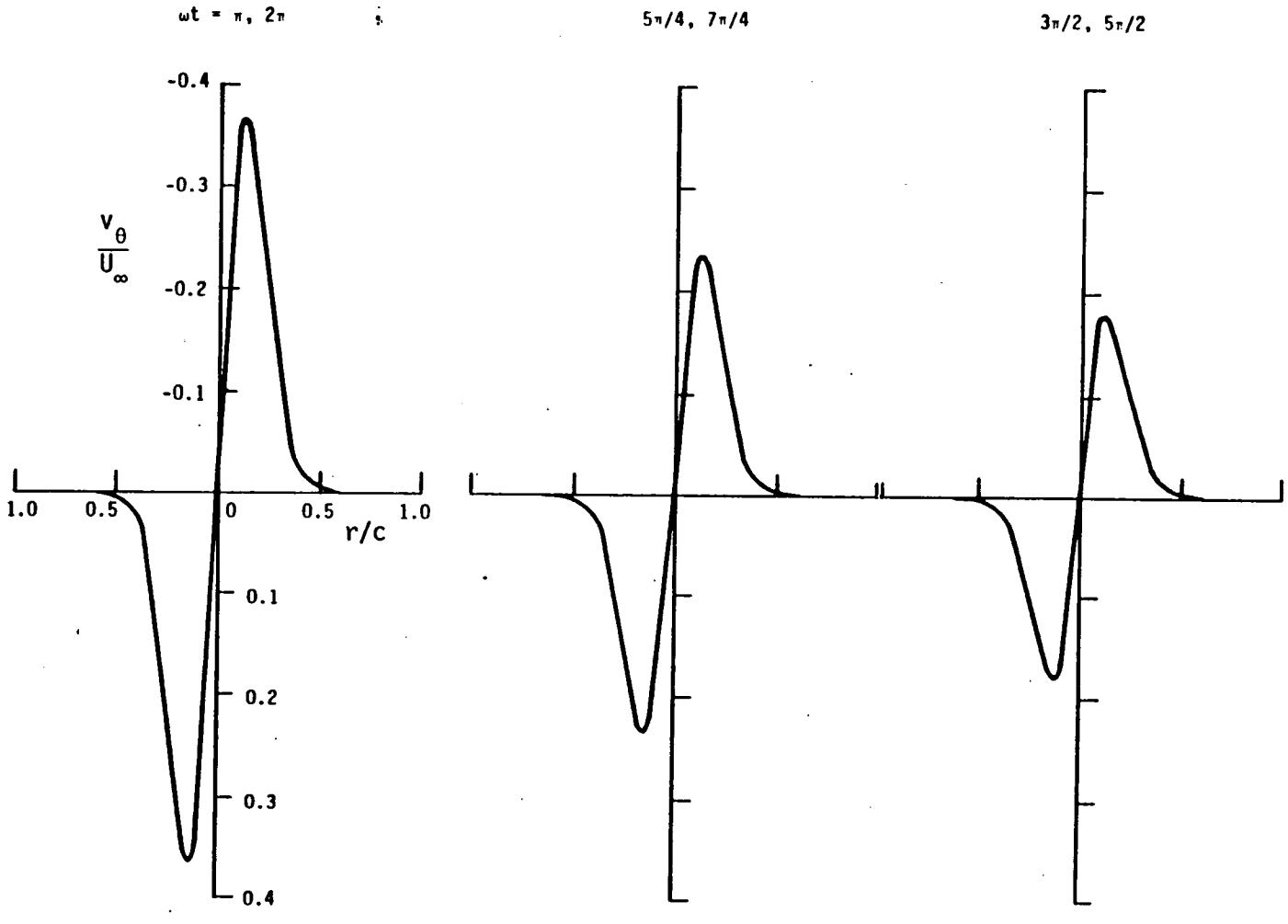


Fig. 14. Velocity distribution of an unsteady spreading vortex.

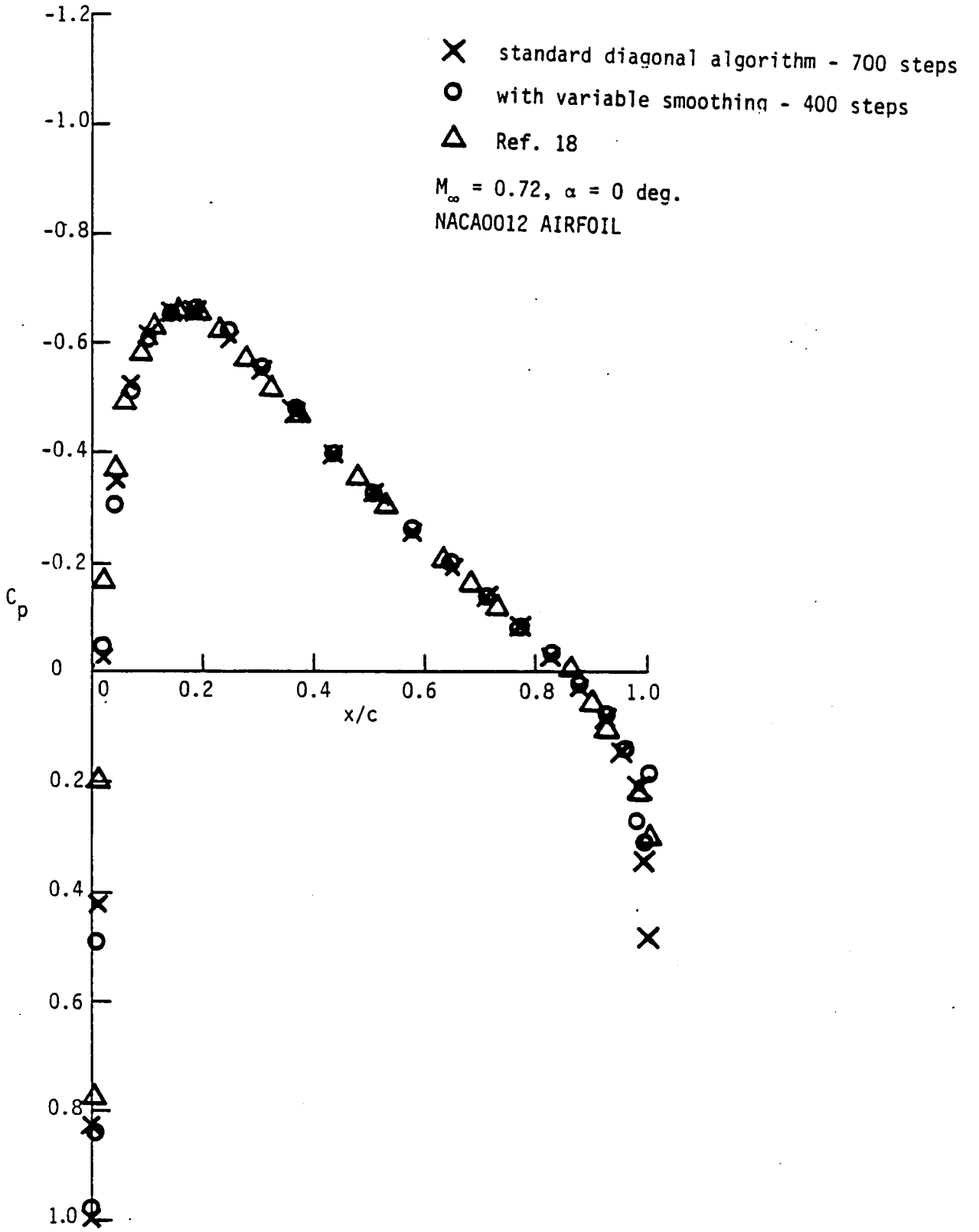


Fig. 16. Effect of variable smoothing.

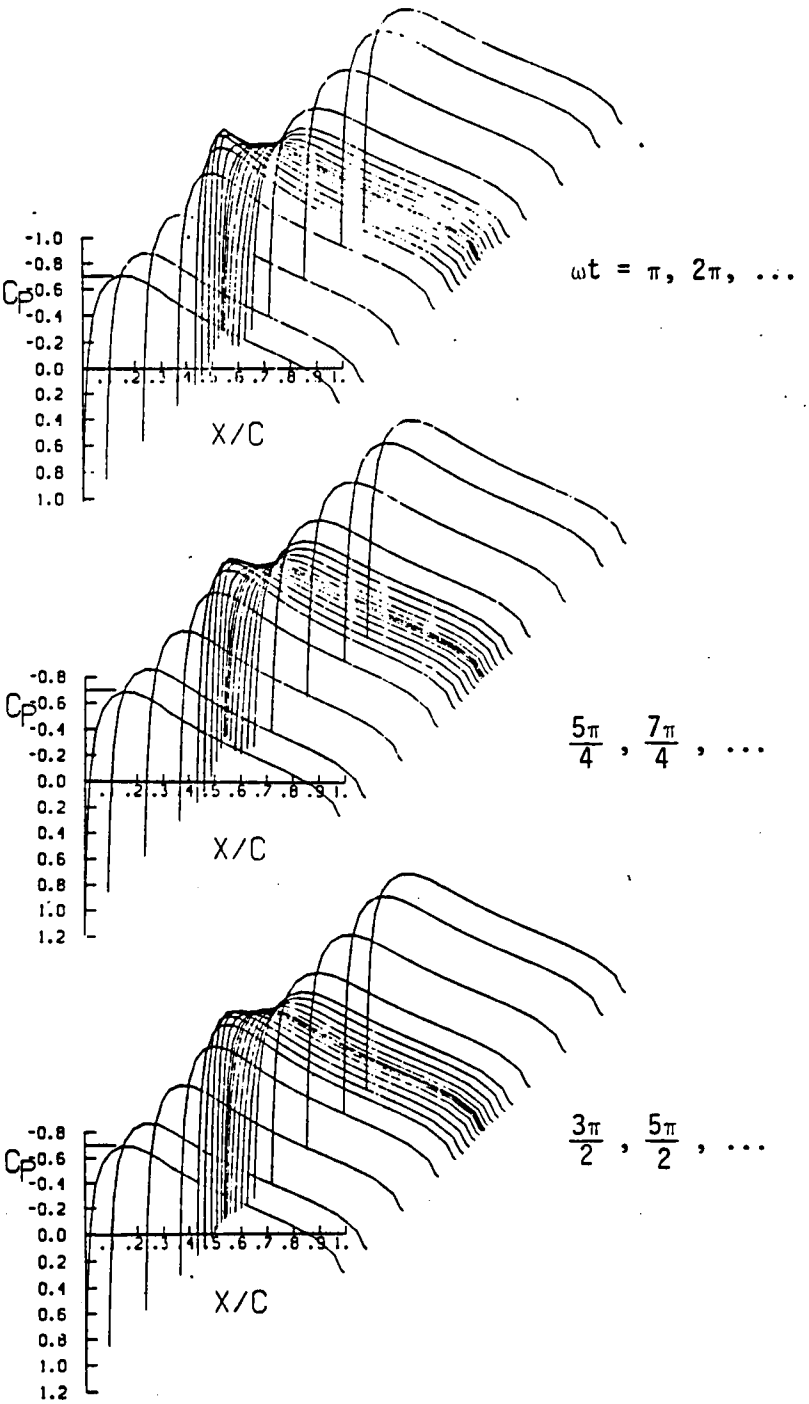


Fig. 15. C_p distribution of unsteady interaction.

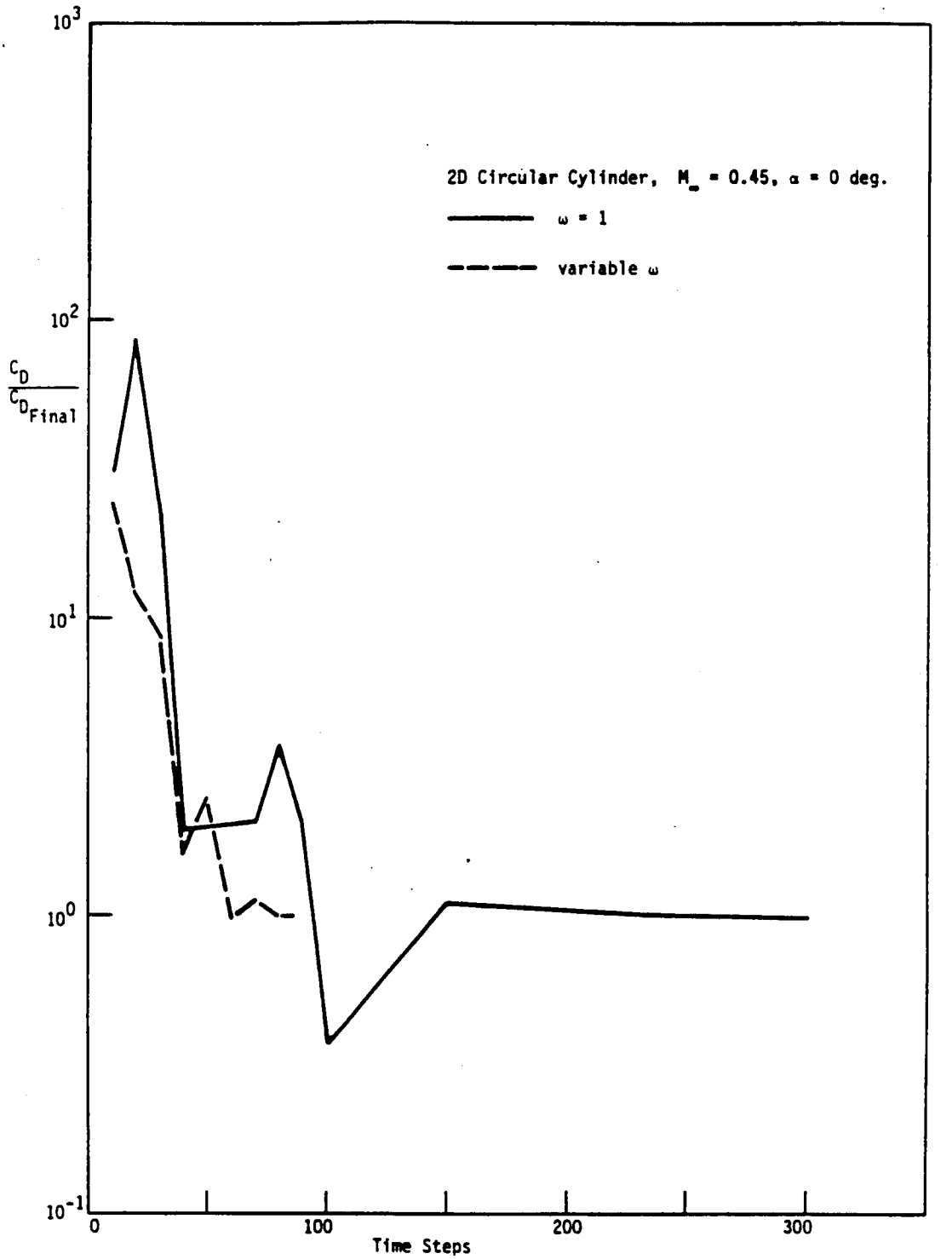


Fig. 17. Effect of scaling for the time step; wave drag of a two dimensional cylinder.

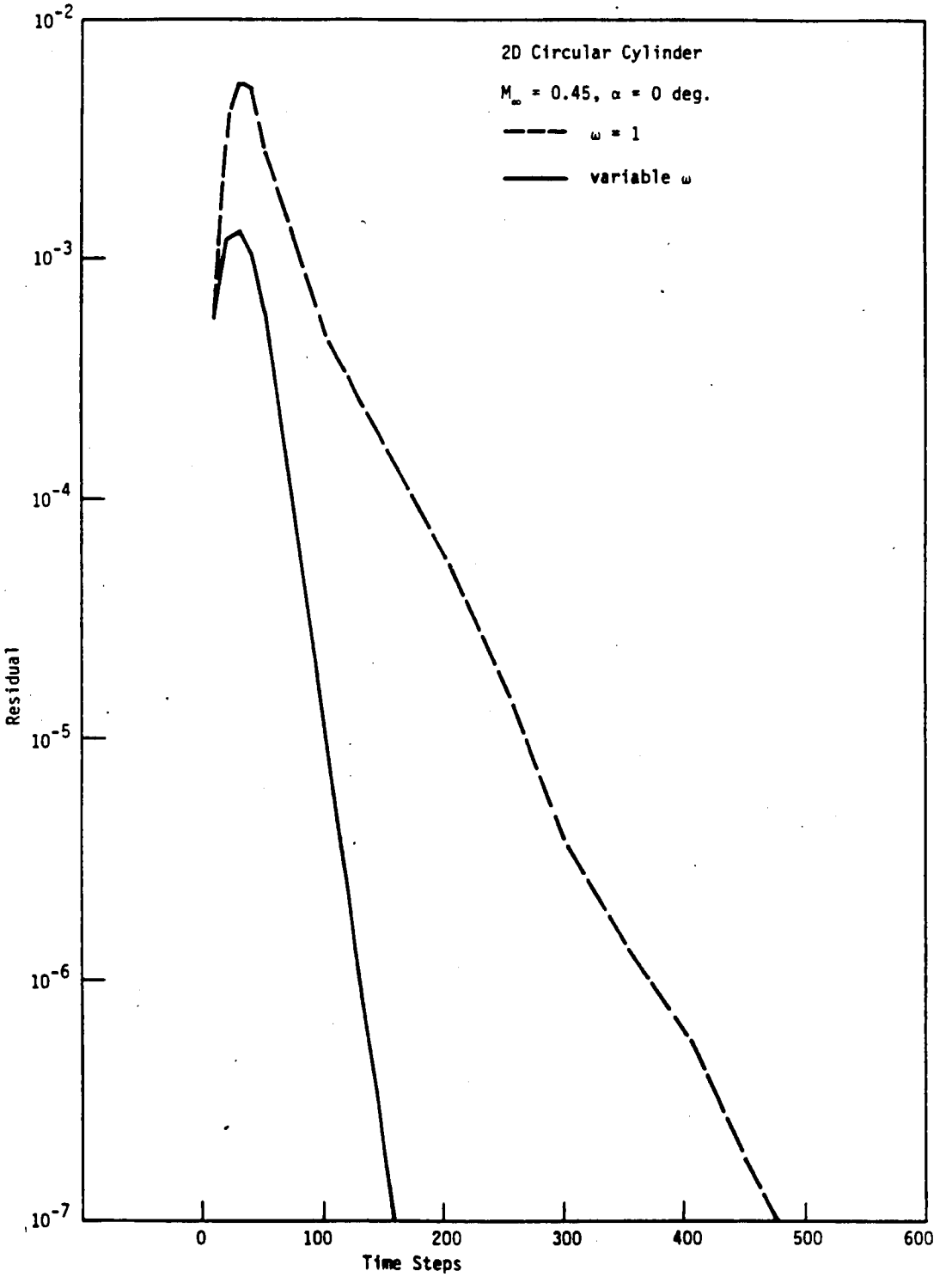


Fig. 18. Residual as a function of time step for two dimensional cylinder.

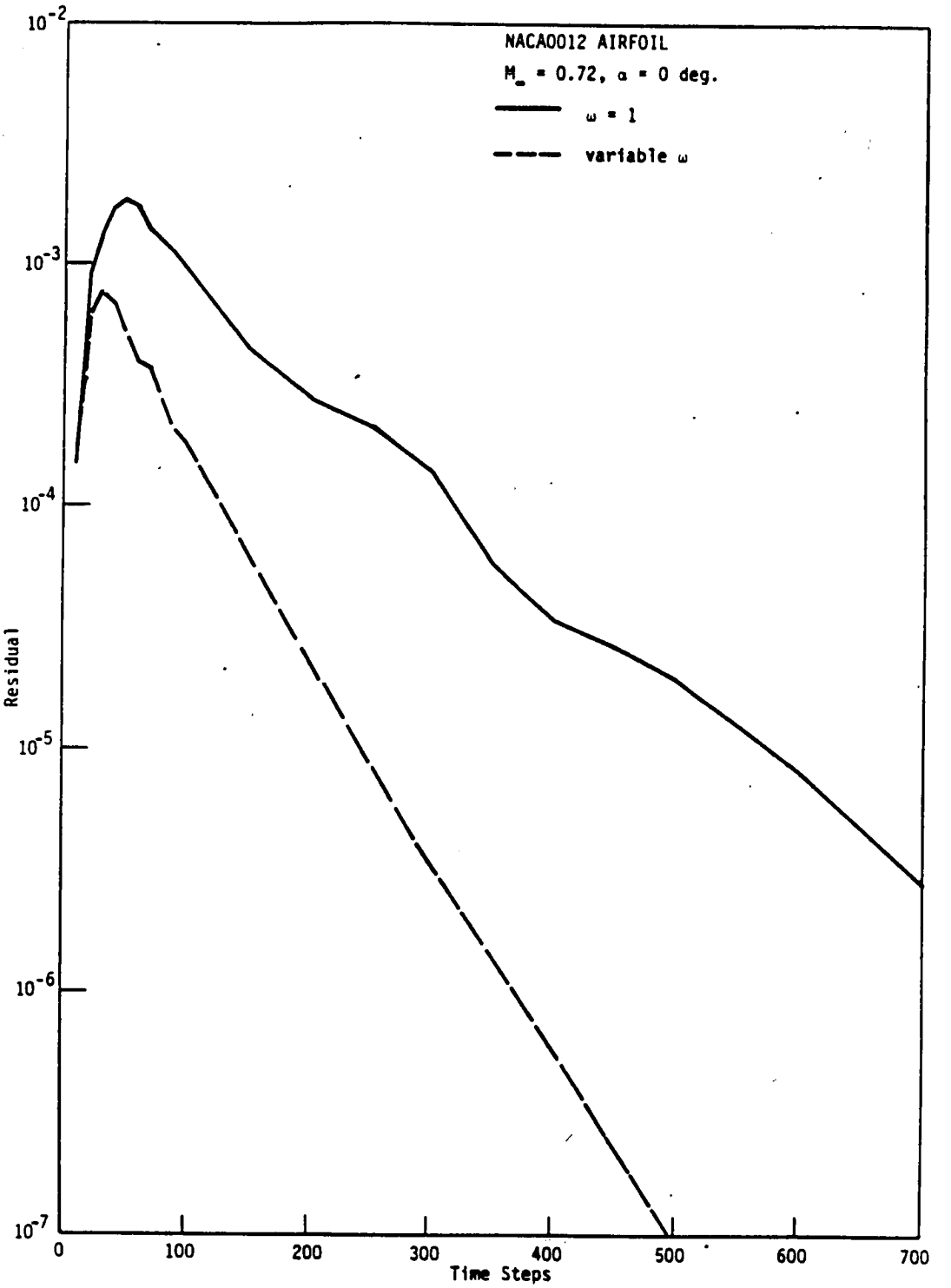


Fig. 19. Residual as a function of time step for two dimensional airfoil.

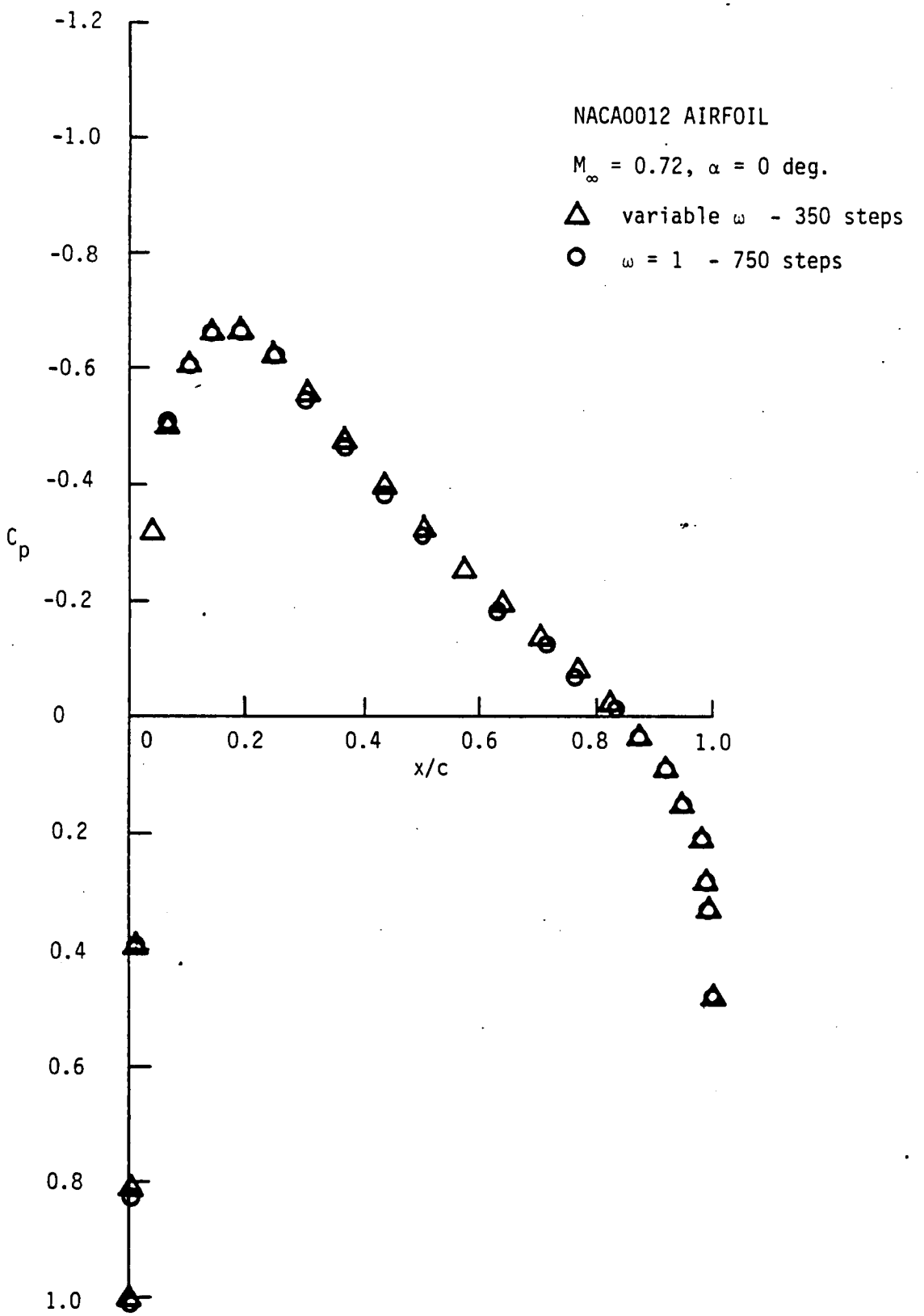


Fig. 20. C_p distribution with scaling for the time step.

2D Circular Cylinder

M_∞ = 0.45, α = 0 deg., 78 × 26 grid

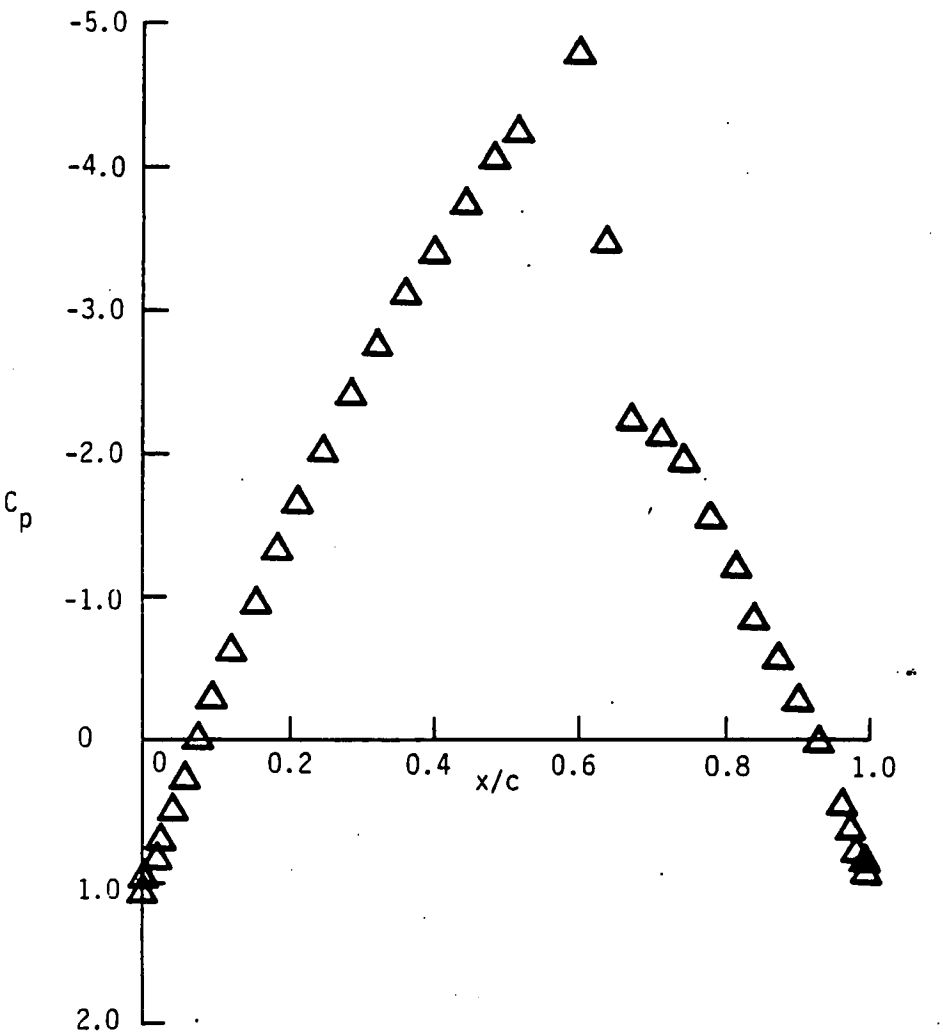


Fig. 21. C_p distribution with far field grid stretched and with scaled time step. Result shown is after 200 steps.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-166251		

4. Title and Subtitle	5. Report Date
Computation of Wing-Vortex Interaction in Transonic Flow Using Implicit Finite Difference Algorithm	March 1981
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
G. Srinivasan, and J. L. Steger	

9. Performing Organization Name and Address	10. Work Unit No.
Flow Simulations, Inc. 298 S. Sunnyvale Ave., Suite 204 Sunnyvale, CA 94086	T-3414Y
	11. Contract or Grant No.
	NAS2-10474

12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
National Aeronautics and Space Administration Washington, DC 20546	Contractor Final Report
	14. Sponsoring Agency Code
	505-42-21

15. Supplementary Notes

Technical Monitor: Wei J. Chyu, Mail Stop 227-8, NASA Ames Research Center, Moffett Field, CA 94035 (415) 965-6116 or FTS 448-6116

16. Abstract

An implicit delta form finite-difference algorithm for Euler equations in conservation law form has been used in preliminary calculations of three-dimensional wing-vortex interaction. Both steady and unsteady transonic flow wing-vortex interactions are computed. The computations themselves are meant to guide upcoming wind tunnel experiments of the same flow field. Various modifications to the numerical method that are intended to improve computational efficiency are also described and tested in both two- and three-dimensions. Combination of these methods can reduce the overall computational time by a factor of 4.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Wing-Vortex Interaction Transonic Flow Euler Equations	Unclassified - Unlimited Subject Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
UNCLASSIFIED	UNCLASSIFIED	61	

*For sale by the National Technical Information Service, Springfield, Virginia 22161

End of Document