DOE/ER/03077-170
March 1981

Courant Institute of

Mathematical Sciences

UNSTEADY TRANSONIC FLOW PAST AIRFOILS IN RIGID-BODY MOTION

I-Chung Chang

Research and Development Report
Prepared for
APPLIED MATHEMATICS RESEARCH PROGRAM
OFFICE OF BASIC ENERGY SCIENCES
U. S. DEPARTMENT OF ENERGY
UNDER CONTRACT DE-AC02-76ER03077; AND ONR

New York University

Courant Mathematics and Computing Laboratory

New York University

Mathematics and Computing DOE/ER/03077-170

UNSTEADY TRANSONIC FLOW PAST AIRFOILS IN RIGID-BODY MOTION

I-Chung Chang

March 1981

Prepared under Contract DE-AC02-76ER03077 with U. S. Department of Energy; and the Office of Naval Research.

Contents

		Page
	ABSTRACT	iii
I.	INTRODUCTION	1
	1. Flow Past Airfoils	2
	2. Engineering Considerations	4
	3. Mathematical Problem	7
	4. Plan of Work	9
II.	BASIC CONCEPTS	11
	1. Customary Flow Models	12
	2. Basic Numerical Concepts	25
III.	POTENTIAL FLOW EQUATION	37
	1. Characteristic Surface	38
	2. Computational Boundary Conditions	41
	3. Coordinate Transformation Technique	44
IV.	NUMERICAL METHOD	53
	1. Finite Difference Scheme	53
	2. Analysis for the Finite Differencing Strategy and the Approximate Factorization Process	60
V.	COMPUTATIONAL RESULTS	64
	1. Steady Calculations	64
	2. Unsteady Calculations	65
VI.	CONCLUSION	69
	1. Summary of the Work	69
	2. Extension of the Technique	70
	REFERENCES	71
	FIGURES	76
	APPENDIX	132
A.	A 5-DIAGONAL MATRIX SOLVER	132
B.	COMPUTER PROGRAM UFLO5	135
	1. Operation of the Program	135
	2. Glossary of the Program	139
	3. Listing of the Program	145

ABSTRACT

With the aim of developing a fast and accurate computer code for predicting the aerodynamic forces needed for a flutter analysis, we review some basic concepts in computational transonics. The unsteady transonic flow past airfoils in rigid body motion is adequately described by the potential flow equation as long as the boundary layer remains attached. The two dimensional unsteady transonic potential flow equation in quasilinear form with first order radiation boundary conditions is solved by an alternating direction implicit scheme in an airfoil attached sheared parabolic coordinate system. Numerical experiments show that the scheme is very stable and is able to resolve the highly nonlinear transonic effects for flutter analysis within the context of an inviscid theory.

I. INTRODUCTION

We begin with a survey of the behavior of flows past conventional airfoils. Then, we introduce the unsteady transonic problem in flutter analysis. Finally, we discuss the mathematical difficulties of solving such a problem.

1. Flow Past Airfoils

We begin our discussion with a brief survey of the behavior of flows past airfoils; when a conventional symmetric airfoil accelerates from subsonic speed to supersonic speed the flow pattern usually develops in the manner shown in Figure 11. As the flight speed of the airfoil reaches the critical speed, the local flow speed equals the local sound speed. Beyond the critical speed, a supersonic region appears on the airfoil which is usually terminated by a nearly normal shock through which the flow speed jumps from supersonic to subsonic. With a further increase in the flight speed, the shock moves aft and the size of the supersonic region and the shock strength both increase. If the pressure jump through the shock is sufficiently large, separation of the boundary layer occurs. This shock induced separation starts when the local Mach number, the ratio of local flow and sound speeds, just upstream of the shock is about 1.25 to 1.3. When the boundary layer downstream of the shock separates, the nature of the flow around the airfoil changes completely and often turbulent flow phenomena, such as buffet or buzz, start to occur.

The other important flight parameter is the angle of attack, the angle between the flight direction and the airfoil chord. The effect of changing the angle of attack of a conventional symmetric airfoil at a given supercritical speed is shown in Figure 9 . When the angle of attack is increased, the speed over the upper surface increases, and the shock strength and the supersonic region on the upper surface both increase.

The flow patterns for a modern supercritical airfoil acceleration in speed or angle are usually similar to the patterns shown in Figures 12 and 10, respectively. The supersonic zone in these cases may consist of several pieces.

2. Engineering Considerations

An aircraft under certain circumstances may experience vibrations of an unstable nature. This phenomenon, called flutter in aeroelasticity, is governed by the interaction of the elastic and inertial forces of structure with the aerodynamic forces generated by the motion of the vehicle. These forces interact in such a way that the vibrating structure extracts energy from the passing flow. This may lead to a progressive increase in the amplitude of vibration and may cause structural damage and loss of control of the vehicle.

For a given vehicle, the aerodynamic forces increase rapidly with the flight speed while the elastic and inertial forces remain essentially unchanged. There is a critical flight speed called the flutter speed, above which flutter occurs. The requirement that a flight vehicle be free of flutter over the entire flight range, which may include subsonic, transonic, supersonic and hypersonic speeds, is one of the most crucial factors in the design and construction of flight vehicles. The vibration characteristics of the vehicle at zero speed can be determined quite accurately by numerical methods or ground vibration tests [44]. Thus flutter analysis depends mainly on the knowledge of the aerodynamic forces. In subsonic and supersonic flight, aerodynamic forces can be predicted reasonably well by current methods based on linear theory. For transonic flight, nonlinear effects make the evaluation of the transient aerodynamic forces considerably more difficult. This has concerned the flutter analyst since the beginning of transonic flight. The transonic regime with its mixed subsonic-supersonic flow patterns, usually containing shock waves, is the most critical regime for the determination of the flutter boundary. A typical flutter boundary with transonic dip is depicted in Figure 1. The flight speed may exceed the flutter speed in the transonic region.

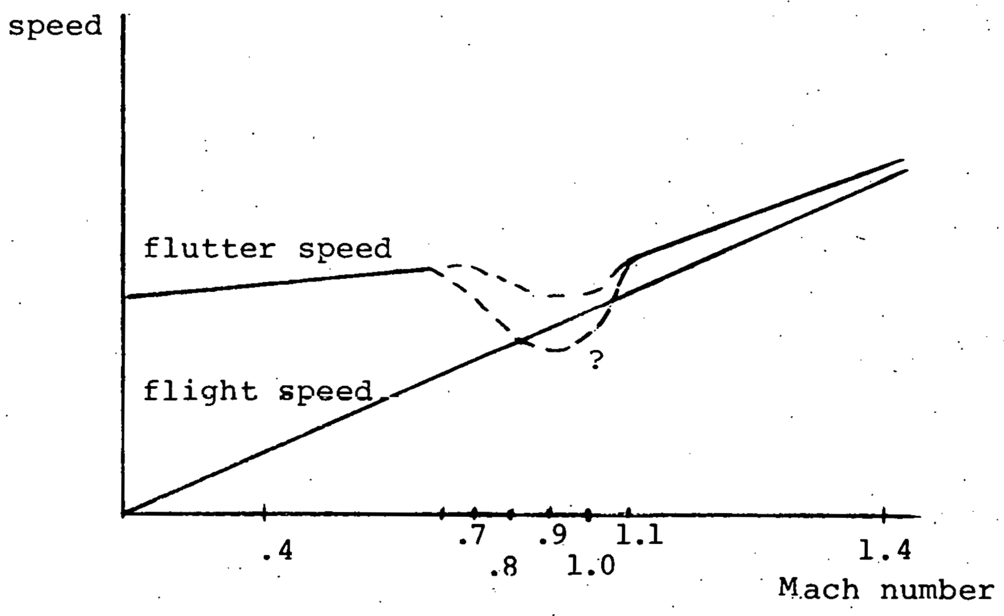


Fig. 1. Typical Flutter Speed vs. Mach Number Curves of a Flight Vehicle.

Currently, supercritical wings make it possible to cruise at transonic speeds with low drag. This leads to a renewed interest in transonic flutter analysis. In this paper we consider inviscid unsteady transonic potential flow past airfoils in rigid body motion with the aim of providing a method of predicting the aerodynamic forces needed for a flutter analysis.

3. Mathematical Problem

In mathematical terms we find solutions to a partial differential equation that describes flow outside a wing section which is in rigid body motion. There are several difficulties in this problem:

1. The equation is nonlinear,
2. The physical time direction is not the time-like direction of the equation when the flow is supersonic,
3. Shock waves occur, and
4. The body surface is moving in time, which is equivalent to saying that there is an essential singularity at infinity in the airfoil attached reference frame.

While much progress has been made in the mathematical theory of transonic flow, many basic questions remain open. For example, even for the small disturbance equation, one of the simplest nonlinear mathematical models, it has not been shown that the problem is well posed in a suitable class of weak solutions. The linear theory is deficient in predicting important features of transonic flow outside airfoils in low reduced frequency motion [29].

At present, a very effective way to study unsteady transonic flow is to obtain approximate solutions by computational methods. We overcome the first difficulty by the use of finite difference methods. This allows the solution to be advanced in time by solving a sequence of linear equations which approximate the nonlinear equation if the time step is small. The second difficulty, as well as the third, is solved by a type dependent differencing strategy which employs central differencing for all terms at subsonic points and upwind differencing for the streamwise derivatives and central differencing for the transversal derivatives at supersonic points. Shocks are captured automatically. The fourth difficulty is solved by using a coördinate system in which the airfoil is fixed. The far field then has an essential singularity that can in turn be treated by introducing radiation boundary conditions at the artificial boundaries which are a finite distance away from the body.

4. Plan of Work

The plan of this work is as follows: In Section II, several flow models derived from the conservation laws of fluid dynamics and the proper constitutional hypothesis are reviewed in decreasing order of complexity. We begin with the Navier-Stokes equations and step down to Euler equations, potential flow equations, small disturbance equation, and low frequency small disturbance equation. We discuss the proper boundary conditions and related concepts in each flow model. We also review some basic numerical concepts and discuss a splitting technique for constructing stable implicit schemes.

In Section III, we restrict our attention to the potential flow equation in quasilinear form. We study the characteristic surfaces of the equation and derive the proper radiation boundary conditions for the artificial boundaries of computational domain. We also discuss coordinate transformations which render the airfoil surface lying along a portion of coordinate surface in the computational domain.

In Section IV, we construct a highly stable alternating direction scheme for the potential flow equation in the computational domain. The finite differencing strategy and approximate factorization technique are analyzed through linear models, convection equation and wave equation. It is shown that the scheme is unconditionally stable for these two cases.

In Section V, we check the scheme by calculating steady flow past some airfoils. The computational results show that the scheme is very stable and there is no problem in calculating sonic flight. Then, we demonstrate our ability to calculate unsteady transonic flow past realistic airfoils in rigid body motion.

In Section VI, we present the conclusion of this work.

In Appendix A, we describe a 5-diagonal matrix solver employed in our scheme.

In Appendix B, we explain the operation of the computer program and the glossary of input parameters. We also present the listing of the computer program UFLO5.

ACKNOWLEDGEMENT

I would like to express my gratitude to Professor Antony Jameson for his advice, encouragement, patience and support. He suggested the problem and made the work possible. I would also like to thank Professors Jerome Berkowitz, Paul Garabedian, Gordon Johnson, Heinz-Otto Kreiss, Henry McKean, William Morris, and Olof Widlund for their support at various stages of my education; Drs. Frances Bauer, Pung-Nien Hu and Wei-Hai Jou, Geoffrey McFadden for their aid and discussions, and encouragement and suggestsions; Mr. John Marine for proof-reading; Ms. Connie Engle and Maie Croner for typing the manuscript; and my family for their support and understanding.

II. BASIC CONCEPTS

With the object of developing a fast and accurate computer code for unsteady transonic flow past airfoils we review in this section some basic mathematical models, including governing equations and boundary conditions for unsteady transonic flows and some relevant numerical concepts including the resolution of the finite difference mesh system, the ideas underlying the splitting technique and the shock capturing technique used to construct an alternating direction implicit scheme and the advantages and disadvantages of conservative and nonconservative difference schemes.

1. Customary Flow Models

In many aeronautical applications turbulent flow is observed. The phenomenon of turbulence is not well understood and currently much attention focusses on finding useful models to describe turbulent flow theoretically. Continuous flow models have been found adequate to describe a large class of flows of practical importance [32].

1.1 Navier-Stokes Equations

With the proper constitutive approximations, the conservation laws of mass, momentum and energy lead to the Navier-Stokes equations in Cartesian x,y coordinates in the conservation form [32,38]

$$(1) \qquad U_t + F_x + G_y = 0$$

where

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}, \quad F = \begin{pmatrix} \rho u \\ \rho u^2 + \sigma_x \\ \rho uv + \tau_{xy} \\ (e+\sigma_x)u + \tau_{yx}v - \kappa \frac{\partial \varepsilon}{\partial x} \end{pmatrix}$$

$$G = \begin{pmatrix} \rho v \\ \rho uv + \tau_{yx} \\ \rho v^2 + \sigma_y \\ (e+\sigma_y)v + \tau_{xy}u - \kappa \frac{\partial \varepsilon}{\partial y} \end{pmatrix}$$

with

$$\tau_{xy} = \tau_{yx} = -\mu \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)$$

$$\sigma_x = P - \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) - 2\mu \frac{\partial u}{\partial x}$$

and

$$\sigma_y = P - \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) - 2\mu \frac{\partial v}{\partial y}$$

in terms of density ρ, pressure P, velocity components u and v, viscosity coefficients λ and μ, total energy per unit mass e, specific internal energy ε and coefficient of heat conductivity κ. To close the system we adjoin the equation of state $p = p(\varepsilon,\rho)$. The simplest equation of state is the polytropic relation (γ-law)

$$P = (\gamma - 1)\varepsilon\rho \;, \qquad \gamma = \text{constant} \;,$$

where γ is the ratio of specific heats, equal to 1.4 for air.

The above system can be rewritten in the nondimensional form [42]

(2) $$U_t + F_x + G_y = R_e^{-1}(R_x + S_y)$$

where

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ e \end{pmatrix}, \quad F = \begin{pmatrix} \rho u \\ \rho u^2+p \\ \rho uv \\ u(e+p) \end{pmatrix}, \quad G = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2+p \\ v(e+p) \end{pmatrix},$$

$$R = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ R_4 \end{pmatrix}, \quad S = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ S_4 \end{pmatrix}$$

with

$$\tau_{xx} = (\lambda + 2\mu)u_x + \lambda v_y$$
$$\tau_{xy} = \mu(u_y + v_x)$$
$$\tau_{yy} = (\lambda + 2\mu)v_y + \lambda u_x$$
$$R_4 = u\tau_{xx} + v\tau_{xy} + \kappa P_r^{-1}(\gamma - 1)^{-1}\ \partial_x a^2$$
$$S_4 = u\tau_{xy} + v\tau_{yy} + \kappa P_r^{-1}(\gamma - 1)^{-1}\ \partial_y a^2$$

and

$$P = (\gamma-1)[e - 0.5\ \rho(u^2 + v^2)]$$

where the local sound speed a is given by

$$a^2 = \gamma(\gamma - 1)[\varepsilon - 0.5(u^2 + v^2)] \ ,$$

λ is taken as $-(2/3)\mu$, the Stokes hypothesis. Note that the nondimensional reference quantities are arbitrary, the Reynolds number R_e and the Prandtl number P_r used in equation (2) are defined in terms of these reference values.

Usually, two types of boundary conditions must be specified to determine flow past airfoils in motion.

a. The body surface condition requires the flow velocity relative to the body be zero (no slip condition), and

b. Appropriate far-field boundary conditions must be specified at the necessarily finite limits of the computational domain.

1.2 Euler Equations

If viscosity and heat conduction are neglected, the flow equations (2) are reduced to

(3) $$U_t + F_x + G_y = 0$$

and the equation of state for the γ-law gas,

(3') $$P = (\gamma - 1)\ \varepsilon\rho$$

In the inviscid flow field, if there are surfaces of discontinuity, the solution of the differential form (3) has to be interpreted as a weak solution of the flow equation with proper entropy condition. $u(x,y,t)$ is a weak solution of differential equation (3) if

$$\int_S^T \iint \left(W_t \cdot U + W_x \cdot F + W_y \cdot G\right) dx\ dy\ dt = \left.\iint W \cdot U\ dx\ dy\right]_S^T$$

for any smooth test function $W(x,y,t)$ which vanishes for $\|(x,y)\|$ large. An equivalent statement of weak solutions of the differential form (3) is that

a. The differential form (3) holds in the smooth region, and

b. Across any surface S of discontinuities, the following jump condition holds:

$$n_t[U] + n_x[F] + n_y[G] = 0 \quad \text{on} \quad S$$

Here $\tilde{n} = (n_x, n_y, n_t)$ is a unit normal vector to the surface S

of discontinuity pointing from the region (1) to the region (2). More specifically, if $\tilde{q}$ is the velocity vector of the flow and s is the velocity of the surface of discontinuity, then the jump relations derived from the conservation laws of mass, momentum and energy are

(4) $$m = (\tilde{n}\cdot\tilde{q}_1 - s)\rho_1 = (\tilde{n}\cdot\tilde{q}_2 - s)\rho_2$$

(5) $$m(\tilde{q}_2 - \tilde{q}_1) = \tilde{n}(p_1 - p_2)$$

and

(6) $$m(\frac{e_1}{\rho_1} - \frac{e_2}{\rho_2}) = p_1(\tilde{n}\cdot\tilde{q}_1) - p_2(\tilde{n}\cdot\tilde{q}_2)$$

Equation (5) implies the following two equations:

(7) $$m(\tilde{n}\cdot\tilde{q}_2 - \tilde{n}\cdot\tilde{q}_1) = p_1 - p_2$$

and

(8) $$m(\tilde{n}\times\tilde{q}_2 - \tilde{n}\times\tilde{q}_1) = 0$$

We can distinguish two cases with either $m \neq 0$ or $m = 0$ across the surface of discontinuity. In the first case, the tangential velocity component $\tilde{n}\times\tilde{q}$ is continuous across the surface which represents a shock wave; in the second case, it is a slip surface across which the pressure and the normal velocity component $\tilde{n}\cdot\tilde{q}$ are continuous while the density and the tangential velocity component can have arbitrary jumps. In the particular case both m vanishes and $\tilde{n}\times\tilde{q}$ is continuous, the slip surface is called a contact discontinuity where only density is discontinuous and there is no relative motion.

If there is a region of supersonic flow in the flow field it is well known [30] that shock waves will generally appear. The entropy condition to pick the right weak solution is that $\tilde{n}\cdot\tilde{q}$ decreases across the shock.

For the inviscid flow model the boundary condition at the body is reduced to the kinematic condition requiring the body to be impenetrable to the flow. Namely, the flow remains tangent to the body surface. In mathematical terms it is subject to the condition

$$\frac{dF}{dt} = F_t + \tilde{q}\cdot\nabla F = 0 \quad \text{on the body surface} \quad F(x,y,t) = 0 \ .$$

At the trailing edge it requires that the pressure and the flow direction be continuous. To be specific, the rate of change of circulation Γ, measured counterclockwise, is given by

$$\begin{aligned}
\frac{d\Gamma}{dt} &= \frac{d}{dt}\oint q \ dr \\
&= \oint \frac{dq}{dt}\cdot dr + \oint \frac{dq^2}{2} \\
&= -\oint \frac{dp}{\rho} + \oint \frac{dq^2}{2} \\
&= \oint \frac{dq^2}{2} \\
&= [q^2] \quad = -\frac{(q_u+q_\ell)}{2}\,(q_u - q_\ell)
\end{aligned}$$

where the velocities q_u and q_ℓ are the upper and lower velocities at the trailing edge. Consequently, if the circulation is to change with time, neither the average velocity nor the jump velocity can be zero. The vortex sheet, comprised of vortex filaments, trailing downstream of the airfoil, is viewed as a slip surface. In reality, the vortex sheet is convected with the motion of the fluid and rolls up on itself due to its self-induced velocities. A consistent model accounting for the roll-up of the sheet would add greatly to the difficulty of constructing a boundary conforming coordinate system. If the convection and roll-up of the sheet are ignored, the vortex sheet may be assumed to be along the streamwise coordinate surface that leaves the airfoil trailing edge smoothly. The constraints applied on it are that the pressure and the normal velocity component be continuous across the vortex sheet.

The appropriate radiation boundary conditions at the artificial boundaries of computational domain are again needed.

We remark that in steady flow calculation, the energy equation can be replaced by Bernoulli's equation for constant total enthalphy $H = \frac{\gamma}{\gamma-1}\frac{P}{\rho} + \frac{u^2+v^2}{2} \equiv \text{const.}$ thereby reducing the number of dependent variables from four (ρ,u,v,e) to three (ρ,u,v).

1.3 Potential Flow Equation

Assuming that the flow can be described by a velocity potential, the Euler equations can be reduced to a single quasilinear equation. This implies that the flow is irrotational and hence in view of Crocco's relation that there are no entropy changes in the flow. The entropy produced by a shock is proportional to the third order of the shock strength [12]. We may assume that the entropy is conserved across the shock if we just consider weak shocks, such as occur on the surface of a well designed airfoil. This approximate model should not be a source of serious error if the Mach number of the normal component of the flow ahead of the shock is less than 1.2.

Let Φ be the velocity potential with $q = \nabla\Phi$ the velocity vector. The equation of motion $Dq/Dt = -\nabla p/\rho$ leads to

$$\frac{\partial q}{\partial t} + \nabla\left(\frac{q^2}{2}\right) - q\times(\nabla\times q) + \frac{\nabla P}{\rho} = 0$$

$$\nabla\left(\Phi_t + \frac{q^2}{2} + \int\frac{dp}{\rho}\right) = 0$$

$$\Phi_t + \frac{q^2}{2} + \int\frac{dp}{\rho} = f(t) + \text{constant.}$$

If $\phi = \Phi - \int f(t)\,dt$ then $\nabla\phi = \nabla\Phi$ and $\phi_t = \Phi_t - f(t)$. We therefore call ϕ velocity potential as well and we have the Bernoulli equation for ϕ

$$\phi_t + \frac{q^2}{2} + \int\frac{dp}{\rho} = \text{constant.} \tag{1}$$

The conservation of mass is

(2) $$\rho_t + (\rho\phi_x)_x + (\rho\phi_y)_y = 0$$

We take the equation of state to be

(3) $$p = (\tfrac{1}{\gamma})\ \rho^\gamma$$

with $\rho_\infty = 1$ and $a_\infty = 1$.

In the smooth region of the flow we may eliminate ρ from the above equation and get a quasilinear equation for ϕ. The equation of continuity yields

$$-\left(\frac{1}{\rho}\frac{D\rho}{Dt}\right) = \nabla\cdot q = \nabla\cdot\nabla\phi = \Delta\phi$$

The Bernoulli equation, after differentiation, leads to

$$\frac{D}{Dt}\left(\phi_t + \frac{q^2}{2}\right) = -\frac{D}{Dt}\left(\int \frac{dp}{\rho}\right) = -\frac{D}{Dt}\left(\frac{\rho^{\gamma-1}}{\gamma-1}\right) = -\rho^{\gamma-2}\frac{D\rho}{Dt}$$

$$= -\frac{a^2}{\rho}\frac{D\rho}{Dt} = a^2\Delta\phi$$

Finally, we combine them and get an equation

(4) $$\frac{D}{Dt}\left(\partial_t + \frac{q}{2}\cdot\nabla\right)\phi = a^2\Delta\phi$$

or

(5) $$\phi_{tt} + 2u\phi_{xt} + 2v\phi_{yt} = (a^2 - u^2)\phi_{xx} - 2uv\phi_{xy} + (a^2 - v^2)\phi_{yy}$$

The shock conditions which will be applied in the model (1), (2) and (3$'$) are

a. $\tilde{n}\times\tilde{q}$ is continuous across the shock which implies ϕ is continuous

b. $(\tilde{n}\cdot\tilde{q} - s)\cdot\rho$ is continuous which says mass is conserved across the shock, where s is the shock speed

c. $\tilde{n}\cdot\tilde{q}$ decreases across the shock. This is the entropy condition.

Here $\tilde{n}$ is the normal to the shock surface.

According to these conditions, a normal shock is to be modeled as a jump between equal points of an isentropic stream tube. The corresponding change in normal momentum is balanced by a force on the discontinuity. The combined force on the body and the discontinuity is zero so that the integral of the pressure over the body surface yields a drag which is an approximation to the wave drag.

The surface condition requires that

$$\nabla\phi\cdot\tilde{n} = v_B\cdot\tilde{n} \quad \text{on the body surface.}$$

Here $\tilde{n}$ is the normal to the body surface and v_B is the body velocity relative to the absolute reference frame. The trailing edge and wake condition are basically the same as for the Euler equations. Specifically, the rate of change of circulation Γ of airfoil is given by

$$\frac{d\Gamma}{dt} = \frac{d}{dt}\oint q\ dr = \frac{d}{dt}\,[\phi]_{TE} = [\phi_t]_{TE} \tag{6}$$

With the continuity of pressure and normal velocity component across the wake, the Bernoulli equation gives

$$[\phi_t] + [\frac{\phi_s^2}{2}] = [\phi_t] + \bar{\phi}_s[\phi_s] = 0 \tag{7}$$

where $\bar{\phi}_s$ stands for the average velocity at any point in the wake. Thus the circulation can change only if there is a velocity jump at the trailing edge. Hence a vortex sheet is shed and the wake condition (7) expresses the equation for the transport of vorticity downstream. We will discuss radiation boundary conditions for equation (5) in a later section.

1.4 Small Disturbance Equation

For small disturbance transonic flows, the flow equation can be further simplified by a perturbation method [1]. Namely, assume that the thickness to chord ratio τ of the airfoil under consideration is small in the sense of $\tau^{2/3} \sim 1 - M_\infty^2 << 1$, where M is the free stream Mach number. If we expand the potential ϕ to the potential flow equation in the powers of τ and retain the lowest approximation, we obtain the small disturbance equation

$$S_1\phi_{tt} + 2S_2\phi_{xt} = v_c\phi_{xx} + \phi_{yy}$$

where

$$S_1 = M_\infty^2(\kappa^2/\tau^{2/3}), \quad S_2 = M_\infty^2(\kappa/\tau^{2/3})$$

and

$$v_c = (1-M_\infty^2)/\tau^{2/3} - (\gamma+1)M_\infty^2\phi_x - (\gamma-1)M_\infty^2\kappa\phi_t$$

The reduced frequency $\kappa = \omega c/q_\infty$ is a measure of the degree of unsteadiness of the flow field since it is the ratio of the time scale of the airfoil flight speed c/q_∞

and that of the unsteady motion $1/\omega$, where c is the chord of airfoil, ω is the frequency of the unsteady motion and q is the flight speed. The flow velocity is the sum of the free stream velocity q_∞ and the gradient of ϕ. We remark that ϕ, t, y and x have been scaled by $c\tau^{2/3}q_\infty$, $1/\omega$, $c/\tau^{1/3}$ and c respectively.

The primary merit of this approximation is that the surface condition is very simple. The surface of the airfoil is transferred to the slit $y = 0,\ 0 < x < 1$, which is the mean surface approximation to the airfoil in the new scaled coordinate system. If $h(x,t)$ is the unsteady displacement of the airfoil surface from the true mean contour $f(x)$, then the surface condition is

$$\phi_y = f_x + h_x + h_t \quad \text{on the slit} \quad y = 0 , \quad 0 < x < 1.$$

The wake condition is that the jump of the pressure coefficient across the wake $y = 0 ,\ 1 < x$, must vanish Namely, $[c_p] = 0$ where $c_p = -\,2\,\tau^{2/3}(\phi_x + \phi_t)$.

1.5 Low Frequency Small Disturbance Equation

For low frequency $\kappa \sim 1-M^2 \sim \tau^{2/3} << 1$, it is well known [1] that the small disturbance equation reduces to

$$2S_2\phi_{xt} = v_c\phi_{xx} + \phi_{yy}$$

where $v_c = (1-M_\infty^2)/\tau^{2/3} - (\gamma+1)M_\infty^2\phi_x$.

The surface boundary condition and the wake condition can be either that of the small disturbance equation or as follows:

a. $\phi_y = f_x + h_x$ on $y = 0$, $0 < x < 1$

b. $[c_p] = 0$ on $y = 0$, $1 < x$ where $c_p = -2\tau^{2/3}\phi_x$.

1.6 When to Use Which Model

Each model has its own limitations based on the assumptions used in developing the flow equations. For example, the low frequency small disturbance equation does not describe high frequency motion well, the small disturbance equation does not describe the blunt leading edge airfoils well, the potential flow equation does not describe the strong shock wave well, the Euler equation does not describe separated flow well. We briefly remark that when strong shocks lead to separation, viscous effects cannot be neglected. Either boundary layer correction equations or the Navier-Stokes equation have to be employed [10]. The consideration of turbulence is probably needed to resolve the complicated flow phenomena such as buffet separation, reattachement, and so on. Here we will consider flows with relatively weak shocks which can be adequately described by the potential flow model.

2. Basic Numerical Concepts

The numerical problem is to find an approximate solution accurate to within some tolerance. The most basic and widely used method to solve time dependent problems is the finite difference method. In this section we review some basic numerical concepts about the finite difference method and propose a finite difference strategy with a splitting technique which result in unconditionally stable schemes for the heat equation, linear advection equation , and wave equation, respectively. And we will apply those ideas to construct an ADI type scheme for the potential flow equation in quasilinear form in Section IV.

2.1 Mesh Spacing

In the finite difference method, one performs all calculations on the grid points of a computational domain which is of finite extent. Once the grid points are given, the resolution of the physical phenomena is naturally limited by the mesh spacing. To be specific, we introduce some terms through the definition [36] of a Fourier mode:

$$\begin{aligned} y &= a\ e^{i(\omega t+\xi x)} \\ &= a\ e^{i\xi(x+(\omega/\xi)t)} \\ &= a\ e^{i\frac{2\pi}{\lambda}(x+ct)} \end{aligned} \tag{1}$$

where a is called the amplitude; ω, the phase rate; ξ, the wave number, $c = \omega/\xi$, the wave speed; $\lambda = 2\pi/\xi$, the wave length; $\omega t + \xi x$, the phase angle; $f = \omega/2\pi$, the frequency; $\tau = 1/f$, the period.

Suppose we express a function u(x) as a Fourier series

$$u(x) = \sum_{-\infty}^{\infty} a_j\ e^{i\xi_j x} \tag{2}$$

On a mesh system containing I equally space points of spacing Δx, the Fourier mode of shortest wave length resolvable in the system is $\lambda_{min} = 2\Delta x$; the longest wave length is $\lambda_{max} = (I-1)\Delta x = L$. The corresponding wave numbers are $\xi_{max} = \pi/\Delta x$ and $\xi_{min} = 2\pi/L$.
So the total number of wave models resolved by this mesh system is $N = (I-1)/2$ and the part of u which can be

resolved by this system is the partial sum

$$\hat{u} = \sum_{-N}^{N} a_j \, e^{i\xi_j x} \quad \text{with} \quad \xi_j = \frac{j\pi}{N\Delta x}$$

In viscous flow, the diffusion and the advection for a Fourier mode $u = a\, e^{i\xi x}$ lead to

$$\mu \frac{d^2u}{dx^2} = -\mu\, \xi^2 u$$

and

$$\rho u \frac{du}{dx} = \rho u (i\xi) u$$

Their ratio is $(\rho u)/(\mu\xi)$. As ξ increases the diffusion becomes stronger and dominates eventually. The mesh spacing should be fine enough to understand the dissipation mechanism. On the other hand, for computational efficiency the number of mesh points must be kept to the minimum required to resolve all the significant phenomena. Hence, in practice [32], a typical computational domain consists of a fine mesh region where viscous effects are important and a coarse mesh region where the flow is essentially inviscid. Some techniques, for instance, coordinate stretching and/or coordinate transformations are useful [14,19,24,41]. Automatic mesh system generation techniques for flow about multiple bodies in a plane have been developed [42,43].

2.2 Time Step and Approximation Factorization Technique

Explicit finite difference methods have demonstrated their ability to solve a wide range of flow problems. However the size of a time step that a solution can be advanced during each step of calculation is restricted by the Courant-Friedrichs-Lewy condition (CFL condition). The CFL condition imposed on the time step is

$$\Delta t \leq \frac{\Delta}{|q|}$$

where Δ is the grid mesh spacing and $|q|$ is the fastest propagation speed anywhere on the mesh system. Therefore, the solution requires long and expensive computation time.

Unlike the explicit method, implicit methods can be theoretically stable for all time step sizes. Unfortunately, an implicit method in two or higher space dimensions requires a set of equations to be solved at the advanced time level which is not always easy to accomplish directly. Accordingly, the splitting technique is introduced to yield feasible computational processes. We illustrate the splitting technique on the heat equation in two space dimensions.

(1) $$\phi_t = \phi_{xx} + \phi_{yy}$$

The finite differencing strategy is replacing the differential operator in time D_t by the forward difference, the ϕ in the right-hand side by the average of ϕ^{n+1} and ϕ^n

and the differential operators D_{xx} and D_{yy} by the second order center difference operators in x and y respectively. Namely,

$$(2)\quad \frac{\phi_{ij}^{n+1}-\phi_{ij}^{n}}{\Delta t} = \left\{\left(\frac{E_x-2I+E_x^{-1}}{\Delta x^2}\right) + \left(\frac{E_y-2I+E_y^{-1}}{\Delta y^2}\right)\right\}\left(\frac{\phi_{ij}^{n+1}+\phi_{ij}^{n}}{2}\right)$$

where $E_x\phi_{ij}^n = \phi_{i+1,j}^n$, similarly $E_y\phi_y^n = \phi_{i,j+1}^n$.
The accuracy of the finite difference equations is of second order in time and space.

We may write the finite difference equation in terms of $\delta_{xx} = E_x-2I+E_x^{-1}$, $\delta_{yy} = E_y-2I+E_y^{-1}$, with $p = \Delta t/2\Delta x^2$ and $q = \Delta t/2\Delta y^2$ as the equation

$$(3)\quad (1-p\delta_{xx}-q\delta_{yy})\phi_y^{n+1} = (1+p\delta_{xx}+q\delta_{yy})\phi_y^n$$

The idea behind the splitting technique is to generate a perturbation of the above equation that permits a simpler computational process. Namely, we may factor equation (3) as follows.

$$(4)\quad (1-p\delta_{xx})(1-q\delta_{yy})\phi_y^{n+1} = (1+p\delta_{xx})(1+q\delta_{yy})\phi_y^n$$

Here, we add a term $(\Delta t^3/4)\phi_{xxyyt}$ of third order in time to the equation (4). The von Neumann stability analysis shows that the scheme is unconditionally stable which means there is no restriction on the time step Δt to the spacial steps Δx and Δy. Indeed, substituting $\phi = \hat{\phi}^k e^{i(mx+ny)}$ into equation (3) we obtain

$$|\hat{\phi}| = \left|\frac{1 - 2p(1-\cos\xi)}{1 + 2p(1-\cos\xi)}\right| \left|\frac{1 - 2q(1-\cos\eta)}{1 + 2q(1-\cos\eta)}\right|$$

with $\xi = m\,\Delta x$ and $\eta = n\,\Delta y$. By the fact that both p and q are positive we conclude that the right hand side is less than 1. This shows the amplification $|\hat{\phi}|$ is bounded by unity without any restriction on p and q.

The algorithm for thc solution of equation (4) consists of three easy steps:

$$X = (1+q\delta_{yy})\phi_y^n$$

$$(1-p\delta_{xx})Y = (1+p\delta_{xx})X$$

$$(1-q\delta_{yy})\phi_y^{n+1} = Y$$

Each of the last two steps requires a 3-diagonal matrix solver which is not expensive at all and can be found in any standard numerical method book [23,41].

It is worthwhile noting that equation (4) can be taken to represent an iterative procedure which converges if

$$\phi_{ij}^{n+1} = \phi_{ij}^{n} = \phi_{ij} \quad \text{for sufficiently large } n.$$

Then, equation (2) is reduced to the standard five-point difference approximation of the Laplace equations. In this case the quantity Δt can be viewed as an iteration parameter and may be varied from iteration to iteration to optimize the convergence of the process.

Equation (4) can be rewritten as

$$(1-\frac{\Delta t}{2}D_{xx})(1-\frac{\Delta t}{2}D_{yy})(\phi_{ij}^{n+1}-\phi_{ij}^{n}) = \Delta t(D_{xx}+D_{yy})\phi_{ij}^{n}$$

or

$$(5) \quad (\alpha-D_{xx})(\alpha-D_{yy})(\phi_{ij}^{n+1}-\phi_{ij}^{n}) = 2\alpha(D_{xx}+D_{yy})\phi_{ij}^{n}$$

It falls in the following general form [20,25],

$$(6) \quad Nc^{n} + \omega R^{n} = 0$$

which is used to solve the steady differential equation $L\phi = 0$. Here, $c^n = \phi^{n+1} - \phi^n$ is the correction, $R^n = L\phi^n$ is the residual which measures how well the finite difference equation is satisfied by the nth level solution ϕ^n, ω is a relaxation parameter and N is chosen as a product of two or more factors indicated by

$$N = N_1N_2$$

The factors N_1 and N_2 are chosen so that (1) their product is an approximation to L, (2) only simple matrix solvers are required, and (3) the overall scheme is stable.

This type of implicit scheme has been found very powerful in the calculation of steady flow. We remark that the parameter α in the equation (5) can be replaced by some lower order differential operator to speed up the convergence rate as well as to introduce damping which is needed in the multigrid technique [26].

2.3 Artificial Dissipation and Upwind Differencing

In inviscid flow calculation, a scheme that seems stable for shock free flows sometimes blows up when it is employed to calculate shock waves. This is due to the fact that using some difference formulas across a discontinuity can lead to oscillations which may grow. To remedy this, the well known shock capturing technique is to add to the inviscid flow equation a proper amount of artificial dissipation to simulate the physical dissipation in the shock layer and to provide the necessary damping for large wave number disturbances so that the shock wave is smeared out over several mesh points [28]. Namely, if we model the physical solution by the inviscid flow equation

$$u_t + \nabla\cdot f(u) = 0 \tag{1}$$

For shock calculations, we look at the solution u of (1) as limit of the viscous flow equation

$$u_t + \nabla\cdot f(u) = \nabla(\varepsilon\nabla u) \tag{2}$$

where ε is positive and is of the order of the mesh spacing.

Equation (2) is of diffusion type and the solution can be shown to exist [31]. Suppose that the solutions $u(\varepsilon)$ of (2) tend to a limit u boundedly almost everywhere as $\varepsilon \to 0$. Then, $u_t(\varepsilon)$ tends to u_t, $\nabla f(u(\varepsilon))$ to $\nabla f(u)$ and $\nabla(\varepsilon\ \nabla\cdot u)$ to 0 in the distribution sense. This says that u satisfies (1) in the distribution sense which is equivalent to saying that u satisfies the conservation law in the integral form.

The artificial viscosity can be viewed as a kind of truncation error exhibited by the approximation to the differential equations. It may be either in explicit or in implicit form. We consider the artificial viscosity introduced by upwind difference for the advection equation

$$(3) \qquad \phi_t + u\phi_x = 0$$

The finite difference approximation for the case u > 0 is

$$(4) \qquad \frac{\phi_i^{n+1}-\phi_i^n}{\Delta t} + \frac{u}{2\Delta x}\left\{(\phi_i^{n+1}-\phi_{i-1}^{n+1}) + (\phi_i^n-\phi_{i-1}^n)\right\} = 0$$

The von Neumann local stability analysis is to substitute a Fourier mode $\phi = \hat{\phi}^k e^{imx}$ into equation (4). This leads to

$$|\hat{\phi}| = \left| \frac{1 - p(1-\cos\xi) - ip\sin\xi}{1 + p(1-\cos\xi) + ip\sin\xi} \right| < 1$$

with $p = u\Delta t/2\Delta x > 0$ and $\xi = m\Delta x$. It is trivial that the scheme is unconditionally stable.

In equation (4), we did add an artificial viscosity implicitly through the upwind differencing in x. We can see it explicitly by Taylor series expansion. Equation (4) is equivalent to the equation

$$(5) \qquad \phi_t + u\phi_x = u\frac{\Delta x}{2}\phi_{xx} + O(\Delta t^2, \Delta x^2) .$$

The extra term $u(\Delta x/2)G_{xx}$ is the leading term in the truncation error and is referred to as the artificial viscosity.

To discuss the diffusion and dispersion properties of this finite differencing, we first derive the dispersion relation of the differential equation (3). Substituting $\phi = e^{i(\omega t+\xi x)}$ into the equation (3), yields the relation $\omega+u\xi = 0$. ω is a real number so that there is no damping of any wave mode and all waves have the same phase speed u.

Next, we apply the same Fourier mode to the viscous differential equation

$$(6) \qquad \phi_t + u\phi_x = |u| \frac{\Delta x}{2} \phi_{xx} .$$

It has the dispersion relation

$$\omega + u\xi = |u| \frac{\Delta x}{2} \xi^2 i$$

So a solution of equation (6) is

$$\phi = e^{i\xi(x-ut)} \cdot e^{-[|u|\frac{\Delta t}{2} \xi^2]t}$$

The magnitude of the damping increases with the wave number ξ and the velocity u. Hence, the effect of artificial viscosity is to introduce larger dissipation for both the larger wave number mode and the faster flow region. We see that there is no dispersion up to the first order approximation. However, if we add an extra term of ϕ_{xxx} to the right of equation (6) then dispersion occurs. This means that different frequency waves propagate with different speeds in the flow field.

The upwind differencing has played a very important role in transonic flow calculations. The main purpose is to exclude the expansion shock.

2.4 Conservative Finite Difference Schemes

The main idea behind the use of conservation form is the fact that if the difference equation to the differential equation in conservation form is again in conservation form, the solution of the finite difference equation satisfies the Rankine Hugoniot jump conditions automatically [30,39].

The differential conservation form

$$(1) \qquad u_t + \operatorname{div} f(u) = 0$$

can be derived from the more general integral form

$$\begin{aligned}\iint_R u\,dx\Big|_s^t + \int_s^t \int_{\partial R} f\cdot\hat{n}\,ds\,dt &= \int_s^t \iint_R u_t\,dx\,dt + \int_s^t \int_{\partial R} f\cdot n\,ds\,dt \\ (2) \qquad &= \int_s^t \iint_R u_t\,dx\,dt + \int_s^t \iint_R \nabla\cdot f\,dx\,dt \\ &= 0\end{aligned}$$

which says that the change in the amount of a substance with density u contained in the region R of space under consideration is due to the flux f of that substance across the boundary ∂R from time s to time t.

The conservative finite difference approximation is then defined having the form

$$(3) \qquad \sum_{\beta_R} \sum_r \left(\frac{u_{\beta r}^{n+1} - u_{\beta r}^{n-1}}{2\Delta t} \right) + \int_{t^{n-1}}^{t^{n+1}} \left(\sum_{\partial R} F_\alpha \right) dt = 0$$

which simulates the integral conservation form.

Our differencing strategy for the flow equation in conservation form yields the finite difference equation

$$(4) \qquad \sum_{\beta_R} \sum_r \left(\frac{u_{\beta r}^{n+1} - u_{\beta r}^{n-1}}{2\Delta t} \right) + \frac{\left(\sum_{\partial R} F_\alpha \right)^{n+1} + \left(\sum_{\partial R} F_\alpha \right)^{n-1}}{2} = 0$$

The question is how to solve for u^{n+1} for this large nonlinear system. Some linearization for F_α is needed.

III. POTENTIAL FLOW EQUATION

In the steady inviscid transonic flow calculation, the nonconservative form method agrees well with wind tunnel pressure data all the way up to the onset of buffet [18]. On the other hand, for the conservative form method, the agreement is less satisfactory and the adequate correlation with experimental data seems to be achieved by making correction with boundary layer shock wave interaction. For mesh sizes of practical interest, instead of doing a better simulation by combining a finer scale model of boundary layer shock wave interaction with conservative transonic equations, we pick up the nonconservative quasilinear potential flow equation as our model and develop a computer code for it.

We first discuss the characteristic surfaces of the equation and explain the domain of dependence for supersonic points. Then, we give a set of radiation boundary conditions which is shown to be very satisfactory with the numerical scheme we propose in Section IV. And, finally, we introduce the coordinate transformation such that the airfoil is fixed on a portion of coordinate line.

1. Characteristic Surface

It is helpful to know the characteristic surface of the flow equation on which the wave front along with information is propagated throughout the flow field. Let s and N be coordinates in the local stream and normal directions respectively. The direction cosines of s are u/q and v/q. ϕ_{ss} and ϕ_{NN} can be expressed locally in terms of the actual coordinates as

$$\phi_{ss} = \frac{1}{q^2}(u^2\phi_{xx} + 2uv\phi_{xy} + v^2\phi_{yy})$$
$$\phi_{NN} = \frac{1}{q^2}(v^2\phi_{xx} - 2uv\phi_{xy} + u^2\phi_{yy})$$

The potential flow equation in Cartesian coordinates locally aligned with the natural coordinate system (s,N) can be written as

$$\phi_{tt} + 2q\phi_{st} = (a^2 - q^2)\phi_{ss} + a^2\phi_{NN} .$$

The characteristic surface satisfies the equation

$$(q^2 - a^2)t^2 - 2qst + s^2 - (\frac{q^2-a^2}{a^2})N^2 = 0 .$$

As shown in Figure 2, on the (N,t) plane, the characteristic equation is reduced to

$$a^2t^2 - N^2 = 0 \quad \text{or} \quad (N-at)(N+at) = 0 .$$

The disturbance propagation speed is a.

On the (s,t) plane, the characteristic equation is reduced to

$$(qt - s)^2 = a^2t^2$$

or

$$(s-(q+a)t)(s-(q-a)t) = 0 .$$

The particle speed is q, the upstream propagation speed is q-a and the downstream propagation speed is q+a. Thus the disturbance information is propagated by the Doppler shifted sound wave velocity. For transonic flow, the particle and downstream waves quickly travel away from the airfoil but upstream waves remain in the vicinity of the airfoil for a much longer time. The slow waves force a slow approach to a steady state solution, while the fast waves stipulate a small time step by the CFL condition $\Delta t \leq \Delta x/(q+a)$.

If a new time coordinate $T = t + qs/(a^2-q^2)$ is introduced, then the potential flow equation can be expressed as

$$(a^2 - q^2)\phi_{ss} + a^2\phi_{NN} = \frac{a^2}{(a^2-q^2)}\,\phi_{TT}$$

So for supersonic points, $a^2 \leq q^2$, T is a space-like direction and s is a time-like direction. This means the differencing in the s-direction should be retarded in the supersonic region in order to have the right domain of dependence. For subsonic points, $a^2 > q^2$, s is a space-like direction and T is actually a time-like direction.

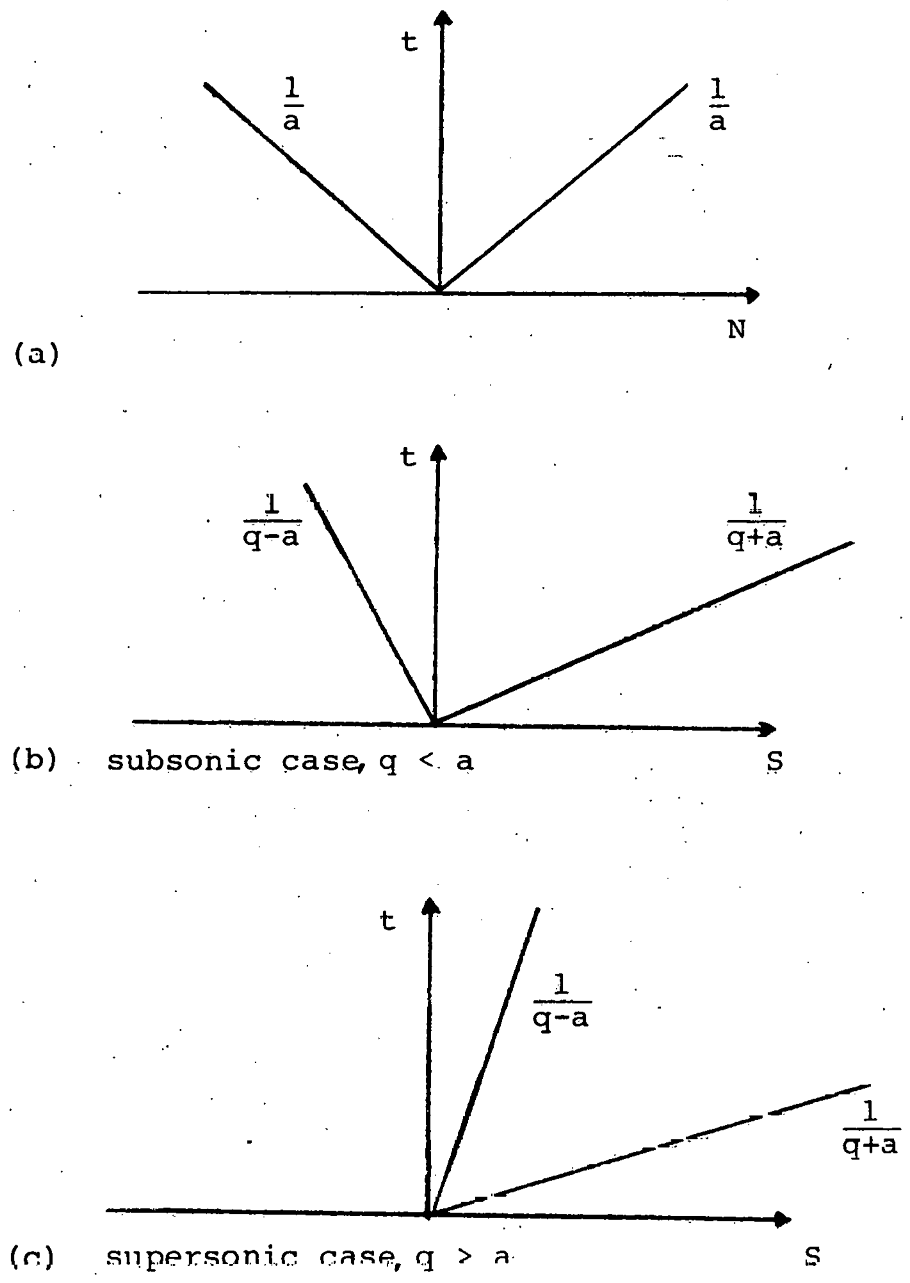


Figure 2. Characteristic Surface of the Potential Flow Equation in Quasilinear Form; (a) (N,t) plane; (b),(c) (S,t) plane.

2. Computational Boundary Conditions

Problems of transonic flow field are usually posed in the exterior of the body which is an unbounded domain [4]. Owing to the finite storage capability of the computer, the numerical computations require that the computational domain be finite. The proper boundary conditions must be developed at these computational boundaries so that the computed solution closely approximates the free space solution which exists in the absence of these computational boundaries [15,16,17].

For steady state calculations in transonic flow, coordinate mapping techniques are a traditional and effective way of handling these computational boundary problems. The reason for the success of coordinate mapping techniques lies in the fact that the steady state far field asymptotic behavior is given by a regular algebraic singularity without oscillation. For genuinely unsteady transonic phenomena, the solution of flow equations usually possesses a strongly oscillatory transient behavior and the far-field asymptotic behavior is an oscillatory singularity. The standard coordinate mapping technique is not adequate to resolve this problem. It must be supplemented by a set of proper boundary conditions at the computational boundaries.

In this section we will give a set of radiation boundary conditions for the potential equation in the Cartesian coordinate system and in a later section we will give its corresponding form in the computational domain. In the physical domain, the computational region for an airfoil in two dimensions is depicted as

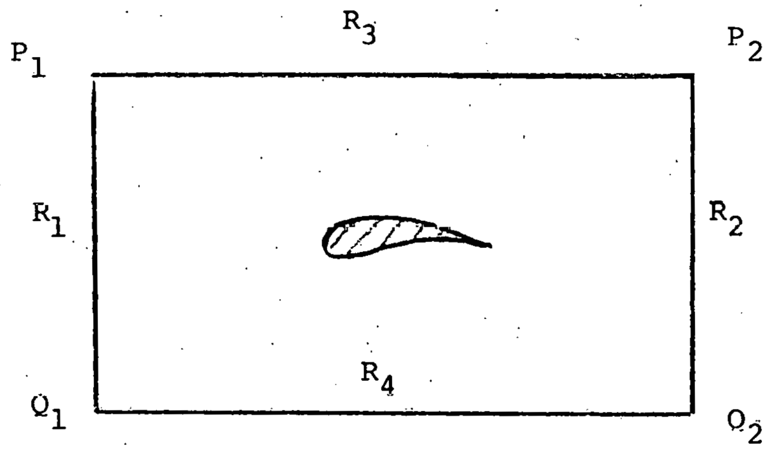


Figure 3. The Typical Computational Region for an Airfoil.

The design of effective far field radiation boundary condition depends on the wave propagation properties of the flow equation. We consider the potential flow equation

(1) $\phi_{tt} + 2u\phi_{xt} + 2v\phi_{yt} = (a^2-u^2)\phi_{xx} - 2uv\phi_{xy} + (a^2-v^2)\phi_{yy}$

For a plane wave $\hat{\phi} = e^{i(\omega t+\xi x+\eta y)}$ to satisfy equation (1), it requires that its wave information satisfy

(2) $\omega^2 + 2u\xi\omega + 2v\eta\omega = (a^2-u^2)\xi^2 - 2uv\xi\eta + (a^2-v^2)\eta^2$
or $(\omega + u\xi + v\eta)^2 = a^2(\xi^2 + \eta^2)$.

A boundary condition on the upstream wall, R_1 boundary, which annihilates the upstream propagating wavelet is given by

(4) $$\omega + u\xi + v\eta = a\xi\sqrt{1+\eta^2/\xi^2}$$

Recalling the dual relationship between $i\omega$, $i\xi$, $i\eta$ and D_t , D_x , D_y respectively, the equation stands for a nonlocal condition. By the first approximation of $\sqrt{1+x} = 1 + 1/2\ x - \frac{1}{8}x^2 + O(x^3)$ we get the first radiation condition for R_1 boundary, namely, $\omega + u\xi + v\xi = a\xi$ which leads, after Fourier transformation, to the condition

(5) $$\phi_t + (u-a)\phi_x + v\phi_y = 0$$

Similarly, we can derive the artificial boundary conditions for the R_2 , R_3 and R_4 boundaries. At their intersection points P_1 , P_2 , Q_1 and Q_2 , we use the average of the corresponding conditions, and have the following general formula

(6) $$\phi_t + \bar{u}\phi_x + \bar{v}\phi_y = 0$$

with $\bar{u} + i\bar{v} = (u + iv) + ae^{i\beta}$ where $\beta = -\frac{\pi}{4}, \frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}$ at Q_2 , P_2 m Q_1 and Q_1 and $\beta = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$, on R_2 , R_3 , R_1 and R_4 respectively.

3. Coordinate Transformation Technique

When the body surface crosses the coordinate lines it is difficult to satisfy the physical boundary conditions. This is particularly the case near the leading edge of the modern supercritical wing section where the surface has a high curvature and the flow is sensitive to small variations in the shape [41]. For the rigid body motion of the airfoil the treatment is facilitated by the use of a moving sheared parabolic coordinate system in which the body contour coincides with a segment of coordinate line and the whole mesh system is moving with the wing section so that the relative position of grid points is kept.

We describe the moving sheared parabolic coordinate system as follows [3,25]:

3.1 Coordinate System

First, we consider the physical plane to be described in a Cartesian coordinate system (x,y), and the airfoil attached coordinate system in Cartesian coordinate system $(x*,y*)$. Let the origin of $(x*,y*)$ system be at the singular point of the parabolic mapping which unwraps the airfoil and will be described in the next step. If the flight velocity of the airfoil is $M_*e^{i(\pi-\theta)}$ at time t, then the position of the origin of the (x^*,y^*) system can be described as $O^*O = \int_0^t M_*e^{i(\pi-\theta)}\,ds$. If the angle of attack of the airfoil at time t is α, then the x^*-axis on which the airfoil chord lies will have an angle $-(\alpha + \theta)$ with respect to the x-axis. Their relation can be seen in Figure 4 and described as the relation

$$(1) \quad (x+iy) = \int_0^t M_*(s)\, e^{i(\pi-\theta(s))}\, ds + (x^*+iy^*)\, e^{-i(\theta+\alpha)}$$

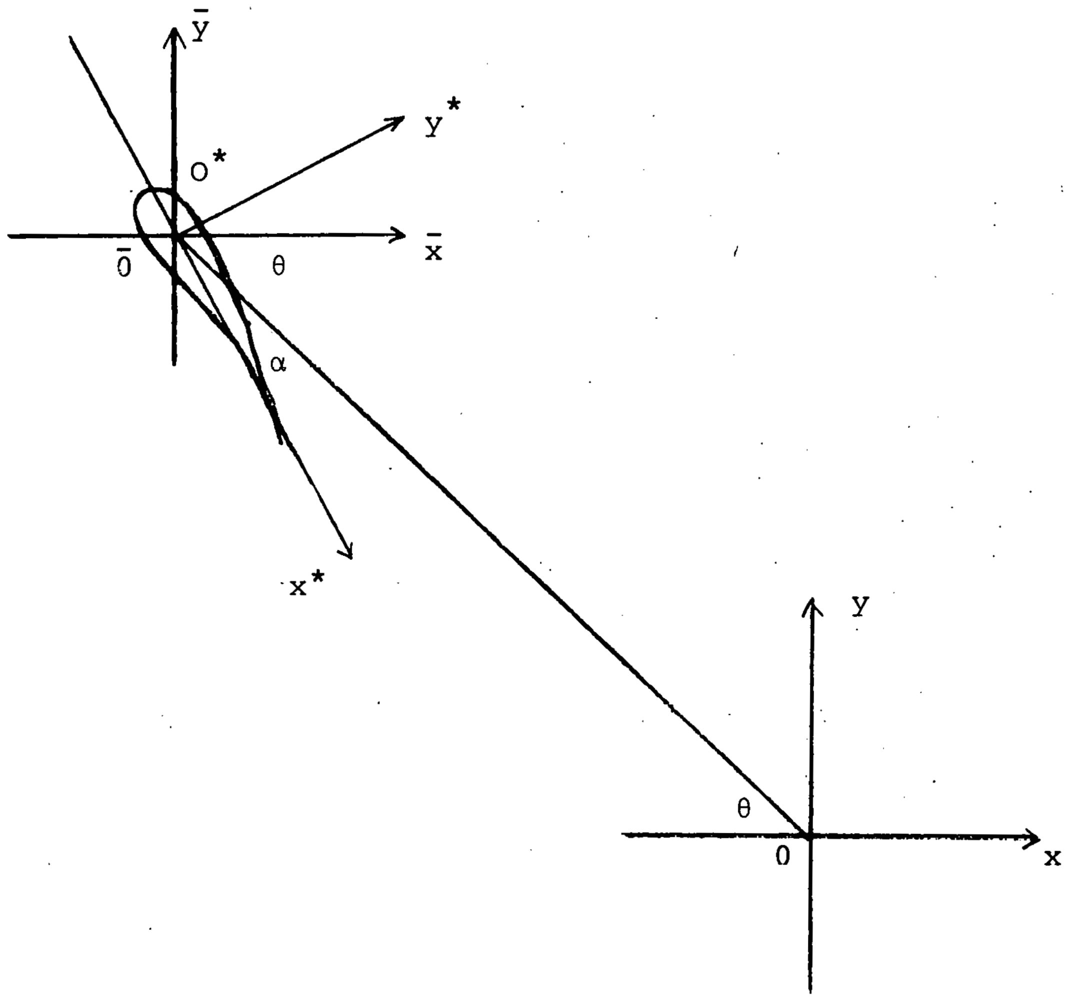


Figure 4. Frames of Reference.

Second, we unwrap the airfoil by introducing the square root mapping

$$2(x^* + iy^*) = (x_1 + iy_1)^2 \tag{2}$$

which maps the entire airfoil contour to a shallow bump near $y_1 = 0$, as shown in Figure 5b.

Third, if we denote the height of the bump as $y_1 = s(x_1)$, then the shearing transformation

$$X + iY = x_1 + i(y_1 - s(x_1)) \tag{3}$$

reduces the airfoil contour to a portion of the line $Y = 0$.

Fourth, we stretch the coordinate line by the stretch mapping to render the computational domain finite. The stretch mapping, for instance,

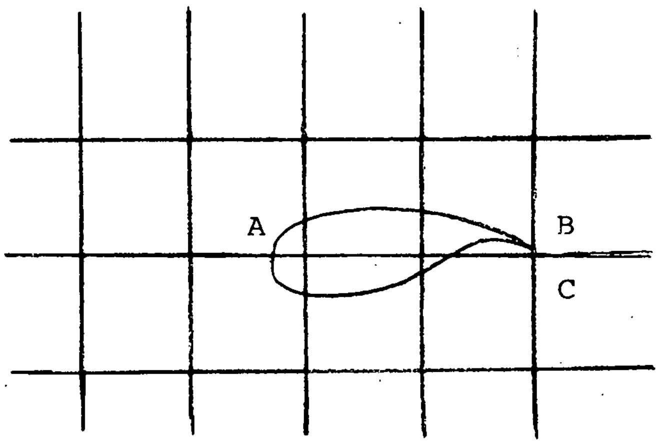
$$Y = \frac{b\bar{Y}}{(1 - \bar{Y}^2)^a} \quad , \quad 0 \le a \le 1 \tag{4}$$

will map the infinite lines $Y = \pm\infty$ to $\bar{Y} = \pm 1$.

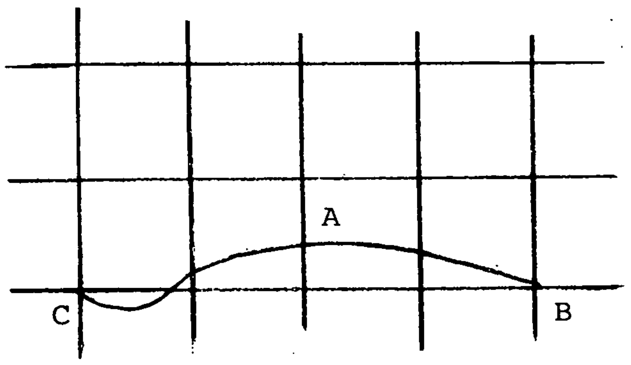
Fifth, avoiding discontinuities at the trailing edge of the wing section, the branch cut is contined smoothly downstream. In physical space, the continuation is represented by

$$\bar{y} = \bar{y}_{te} + \tau[\bar{x}_{te} - \bar{x}^*] \; \frac{\ell n \left[\dfrac{\bar{x} - \bar{x}^*}{\bar{x}_{te} - \bar{x}^*} \right]}{\left[\dfrac{\bar{x} - \bar{x}^*}{\bar{x}_{te} - \bar{x}^*} \right]} \tag{5}$$

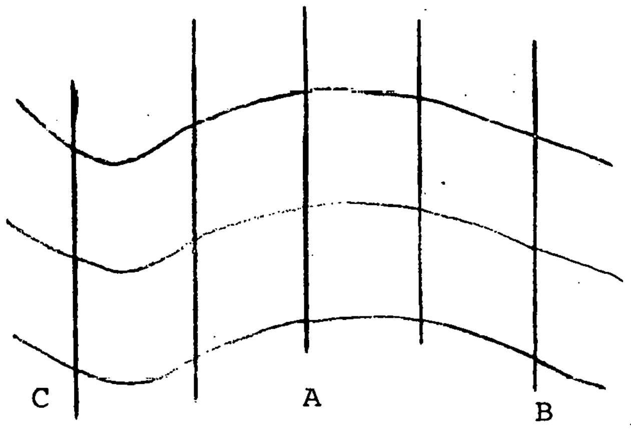
where τ is the mean of the upper and lower surface slopes at the trailing edge $(\bar{x}_{te}, \bar{y}_{te})$ and $\bar{x}^*$ is a suitably chosen scaling constant (usually taken as the ordinate of the local quarter-chord point).



(a) Cartesian coordinates



(b) Parabolic coordinates



(c) Sheared parabolic coordinates

Figure 5.

3.2 Flow Equation

The transformations (1) and (2) are conformal. We will write the flow equation on the (x_1,y_1) coordinate system, and use the chain rule to convert the equation into the $(\bar{x},\bar{y})$ system. Several key formulas are written down for reference. We begin with some notation.

Let $z = x + iy$, $z^* = x^* + iy^*$,

$z_1 = x_1 + iy_1$, $Z = X + iY$, $\bar{Z} = \bar{X} + i\bar{Y}$

Then the mapping (1) and (2) may be expressed as the following compact forms.

$$z = \int_0^t M_*(s)\, e^{i(\pi-\theta(s))}\, ds + z^*\, e^{-i(\theta(t)+\alpha(t))}$$

$$z_1^2 = 2z^*$$

The modulus of the mapping function to the z_1 plane can be evaluated as

$$H = \left|\frac{dz}{dz_1}\right| = \sqrt{x_1^{\,2}+y_1^2} \quad ; \text{ thus } \nabla = \left(\frac{d}{H\, dx_1}, \frac{d}{H\, dy_1}\right)$$

The velocity components in the (x_1,y_1) system:

$$u = \frac{\phi_{x_1}}{H} \quad , \quad v = \frac{\phi_{y_1}}{H}$$

The chain rule gives the relation for ϕ_t in the (x_1,y_1) system:

$$\phi_t\Big|_{z\text{ fixed}} = \phi_{T_1} + \phi_{x_1}\frac{dx_1}{dt}\Big|_{z\text{ fixed}} + \phi_{y_1}\frac{dy_1}{dt}\Big|_{z\text{ fixed}}$$

where dx_1/dt and dy_1/dt can be found as follows. Since

$$z_1{}^2 = 2z^* = 2e^{i(\theta+\alpha)}\left\{ z - \int_0^t M_* \, e^{i(\pi-\theta)} \, ds \right\}$$

we take differentiation with respect to time t and hold z fixed. Hence

$$\left.\frac{dz_1}{dt}\right|_{z \text{ fixed}} = \frac{z_1}{2}\,[i(\theta_t+\alpha_t)] + \frac{Me^{i\alpha}}{z_1}$$

We remark that $\phi_t = \phi_{T_1} - (v_R\cdot\nabla)\phi$ if v_R is the relative velocity of the origin of the (x_1, y_1) system to the (x,y) syatem [36]. Then, we conclude that

$$V_R = -\left(H\frac{dx_1}{dt}, \; H\frac{dy_1}{dt}\right) .$$

The same differentiation applied twice on z_1 yields

$$\left.\frac{d^2z_1}{dt^2}\right|_{z \text{ fixed}} = \frac{z_1}{2}\,[i(\theta_{tt}+\alpha_{tt})] + \frac{\bar{z}_1}{H^2}\left\{\frac{dM_*}{dt} + M_*(i\alpha_t)\right\}e^{i\alpha} - \frac{(\theta_t+\alpha_t)^2}{4}\,z_1 - \frac{M_*^2}{z_1^3}\,e^{i2\alpha} ,$$

which will be needed in the evaluation of the following term:

$$\phi_{tt} - \phi_{T_1T_1} + \phi_{x_1T_1}\frac{dx_1}{dt} + \phi_{x_1T_1}\frac{dy_1}{dt} + \phi_{x_1}\frac{d^2x_1}{dt^2} + \phi_{y_1}\frac{d^2y_1}{dt^2}$$
$$+ \left(\phi_{x_1T_1} + \phi_{x_1x_1}\frac{dx_1}{dt} + \phi_{x_1y_1}\frac{dy_1}{dt}\right)\frac{dx_1}{dt}$$
$$+ \left(\phi_{y_1T_1} + \phi_{x_1y_1}\frac{dx_1}{dt} + \phi_{y_1y_1}\frac{dy_1}{dt}\right)\frac{dy_1}{dt}$$

Similarly, the chain rule gives

$$\phi_{x_1t} = \phi_{x_1T_1} + \phi_{x_1x_1}\frac{dx_1}{dt} + \phi_{x_1y_1}\frac{dy_1}{dt} + \phi_{x_1}\left(\frac{dx_1}{dt}\right)_{x_1} + \phi_{y_1}\left(\frac{dy_1}{dt}\right)_{x_1}$$

$$\phi_{y_1t} = \phi_{y_1T_1} + \phi_{x_1y_1}\frac{dx_1}{dt} + \phi_{y_1y_1}\frac{dy_1}{dt} + \phi_{y_1}\left(\frac{dy_1}{dt}\right)_{y_1} + \phi_{y_1}\left(\frac{dy_1}{dt}\right)_{y_1}$$

Recalling that

$$\frac{1}{2}(\nabla\phi\nabla)(|\nabla\phi|^2) = \frac{1}{H^2}\left\{u^2\phi_{x_1x_1} + 2uv\phi_{x_1y_1} + v^2\phi_{y_1y_1} - \left(\frac{u^2+v^2}{H}\right)(ux_1 + vy_1)\right\}$$

and

$$\Delta\phi = \frac{1}{H^2}\{\phi_{x_1x_1} + \phi_{y_1y_1}\}$$

Finally, we can write down the potential flow equation in the (x_1,y_1) frame as the partial differential equation

$$\phi_{T_1T_1} + 2\frac{u_r}{H}\phi_{x_1T_1} + 2\frac{v_r}{H}\phi_{y_1T_1} + \phi_{x_1x_1}\frac{u_r^2}{H^2} + 2\phi_{x_1y_1}\frac{u_rv_r}{H^2}$$
$$+ \phi_{y_1y_1}\frac{v_r^2}{H^2} + \phi_{x_1}\left\{\frac{d^2x_1}{dt^2} + 2\frac{u}{H}\left(\frac{dx_1}{dt}\right)_{x_1} + \frac{2v}{H}\left(\frac{dy_1}{dt}\right)_{y_1}\right\}$$
$$+ \phi_{y_1}\left\{\frac{d^2y_1^2}{dt^2} + 2\frac{u}{H}\left(\frac{dy_1}{dt}\right)_{x_1} + \frac{2v}{H}\left(\frac{dy_1}{dt}\right)_{y_1}\right\} - \frac{u^2+v^2}{H^3}(ux_1+vy_1)$$
$$= \frac{a^2}{H^2}\{\phi_{x_1x_1} + \phi_{y_1y_1}\}$$

where $u_r = u + H\frac{dx_1}{dt}$ and $v_r = v + H\frac{dy_1}{dt}$.

The shearing and the stretching transformations will further bring the flow equation in much more complicated form. We will not write down the flow equation in the $(\bar{X},\bar{Y})$ frame here. Instead, we write the useful formulas

$$\phi_{x_1} = \phi_X - s_X\phi_Y$$

$$\phi_{y_1} = \phi_Y$$

$$\phi_{x_1x_1} = \phi_{XX} - 2s_X\phi_{XY} + s_X^2\phi_{YY} - s_{XX}\phi_Y$$

$$\phi_{x_1y_1} = \phi_{XY} - s_X\phi_{YY}$$

$$\phi_{y_1y_1} = \phi_{YY}$$

$$\phi_{x_1T_1} = \phi_{XT} - s_X\phi_{YT}$$

$$\phi_{y_1T_1} = \phi_{YT}$$

and

$$\phi_X = \phi_{\bar{X}} \frac{d\bar{X}}{dX}$$

$$\phi_Y = \phi_{\bar{Y}} \frac{d\bar{Y}}{dY}$$

$$\phi_{XX} = \phi_{\bar{X}\bar{X}} \left(\frac{d\bar{X}}{dX}\right)^2 + \phi_{\bar{X}} \frac{d^2\bar{X}}{dX^2}$$

$$\phi_{XY} = \phi_{\bar{X}\bar{Y}} \left(\frac{d\bar{X}}{dX}\right)\left(\frac{d\bar{Y}}{dY}\right)$$

$$\phi_{YY} = \phi_{\bar{Y}\bar{Y}} \left(\frac{d\bar{Y}}{dY}\right)^2 + \phi_{\bar{Y}} \left(\frac{d^2\bar{Y}}{dY^2}\right)$$

3.3 Body Surface Condition

The velocity observed in the (x_1, y_1) frame is $q_r = (u_r, v_r) = \nabla\phi - V_R$. Thus, the nonpenetrating surface condition requires that $\tilde{q}_r \cdot \tilde{n} = 0$, which leads to

$$\phi_{y_1} = s_{x_1}\phi_{x_1} + H^2\left\{s_{x_1}\frac{dx_1}{dt} - \frac{dy_1}{dt}\right\}$$

3.4 Wake Condition

The zero pressure jump in the wake which lies on the portion of the singular line along which the airfoil is opened up can be [21] expressed as

$$[\phi_{T_1}] + \frac{dx_1}{dt}[\phi_{x_1}] + \frac{\bar{u}}{H}[\phi_{x_1}] = 0$$

where $\bar{u}$ is the average of the upper and lower wake velocities.

3.5 Computational Boundary Conditions

The computational domain is depicted as

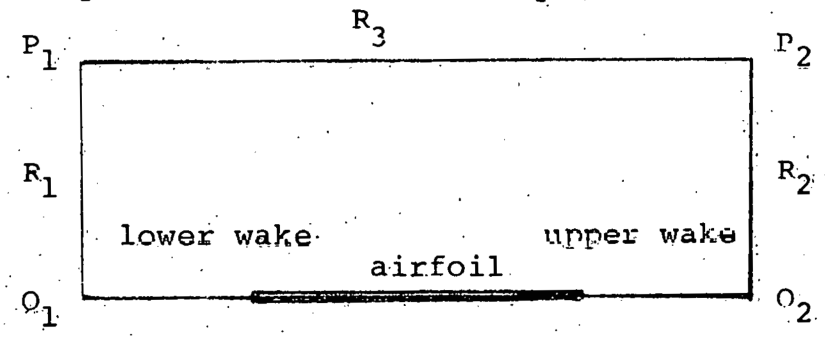


Figure 6.

The radiation boundary condition is of the form

$$\phi_{T_1} + \frac{dx_1}{dt}\phi_{x_1} + \frac{dy_1}{dt}\phi_{y_1} + \bar{u}\phi_{x_1} + \bar{v}\phi_{y_1} = 0$$

where $\bar{u}$ and $\bar{v}$ are defined as before.

IV. NUMERICAL METHOD

In this section, we apply the idea introduced before to design our numerical solver for potential flow equation in quasilinear form. First, we use type dependent differencing to introduce the proper amount of dissipation into the finite difference approximation such that the scheme is stable and shock waves are captured automatically. Then, we factor the finite difference equation into one-dimensional factors so that we are able to solve the equation efficiently by employing a 5-diagonal matrix solver. Since the disturbance of potential flow is propagated by the total effect of advection and wave propagation, we examine the stability of our method to two linear models: advection and wave equations. The local stability analysis shows that our finite differencing strategy and approximate factorization technique result in unconditionally stable schemes for these two models.

1. Finite Difference Scheme

The finite difference scheme is a time marching alternating direction implicit scheme. Before we write down the differencing strategy we need the following convention:

$\phi N = \phi^{n+1} - \phi^{n}$

$\phi M = \phi^{n} - \phi^{n-1}$

D = central difference

$\overleftarrow{D}$ = upwind difference

$\overline{D}$ = Type dependent difference

To be specific, we define the operators in x-direction,

$$D_x\phi_i = \frac{\phi_{i+1}-\phi_{i-1}}{2\Delta x}$$

$$u\overleftarrow{D}_x\phi_i = \begin{cases} u^*\left(\frac{\phi_i-\phi_{i-1}}{\Delta x}\right) & \text{if } u > 0 \\ u^*\left(\frac{\phi_{i+1}-\phi_i}{\Delta x}\right) & \text{if } u < 0 \end{cases}$$

$$\overline{D}_x = \begin{cases} D_x & \text{for subsonic point} \\ \overleftarrow{D}_x & \text{for supersonic point .} \end{cases}$$

1.1 <u>On interior point of computational domain</u>

The potential flow equation in Cartesian coordinates locally aligned with the natural coordinates system assumes the canonical form [21]

$$\phi_{tt} + 2q\phi_{st} = (a^2-q^2)\phi_{ss} + a^2\phi_{NN}$$

with q = (u,v) the velocity, s and N are coordinates in the local stream and normal directions.

Basically, the velocity components u and v at the grid point are evaluated by using central difference at time level n. The ϕ_t term in the Bernoulli equation is evaluated by backward difference in the natural way.

1.11 <u>For ϕ_{tt} term</u>, central difference will be used for temporal derivative,

$$\phi_{tt} = \frac{1}{(\Delta t)^2}(\phi^{n+1}-2\phi^n+\phi^{n-1}) = \frac{1}{(\Delta t)^2}(\phi N-\phi M)$$

1.12 <u>For contributions to ϕ_{st}</u>, upwind differences will be used for all spatial derivatives and central difference will be used for temporal derivagives,

$$\begin{aligned} 2q\phi_{st} &= 2u\phi_{xt} + 2v\phi_{yt} \\ &= (2u\overleftarrow{D}_x+2v\overleftarrow{D}_y)\phi_t \\ &= (2u\overleftarrow{D}_x+2v\overleftarrow{D}_y)\left(\frac{\phi^{n+1}-\phi^{n-1}}{2\Delta t}\right) \\ &= \frac{1}{\Delta t}(u\overleftarrow{D}_x+v\overleftarrow{D}_y)(\phi N+\phi M). \end{aligned}$$

1.13 <u>For contributions to ϕ_{ss}</u>, type dependent differences will be used for all terms. The term ϕ^n is substituted by the mean of ϕ^{n+1} and ϕ^{n-1}, i.e., $\phi^n = \frac{\phi^{n+1}+\phi^{n-1}}{2}$,

$$\begin{aligned} \phi_{ss} &= \frac{1}{q^2}(u^2\phi_{xx}+2uv\phi_{xy}+v^2\phi_{yy}) \\ &= \frac{1}{q^2}(u^2\bar{D}_{xx}+2uv\bar{D}_{xy}+v^2\bar{D}_{yy})\left(\frac{\phi^{n+1}+\phi^{n-1}}{2}\right) \\ &= \frac{1}{q^2}(u^2\bar{D}_{xx}+2uv\bar{D}_{xy}+v^2\bar{D}_{yy})\left(\frac{\phi N-\phi M+2\phi^n}{2}\right). \end{aligned}$$

1.14 <u>For contributions to ϕ_{NN}</u>, central differences will be used for all terms. The term ϕ^n is again replaced by the mean of ϕ^{n+1} and ϕ^{n-1}.

$$\begin{aligned}\phi_{NN} &= \frac{1}{q^2}(v^2\phi_{xx}-2uv\phi_{xy}+u^2\phi_{yy})\\ &= \frac{1}{q^2}(v^2D_{xx}-2uvD_{xy}+u^2D_{yy})\left(\frac{\phi N-\phi M+2\phi^n}{2}\right).\end{aligned}$$

Finally, the finite difference approximation can be written as

$$\begin{aligned}&\frac{\phi N-\phi M}{(\Delta t)^2}+\frac{1}{\Delta t}(u\overleftarrow{D}_x+v\overleftarrow{D}_y)(\phi N+\phi M)\\ &= \frac{(a^2-q^2)}{q^2}(u^2\bar{D}_{xx}+2uv\bar{D}_{xy}+v^2\bar{D}_{yy})\left(\frac{\phi N-\phi M+2\phi^n}{2}\right)\\ &\quad+\frac{a^2}{q^2}(v^2D_{xx}-2uvD_{xy}+u^2D_{yy})\left(\frac{\phi N-\phi M+2\phi^n}{2}\right)\end{aligned}$$

or

$$\begin{aligned}&\left[1+\Delta tu\overleftarrow{D}_x+\Delta tv\overleftarrow{D}_y-\left(\frac{a^2-q^2}{q^2}\right)\left(\frac{\Delta t^2}{2}\right)(u^2\bar{D}_{xx}+2uv\bar{D}_{xy}+v^2\bar{D}_{yy})\right.\\ &\quad\left.-\frac{a^2}{q^2}\left(\frac{\Delta t^2}{2}\right)(v^2D_{xx}-2uvD_{xy}+u^2D_{yy}\right]\phi N\\ &= (\Delta t)^2\left[\left(\frac{a^2-q^2}{q^2}\right)(u^2\bar{D}_{xx}+2uv\bar{D}_{xy}+v^2\bar{D}_{yy})+\frac{a^2}{q^2}(v^2D_{xx}-2uvD_{xy}+u^2D_{yy})\right]\phi^n\\ &\quad+\left[1-(\Delta t)(u\overleftarrow{D}_x+v\overleftarrow{D}_y)-\frac{(\Delta t)^2}{2}\left(\frac{a^2-q^2}{q^2}\right)(u^2\bar{D}_{xx}+2uv\bar{D}_{xy}+v^2\bar{D}_{yy})\right.\\ &\quad\left.-\frac{(\Delta t)^2}{2}\left(\frac{a^2}{q^2}\right)(v^2D_{xx}-2uvD_{xy}+u^2D_{yy})\right]\phi M \qquad (1)\end{aligned}$$

The discretization errors associated with the finite difference approximation is of first order in space and second order in time. The leading error terms in the space derivative introduce the desired shock viscosity. The system of algebraic equation generated by the equation (1) is large and cannot be solved efficiently. However, this equation can be factored within the same order of accuracy in time and space by the spirit of splitting technique. The following factorization has been tested and found to be numerically stable with time steps much larger than the time step allowed by the CFL condition for explicit methods. Let $M^2 = q^2/a^2$

$$L_x = [1 + \Delta t u\overleftarrow{D}_x + \frac{\Delta t^2}{2} u^2\bar{D}_{xx} - \frac{1}{M^2}(\frac{\Delta t^2}{2})(u^2\bar{D}_{xx}+v^2 D_{xx})]$$

and

$$L_y = [1 + \Delta t v\overleftarrow{D}_y + \frac{\Delta t^2}{2} v^2\bar{D}_{yy} - \frac{1}{M^2}(\frac{\Delta t^2}{2})(v^2\bar{D}_{yy}+u^2 D_{yy})]$$

Then, the approximate factorization of the equation (1) can be written as

$$L_x \cdot L_y \phi N = RHS \qquad (2)$$

This factorization reduces the large complicated matrix inversion problem to two one-dimensional problems. The algorithm can be expressed as

$$\begin{cases} L_x X = RHS \\ L_y \phi N = X \end{cases}$$

Each of the above steps requires a 5-diagonal matrix solver which will be described in the Appendix A.

1.2 <u>On the boundary points of the computation domain</u>

The artificial radiation boundary condition is of the general form

$$\phi_t + \bar{u}\phi_x + \bar{v}\phi_j = 0$$

We approximate it by

$$\frac{\phi^{n+1}-\phi^{n-1}}{2\Delta t} + (\bar{u}\overleftarrow{D}_x+\bar{v}\overleftarrow{D}_y)\frac{\phi^{n+1}+\phi^{n-1}}{2} = 0$$

which can be expressed as

$$(1+\Delta t\bar{u}\overleftarrow{D}_x+\Delta t\bar{v}\overleftarrow{D}_y)\phi N = (-1+\Delta t\bar{u}\overleftarrow{D}_x+\Delta t\bar{v}\overleftarrow{D}_y)\phi M$$
$$- 2\Delta t(u\overleftarrow{D}_x+v\overleftarrow{D}_y)\phi^n$$

which can be factored within first order in space and second order time as

$$(1+\Delta t\bar{u}\overleftarrow{D}_x)(1+\Delta t\bar{v}\overleftarrow{D}_y)\phi N = \text{RHS}$$

The algorithm consists of the following two steps.

$$(1+\Delta t\bar{u}\overleftarrow{D}_x)X = \text{RHS}$$
$$(1+\Delta t\bar{v}\overleftarrow{D}_y)\phi N = X$$

1.3 <u>Wake condition</u>

As before, we assume that both pressure and normal velocity components are continuous across the wake which is assumed to lie on a segment of the x-axis. The wake condition is

$$[\phi_t] + \bar{u}[\phi_x] = 0 \text{ on the wake}$$

where $\bar{u}$ is the average velocity of the upper and lower wake velocities.

The finite difference approximation is given by

$$\frac{1}{\Delta t}\{[\phi_i^n]-[\phi_i^{n-1}]\} + \frac{\bar{u}}{\Delta x}\{[\phi]_i^n-[\phi_{i-1}^n]\} = 0$$

Let $B = \frac{\bar{u}\Delta t}{\Delta t}$ then

$$[\phi_i^n] = \frac{[\phi_i^{n-1}]+B[\phi_{i-1}^n]}{1+B} \tag{1}$$

Hence, once the jump at the tail point has been estimated, all the jump in the wake can be calculated from the equation (1).

We remark that the artificial viscosity we have added in the finite difference scheme is of amount

$$\{h(\text{sign } u)uu_{xt} + k(\text{sign } v)vv_{yt}\}$$

$$+ \mu\{h(\text{sign } u)u(uu_{xx}+vv_{xx}) + k(\text{sign } v)v(uu_{yy}+vv_{yy})\}$$

with $\mu = \max\{0; (1 - \frac{1}{M^2})\}, h = \Delta x$, and $k = \Delta y$.

The term in first braces is an advection viscosity which will damp out some noise generated either by the artificial boundaries or the body surface. The term in second braces is the desired shock viscosity. The whole artificial viscosity can be cast into the divergence form

$$P_x + Q_y$$

with $P = h\ \{(\text{sign } u)uu_t + \mu(\text{sign } u)u(\frac{u^2+v^2}{2})_x\}$

and $Q = k\ \{(\text{sign } v)vv_t + \mu(\text{sign } v)v(\frac{u^2-v^2}{2})_y\}$

Failure to maintain proper conservation form can result in computed shock speeds that depend on grid spacing.

2. <u>Analysis for the Finite Differencing Strategy and The Approximate Factorization Process</u>

We have shown that the disturbance information in the potential flow field is propagated as the Doppler sound wave which consists of the advection and wave propagation effects. The potential flow equation is nonlinear. As a guide to the stability of the difference scheme, we consider two linear models, the advection and the wave equations.

2.1 The radiation boundary conditions is modeled by the two-dimensional advection equation

$$\phi_t + u\phi_x + v\phi_j = 0 \tag{1}$$

Our finite differencing strategy says that

$$\frac{\phi^{n+1}-\phi^{n-1}}{2\Delta t} + (u\overleftarrow{D}_x + v\overleftarrow{D}_y)(\frac{\phi^{n+1}+\phi^{n-1}}{2}) = 0 \tag{2}$$

We examine the amplification of a Fourier mode. Substituting $\phi = \hat{\phi}^{\kappa} e^{mx+ny}$ for ϕ at the κ-level, the growing factor $\hat{\phi}$ is governed by

$$\hat{\phi}^2 - 1 = (pu(e^{-i\xi}-1) + qv(e^{-i\eta}-1))(\hat{\phi}^2+1)$$

for the case that $u > 0$ and $v > 0$ where $p = \Delta t/\Delta x$, $q = \Delta t/\Delta y$, $\xi = m\Delta x$, $\eta = n\Delta y$. Hence,

$$|\hat{\phi}^2| = \left|\frac{1-pu(1-\cos\xi)-qv(1-\cos\eta)-i(pu\sin\xi+qv\sin\eta)}{1+pu(1-\cos\xi)+qv)1-\cos\eta)+i(pu\sin\xi+qv\sin\eta)}\right| < 1$$

So it is unconditionally stable for this case.

The scheme for the other cases where either u, or v, or both of them may be negative are easily shown to be unconditionally stable.

Next, we examine the approximate factorization method for this finite difference approximate for the advection equation. Our approximate factorization says that (2) can be factored as

$$(1+\ tu\overleftarrow{D}_x)(1+\ tv\overleftarrow{D}_y)\phi N = -(1-\Delta tu\overleftarrow{D}_x)(1-\Delta tv\overleftarrow{D}_y)\phi M\ -2\Delta t(u\overleftarrow{D}_x+v\overleftarrow{D}_y)\phi^n$$

By Fourier analysis, we substitute $\phi = \hat{\phi}^{\kappa}e^{i(mx+ny)}$.

$$[1+pu(1-e^{-i\xi})][1+qv(1-e^{-i\eta})](\hat{\phi}^2-\hat{\phi})$$

$$= -[1-pu(1-e^{-i\xi})][1-qv(1-e^{-i\eta})](\hat{\phi}-1)$$

$$-2[pu(1-e^{-i\xi}) + qv(i-e^{-i\eta})]\hat{\phi}$$

$$|\hat{\phi}^2| - \left|\frac{1-pu(1-\cos\xi)+ipu\sin\xi}{1+pu(1-\cos\xi)-ipu\sin\xi}\right|\left|\frac{1-qv(1-\cos\eta)+iqv\sin\eta}{1+qv(1-\cos\eta)-iqv\sin\eta}\right| < 1$$

We, therefore, conclude that the approximate factorization method does preserve the unconditional stability of our finite differencing strategy and that our numerical scheme for the advection equation is unconditionally stable.

2.2 As a guide to the difference scheme at the interior points of the computational domain, we consider the wave equation

$$\phi_{tt} = \phi_{xx} + \phi_{yy}$$

Our finite differencing yields

$$D_{tt}\phi^n = (D_{xx} + D_{yy}) \frac{\phi^{n+1}+\phi^{n-1}}{2}$$

or

$$[1 - \frac{\Delta t^2}{2} (D_{xx}+D_{yy})]\phi N = [1 - \frac{\Delta t^2}{2}(D_{xx}+D_{yy})]\phi M + \Delta t^2 (D_{xx}+D_{yy})\phi^n$$

Substituting $\phi = e^{i(kt+mx+ny)}$ and letting $\omega = k\Delta t$, we have

$$(\cos \omega-1) = [p^2(\cos \xi- 1) + q^2(\cos \eta- 1)] \cos \omega$$

$$\cos \omega = \frac{1}{1 + p^2(1- \cos \xi)+ q^2(1- \cos \eta)}$$

As long as p,q, are real, ω is real for all ξ and η . This means that the finite difference approximation is unconditionally stable .

Next, our approximate factorization preserves this property and permits us to solve the large algebraic system easily. Indeed, if we write

$$[1 - \frac{\Delta t^2}{2} D_{xx}][1 - \frac{\Delta t^2}{2} D_{yy}]\phi N$$

$$= [1 - \frac{\Delta t^2}{2} D_{xx}][1 - \frac{\Delta t^2}{2} D_{yy}]\phi M + \Delta t^2 (D_{xx}+D_{yy})\phi^n$$

Let $\phi = e^{i(kt+mx+ny)}$, we have

$$\cos\omega = \frac{1+p^2q^2(1-\cos\xi)(1-\cos\eta)}{1+p^2(1-\cos\xi)+q^2(1-\cos\eta)+p^2q^2(1-\cos\xi)(1-\cos\eta)}$$

For all real p and q, ω is real if ξ and η are real. In other words, t is real whenever x and y are real. This means that the scheme is unconditionally stable.

Finally, we remark that the scheme has no time step Δt restriction based on a linear stability analysis. However, in actual computation, an instability can be generated by the motion of shocks across which the differencing switches from upwind to central. To prevent this instability from occurring, it has been found in practice that the time step Δt must be chosen small enough that such shocks do not move a distance greater than one spatial grid point per time step. This restriction is necessary to maintain time accuracy anyway, and it is much less severe than the time step Δt restrictions associated with explicit methods.

V. COMPUTATIONAL RESULTS

Our computer code UFLO5 consists of steady and unsteady modes. The steady mode is the standard line relaxation scheme for the steady equation. We use it to generate a good initial guess for the unsteady mode. In fact, any steady potential flow solver can be used to replace this steady routine. The unsteady mode can also be used to compute the steady solution. In this section we first check the unsteady mode by calculating some steady solutions. Then we present some computational results for conventional and supercritical wing sections in rigid body motion.

1. Steady Calculations

As a test case, steady state calculations for the NACA0012 airfoil at Mach number $M_\infty = 0.79$ and angle of attack $\alpha = 0°$ are performed by the standard line relaxation method for the steady equation and by the unsteady scheme. The two modes produce virtually identical results. The time step size for the unsteady mode in this calculation is set to $\Delta t = 10\Delta x$ which is much larger than the time step allowed by the CFL condition for the explicit method. For a coarse mesh of 32×8 grid points, the time required to converge to the steady state using the unsteady mode is comparable to the steady mode.

The optimal location of the artificial boundary is problem dependent. If the artificial boundary is moved too close to the airfoil, instability can occur. The computational domain shown in Figure 8c has also been used for the NACA0012 airfoil at $M_\infty = 0.79$ and $\alpha = 0°$. Note that the upstream boundary is about 1.5 chord lengths from the nose as compared to a distance of about 10.5 chordlengths used for the above example. The ratio $\Delta t/\Delta x$ is given the value 10 as above. The correction is observed to decrease much more slowy in this case. Since the grid system is stretched in the code, the reduction in the computational mesh is not linearly proportional to the physical distance of the boundary from the airfoil. The benefit obtained by the reduced number of mesh points is overshadowed by the reduced numerical stability.

There is no difficulty in calculating flows with sonic flight speed. A Joukowski airfoil at $M_\infty = 1$ and $\alpha = 0°$ is chosen as an example, with the ratio $\Delta t/\Delta x$ set to 3.5 in this case. Usually, the numerical stability of the unsteady mode, in terms of the ratio $\Delta t/\Delta x$, decreases with either flight speed or angle of attack.

2. Unsteady Calculations

The NACA0012 airfoil and KORN airfoil (75-06-12) are chosen as prototypes for conventional and supercritical airfoils, respectively. Both airfoils have the same thickness to chord ratio. The rigid body motion of an airfoil can be described by three parameters: angle of attack, flight speed, and flight angle. We consider the flow past each airfoil when these parameters are varied separately.

2.1 Variation of flight speed

First, we consider the acceleration of the airfoil in the streamwise direction. That is, the airfoil moves with a sinusoidal variation in flight speed but with flight angle and angle of attack fixed.

In Figure 11 a,b,c, for the NACA0012 airfoil, as the flight speed increases (decreases), the supersonic region grows (shrinks) in size and the shock strengthens (weakens) and moves aft (fore). The shock wave displacement can also be observed in the pressure distributions in Figure 11 d.e.f., or from the position of peaks in the traces of pressure sensors on the upper surface of the airfoil in Figure 11 g . The peaks in those local pressure traces are produced as the shock wave passing by the pressure sensors. The nonsinusoidal trace curves demonstrate the nonlinear transonic effects caused by the shock wave displacement. The same calculations for the KORN airfoil appear in Figure 12. The unsteady loading distributions are shown in Figure 12 e,f, where peaks in the loading distributions are again due to shock waves. The loading is the difference of lower and upper pressure coefficients.

2.2 Variation of angle of attack

Next we consider the pitching airfoil which moves with a sinusoidal variation of the angle of attack, but with the flight velocity fixed. Small variations in angle of attack may lead to considerable changes in the pressure distribution, shock position and shock strength. As shown in Figure 9 a,b,c, for the NACA0012 airfoil, when the angle of attack increases (decreases), the supersonic region on the upper surface of the airfoil grows (shrinks) in size, and the shock wave strengthens (weakens) and moves aft (fore). In Figure 9j the nonsinusoidal trace of the pressure at location *6 on the airfoil surface clearly displays the shock wave movement. The unsteady pressure loading distributions are shown in Figure 9g,h,i.

The same calculations were performed for the KORN airfoil as shown in Figure 10. It is worthwhile noticing that the unsteady traces of the local pre-sure sensors for the KORN airfoil are much more nonlinear than those for the NACA0012 airfoil. This pattern is also observed in the loading distribution for the two airfoils. The fact that the shock excursion amplitude decreases with an increase in oscillatory frequency can be seen from the unsteady traces of the pressure sensor *6 in Figure 9 j,k,l.

2.3 Variation of flight angle

Finally, we consider changes in the airfoil's flight angle while keeping the angle of attack and flight speed fixed. The motion, for angle of attack $\alpha = 0$, of the airfoil is described in Figure 7.

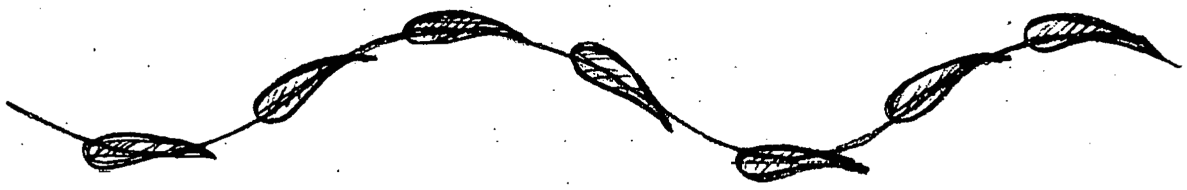


Figure 7. The Chords of Airfoils are Tangent to the Flight Trajectory.

The characteristics of this case are similar to those of the pitching airfoil. Computational results for the NACA0012 and KORN airfoils are shown in Figures 13 and 14 , respectively.

VI. CONCLUSION

1. Summary of the Work

A numerical method has been presented for determining the inviscid transonic flow past airfoils in rigid body motion. The method is based on the unsteady transonic potential flow equation in a computational domain designed for accurate application of the body surface boundary condition. A set of first order radiation boundary conditions are applied at the artificial computational boundaries located at a finite distance away from the airfoil surface. The finite difference approximations of the potential flow equation and radiation boundary conditions are constructed by using a type dependent differencing strategy. The leading truncation term provides the necessary dissipation to stabilize the scheme and to capture the shock waves automatically. The finite difference approximations are perturbed within the same order of accuracy in order to permit their factorzation into one-dimensional operators. Consequently, the problem can be solved efficiently by using a 5-diagonal matrix solver. The resulting algorithm is a time marching scheme without any iteration process in each time step.

Numerical experiments show that the scheme is very stable. The results presented in Section V demonstrate that the method is able to resolve the highly nonlinear transonic flow effects for flutter analysis of airfoils as long as the boundary layer remains attached.

2. Extension of the Technique

Our numerical method can be extended to three space dimensional problems. An important application is the unsteady transonic flow past wing-body combinations that model our airplane. The necessary geometric mapping techniques are available in analysis codes that compute steady flow past a wing-body combination [27]. Singularities associated with the geometric mapping would not be a serious problem and could be treated in a manner similar to the one used in the steady calculation. A further application could be the helicopter rotor in forward flight. Here the flow is unsteady because of the relative velocity of the advancing and retreating blades [8].

In principle, shock accuracy would be improved by shock fitting methods [45] or, alternatively, by the use of a difference scheme in conservative form. In practice, however, a turbulent boundary layer correction would be needed for more exact shock jump modeling [10, 13, 35].

REFERENCES

1. Ballhaus, W.F.; 1978. "Some Recent Progress in Transonic Flow Computations," Numerical Methods in Fluid Dynamics, ed. Wirz, H.J., Smolderen, J.J., pp. 155-235. New York: McGraw-Hill.
2. Ballhaus, W.F.; Goorjian, P.M.; 1977. "Implicit Finite Difference Computations of Unsteady Transonic Flows about Airfoils," AIAA J., Vol. 15, pp. 1728-1735.
3. Bauer, F.; Garabedian, P.R.; Korn, D.; Jameson, A.; 1975. Supercritical Wing Sections II, Lecture Notes in Economics and Mathematical Systems, Vol. 108, New York: Springer-Verlag.
4. Bayliss, A.; Turkel, E.; 1979. "Radiation Boundary Conditions for Wave-Like Equation," ICASE Report 79-26.
5. Beam, R.M.; Warming, R.F.; 1978. "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," AIAA J., Vol. 16, pp. 393-402.
6. Beam, R.M.; Warming, R.F.; 1979. "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations II: The Numerical ODE Connection," Proceedings of AIAA 4th Computational Fluid Dynamics Conference, pp. 1-13.
7. Bers, L.; 1958. Mathematical Aspects of Subsonic and Transonic Gas Dynamics. New York: John Wiley.
8. Caradonna, F.X.; Isom, M.P.; 1975. "Numerical Calculation of Unsteady Transonic Potential Flow over Helicopter Rotor Blades," AIAA paper 75-168.

9. Carlson, L.A.; 1976. "Transonic Airfoil Analysis and Design Using Cartesian Coordinates," J. of Aircraft, Vol. 13, pp. 349-356.

10. Chapman, D.R.; 1979. "Computational Aerodynamics Development and Outlook," AIAA J. Vol. 17, pp. 1293-1313.

11. Chipman, R.R.; 1980. "An Alternating Direction Implicit Algorithm for the Unsteady Potential Equation in Conservation Form," Grumman Aerospace Corp. Report.

12. Courant, R.; Friedrichs, K.O.; 1948. Supersonic Flows and Shock Waves, New York: Interscience-Wiley.

13. Deiwert, G.S.; Bailey, H.E.; 1978. "Prospects of Computing Airfoil Aerodynamics with Reynolds Averaged Navier-Stokes Codes," NASA cp 2045.

14. Eiseman, P.R.; 1980. "Geometric Methods in Computational Fluid Dynamics," ICASE Report No. 80-11.

15. Engquist, B.; Majda, A.; 1977. "Absorbing Boundary Conditions for the Numerical Simulations of Waves," Math. of Computation, Vol. 31, pp. 629-651.

16. Engquist, B.; Majda, A.; 1979. "Numerical Radiation Boundary Conditions for Unsteady Transonic Flow," preprint.

17. Fung, K.Y.; 1980. "Far Field Conditions for Unsteady Transonic Flows", preprint.

18. Garabedian, P.R.; 1976. "Computation of Wave Drag for Transonic Flow," J. D'Analyse Math., Vol. 30, pp. 164-171.

19. Grossman, B.; Volpe, G.; 1977. "An Analysis of the Inviscid Transonic Flow over Two Element Airfoil Systems," Grumman Aerospace Corp. Report.

20. Holst, T.L.; 1979. "A Fast Conservative Algorithm for Solving the Transonic Full-Potential Equation," Proceedings of AIAA 4th Computational Fluid Dynamics Conference, pp. 109-121.

21. Isogai, K.; 1977. "Calculation of Unsteady Transonic Flow over Oscillating Airfoils Using the Full Potential Equation," AIAA Paper No. 77-448.

22. Isogai, K.; 1978. "Numerical Study of Transonic Flow over Oscillating Airfoils Using the Full Potential Equation," NASA Tech. Pap. 1120.

23. Isaacson, E.; Keller, H.B.; 1966. Analysis of Numerical Methods. New York: Wiley.

24. Ives, D.C.; 1976. "A Modern Look at Conformal Mapping Including Multiple Connected Regions," AIAA J., Vol. 14, pp. 1006-1011.

25. Jameson, A.; 1978. "Transonic Flow Calculations," Numerical Methods in Fluid Dynamics, ed. Winz, H.J.; Smolderen, J.J., pp. 1-87. New York: McGraw-Hill.

26. Jameson, A.; 1979. "Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multiple Grid Method," Proceedings AIAA 4th Compotational Fluid Dynamics Conference, pp. 122-146.

27. Jameson, A.; Caughey, D.A.; 1980. "Progress in Finite Volume Calculations for Wing-Fuselage Combinations," AIAA J. pp. 1218-1288.

28. Kreiss, H-O.; 1977. "Numerical Methods for Solving Time Dependent Problems for Partial Differential Equations," Lecture Notes, Courant Institute.

29. Landahl, M.T.; 1976. "Some Development in Unsteady Transonic Flow Research," Symp. Transonicum II. ed., Oswatitsch, K. Berlin: Springer-Verlag.

30. Lax, P.D.; 1972. "Hyperbolic System of Conservation Laws and the Mathematical Theory of Shock Waves," SIAM Regional Conference in Applied Math.

31. Lax, P.D.; 1978. "The Numerical Solution of the Equations of Fluid Dynamics," Lectures on Combustion Theory, ed., Burstein, S.Z.; Lax, P.D.; Sod, G.A.; Courant Institute Report.

32. MacCormack, R.W.; Lomax, H.; 1979. "Numerical Solution of Compressible Viscous Flows," Ann. Rev. Fluid Mech., Vol. 11, pp. 289-316.

33. Magnus, R.J.; 1977. "Computational Research on Inviscid Unsteady Transonic Flow over Airfoils," ONRCASD/LVP 77-010.

34. McCorsky, W.J.; 1977. "Some Current Research in Unsteady Fluid Dynamics," Trans. ASME J. Fluids Eng., pp. 8-39.

35. Melnik, R.E.; Chow, R.; Mead, H.R.; 1977. "Theory of Viscous Transonic Flow over Airfoils at High Reynolds Number," AIAA Pap., 77-680.

36. Milne-Thompson, L.M.; 1968. Theoretical Hydrodynamics, 5th ed. New York: MacMillan.

37. Murman, E.M. and Cole, J.D.; 1971. "Calculation of Plane Steady Transonic Flows," AIAA J., Vol. 9, pp. 114-121.

38. Peyret, R.; Viviand, H.; 1975. "Computation of Viscous Compressible Flows Based on the Navier-Stokes Equations," AGARDograph No. 212

39. Richtmyer, R.D.; Morton, K.W.; 1967. Difference Methods for Initial Value Problems, 2nd ed. New York: Wiley-Interscience.

40. Richtmyer, R.D.; 1978. Principles of Advanced Mathematical Physics Vol. 1, New York: Springer-Verlag.

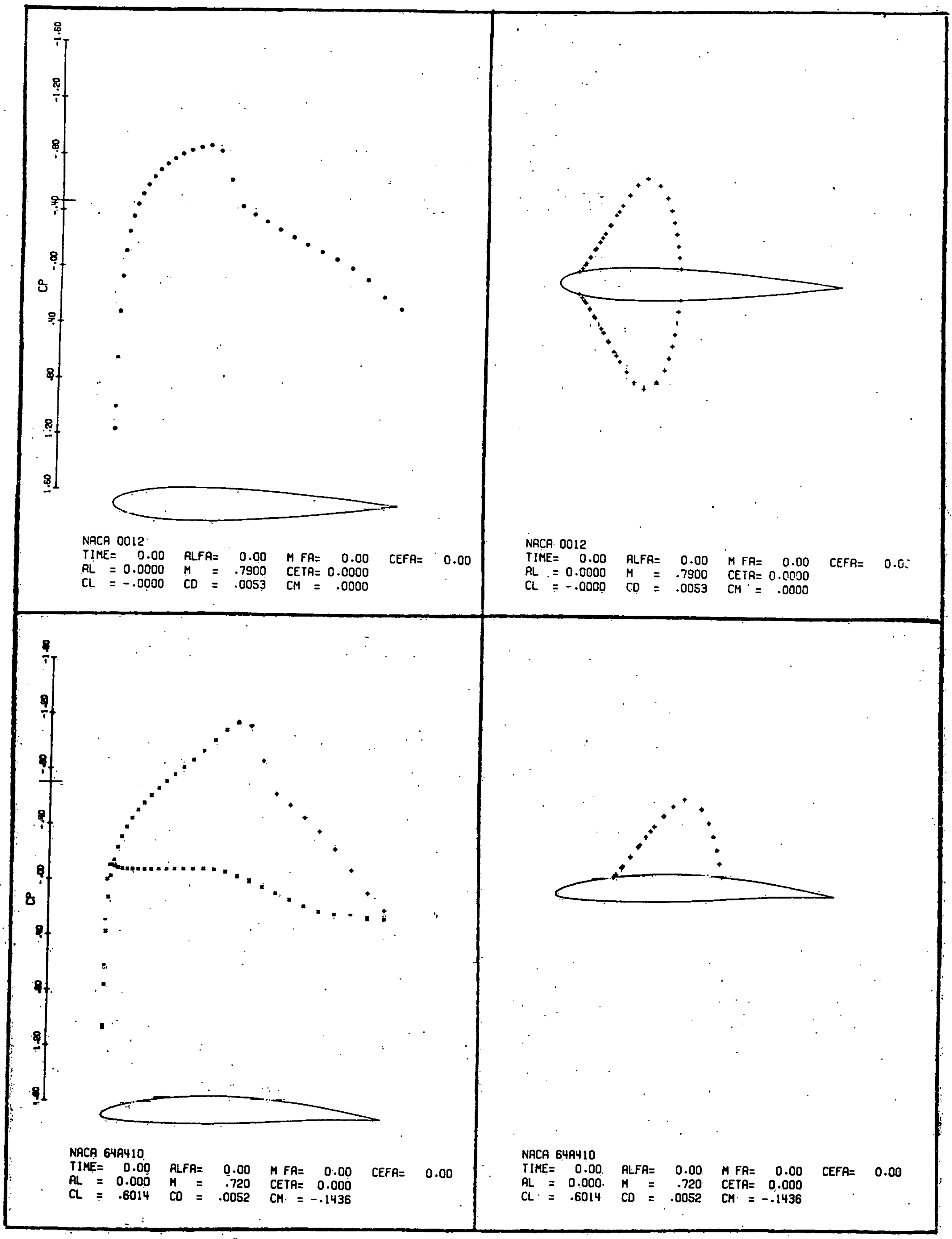
41. Roache, P.J.; 1976. Computational Fluid Dynamics, revised ed., N.M.: Hermosa Publishers.

42. Steger, J.L.; 1977. "Implicit Finite Difference Simulation of Flow about Arbitrary Geometries with Application to Airfoils," AIAA Paper No. 77-665.

43. Thompson, J.F.; Thames, F.C.; Mastin, C.W.; 1974. "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinates System for Field Containing Any Number of Arbitrary Two-dimensional Bodies," J. Comp. Physics, Vol. 15, pp. 299-319.

44. Tijdeman, H.; Seebass, R.; 1980. "Transonic Flow Past Oscillating Airfoils," Ann. Rev. Fluid Mech., Vol. 12, pp. 181-222.

45. Yu, N.J.; Seebass, A.R.; Ballhaus, W.F.; 1978. "Implicit Shock Fitting Scheme for Unsteady Transonic Flow Computations," AIAA J., Vol. 16, pp. 673-678.



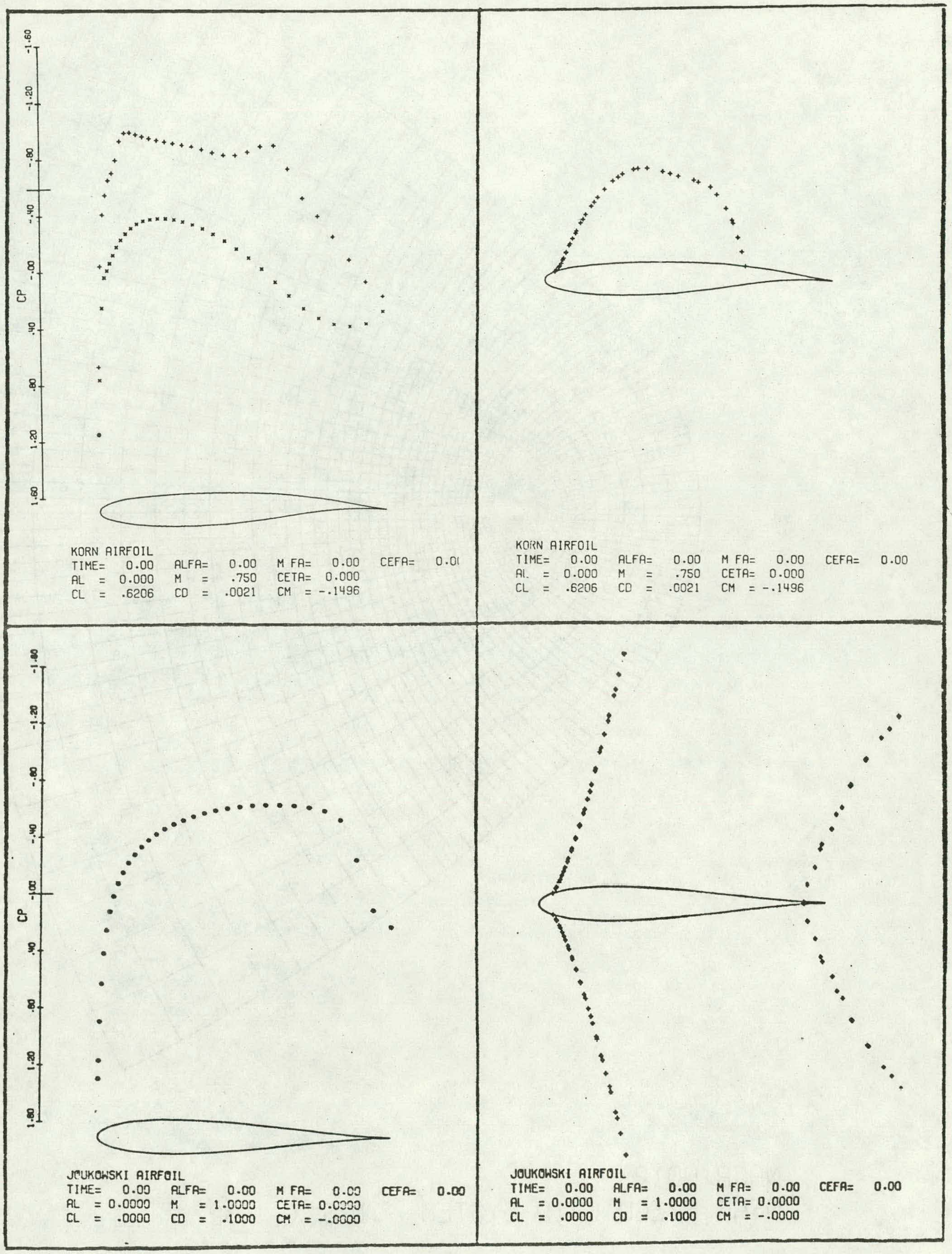
```
NACA 0012
TIME=   0.00   ALFA=   0.00   M FA=   0.00   CEFA=   0.00
AL  = 0.0000   M   =  .7900   CETA= 0.0000
CL  = -.0000   CD  =  .0053   CM  =  .0000
```

```
NACA 0012
TIME=   0.00   ALFA=   0.00   M FA=   0.00   CEFA=   0.0
AL  = 0.0000   M   =  .7900   CETA= 0.0000
CL  = -.0000   CD  =  .0053   CM  =  .0000
```

```
NACA 64A410
TIME=   0.00   ALFA=   0.00   M FA=   0.00   CEFA=   0.00
AL  =  0.000   M   =   .720   CETA=  0.000
CL  =  .6014   CD  =  .0052   CM  = -.1436
```

```
NACA 64A410
TIME=   0.00   ALFA=   0.00   M FA=   0.00   CEFA=   0.00
AL  =  0.000   M   =   .720   CETA=  0.000
CL  =  .6014   CD  =  .0052   CM  = -.1436
```

Figure 8a



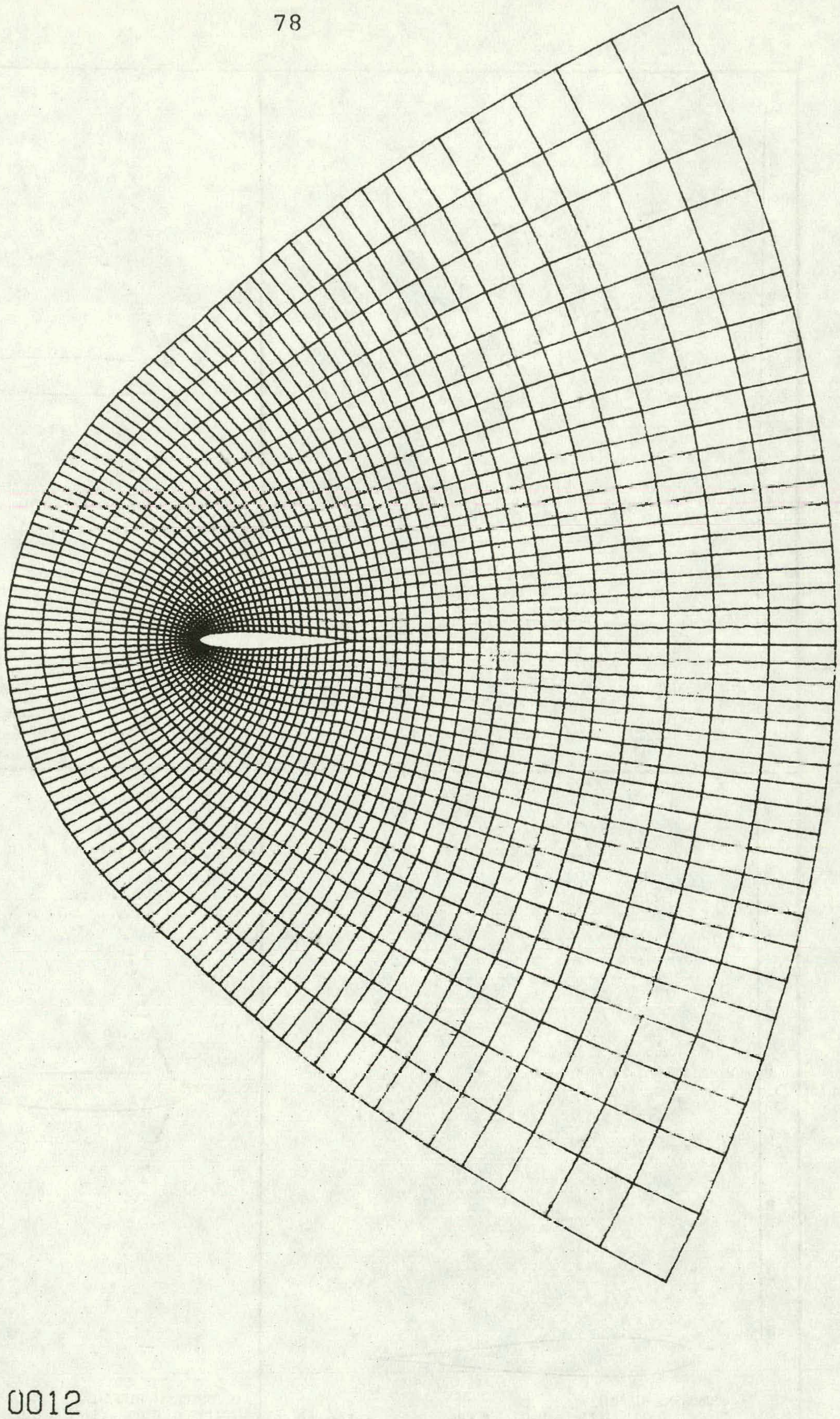
```
KORN AIRFOIL
TIME=   0.00   ALFA=   0.00   M FA=   0.00   CEFA=   0.0(
AL  =  0.000   M   =   .750   CETA=  0.000
CL  =  .6206   CD  =  .0021   CM  = -.1496
```

```
KORN AIRFOIL
TIME=   0.00   ALFA=   0.00   M FA=   0.00   CEFA=   0.00
AL  =  0.000   M   =   .750   CETA=  0.000
CL  =  .6206   CD  =  .0021   CM  = -.1496
```

```
JOUKOWSKI AIRFOIL
TIME=   0.00   ALFA=   0.00   M FA=   0.00   CEFA=   0.00
AL  = 0.0000   M   = 1.0000   CETA= 0.0000
CL  =  .0000   CD  =  .1000   CM  = -.0000
```

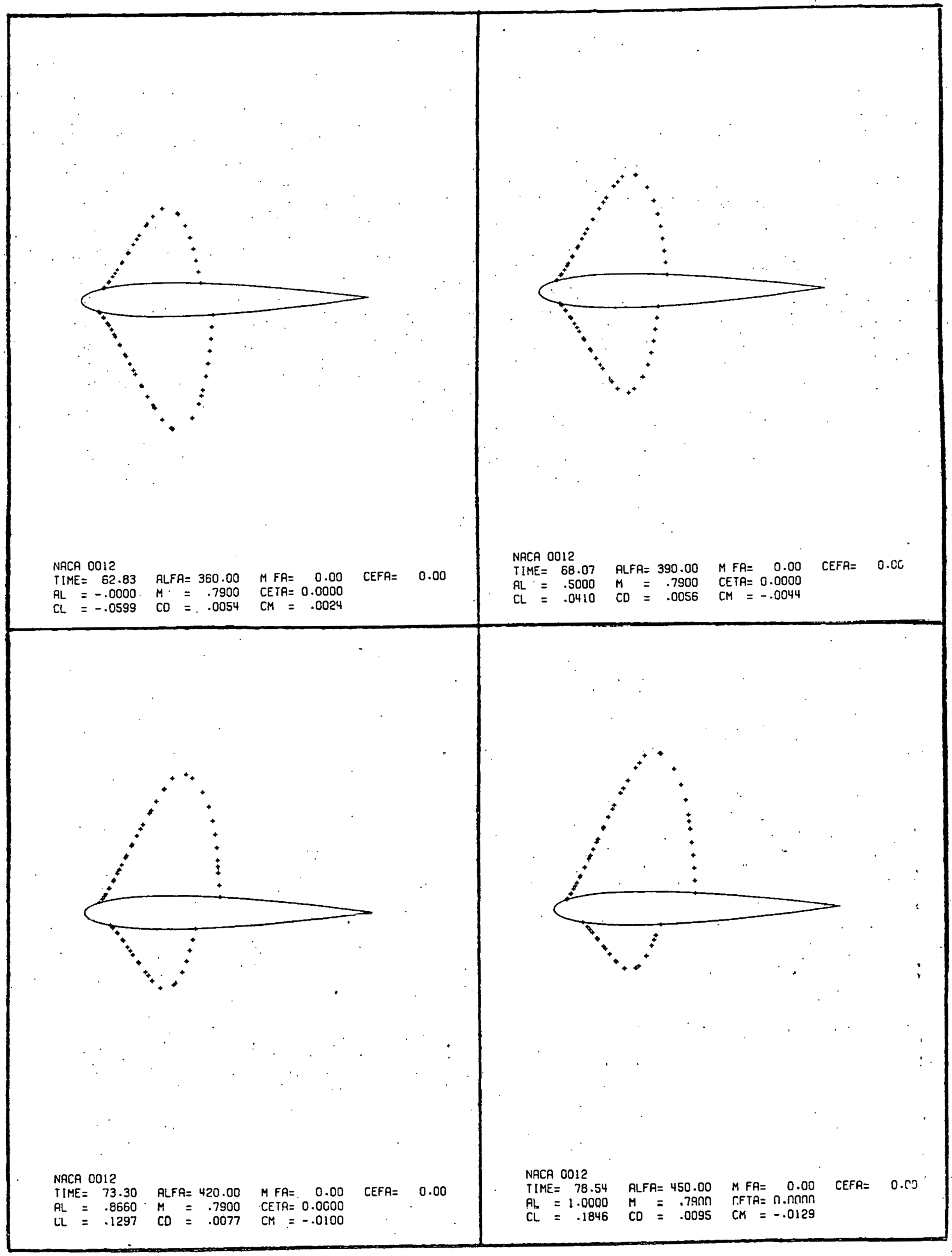
```
JOUKOWSKI AIRFOIL
TIME=   0.00   ALFA=   0.00   M FA=   0.00   CEFA=   0.00
AL  = 0.0000   M   = 1.0000   CETA= 0.0000
CL  =  .0000   CD  =  .1000   CM  = -.0000
```

Figure 8b



NACA 0012
NEAR FIELD GRID SYSTEM   128 X   32

Figure 8c



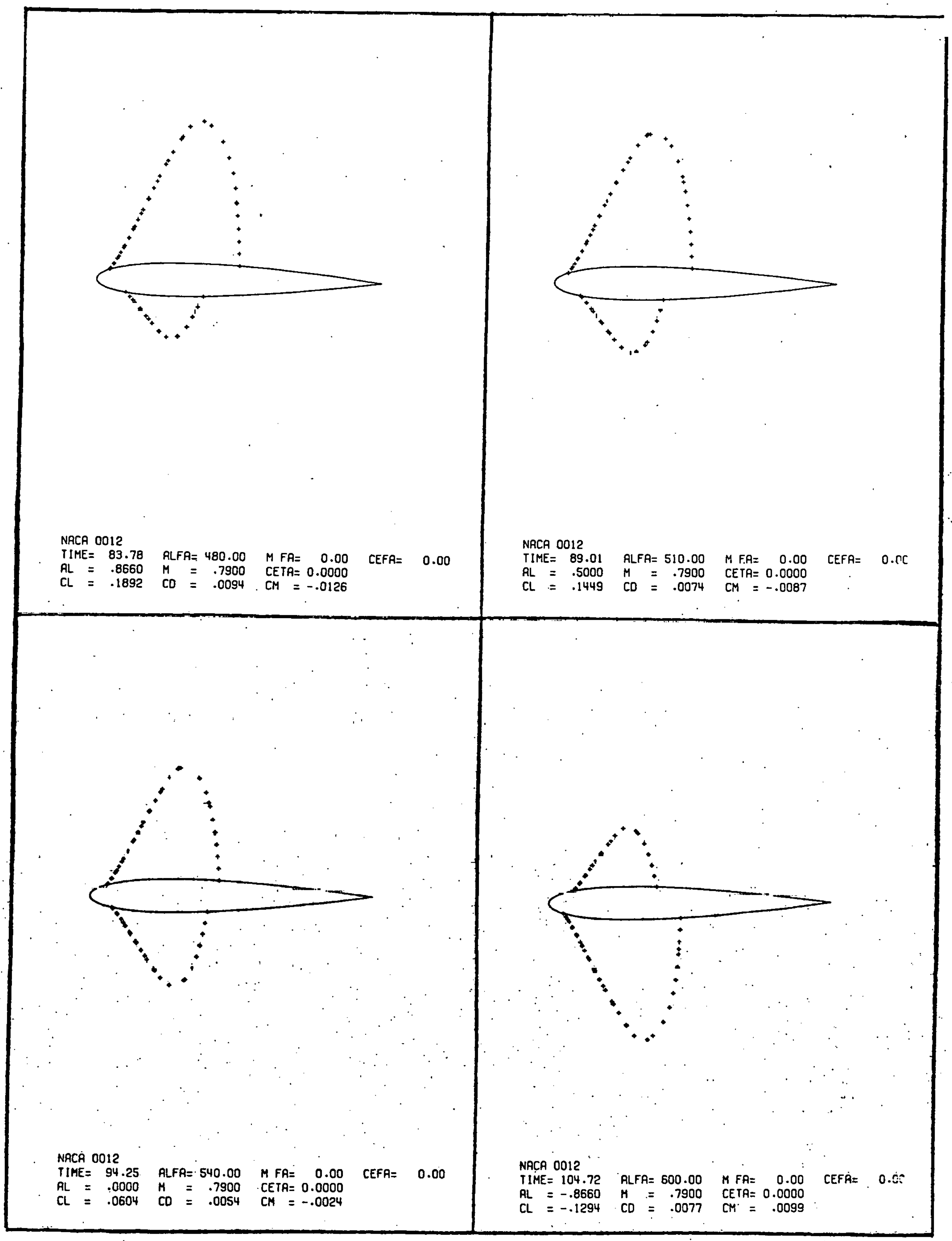
```
NACA 0012
TIME=  62.83   ALFA= 360.00   M FA=   0.00   CEFA=    0.00
AL   = -.0000  M    =  .7900  CETA= 0.0000
CL   = -.0599  CD   =  .0054  CM   =  .0024
```

```
NACA 0012
TIME=  68.07   ALFA= 390.00   M FA=   0.00   CEFA=    0.00
AL   =  .5000  M    =  .7900  CETA= 0.0000
CL   =  .0410  CD   =  .0056  CM   = -.0044
```

```
NACA 0012
TIME=  73.30   ALFA= 420.00   M FA=   0.00   CEFA=    0.00
AL   =  .8660  M    =  .7900  CETA= 0.0000
CL   =  .1297  CD   =  .0077  CM   = -.0100
```

```
NACA 0012
TIME=  78.54   ALFA= 450.00   M FA=   0.00   CEFA=    0.00
AL   = 1.0000  M    =  .7900  CETA= 0.0000
CL   =  .1846  CD   =  .0095  CM   = -.0129
```

Fiqure 9a



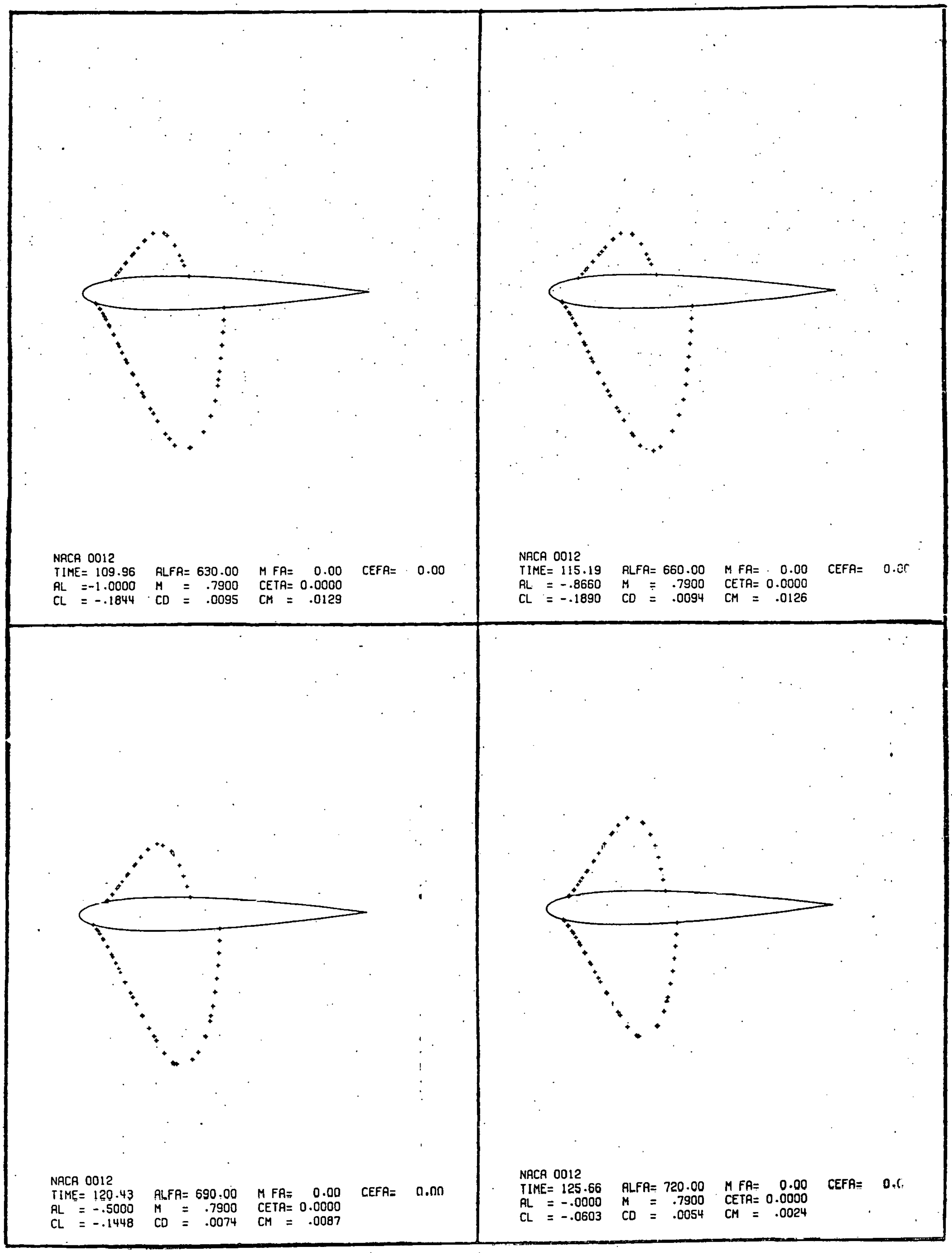
```
NACA 0012
TIME=  83.78   ALFA= 480.00   M FA=   0.00   CEFA=    0.00
AL   =  .8660  M    =  .7900  CETA= 0.0000
CL   =  .1892  CD   =  .0094  CM   = -.0126
```

```
NACA 0012
TIME=  89.01   ALFA= 510.00   M FA=   0.00   CEFA=    0.00
AL   =  .5000  M    =  .7900  CETA= 0.0000
CL   =  .1449  CD   =  .0074  CM   = -.0087
```

```
NACA 0012
TIME=  94.25   ALFA= 540.00   M FA=   0.00   CEFA=    0.00
AL   =  .0000  M    =  .7900  CETA= 0.0000
CL   =  .0604  CD   =  .0054  CM   = -.0024
```

```
NACA 0012
TIME= 104.72   ALFA= 600.00   M FA=   0.00   CEFA=    0.00
AL   = -.8660  M    =  .7900  CETA= 0.0000
CL   = -.1294  CD   =  .0077  CM   =  .0099
```

Figure 9b



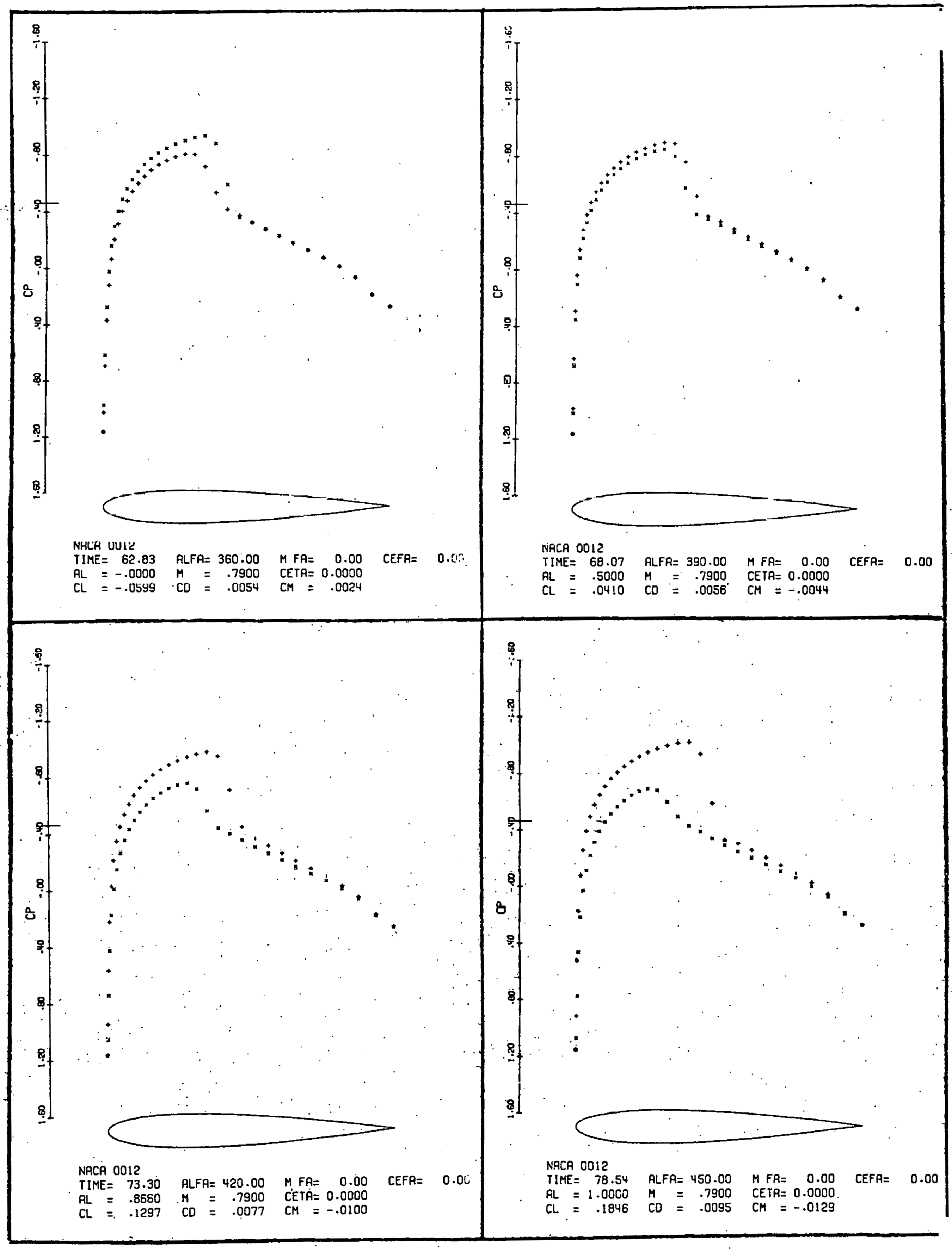
```
NACA 0012
TIME= 109.96   ALFA= 630.00   M FA=   0.00   CEFA=   0.00
AL  =-1.0000   M   =  .7900   CETA= 0.0000
CL  = -.1844   CD  =  .0095   CM  =  .0129
```

```
NACA 0012
TIME= 115.19   ALFA= 660.00   M FA=   0.00   CEFA=   0.0
AL  = -.8660   M   =  .7900   CETA= 0.0000
CL  = -.1890   CD  =  .0094   CM  =  .0126
```

```
NACA 0012
TIME= 120.43   ALFA= 690.00   M FA=   0.00   CEFA=   0.00
AL  = -.5000   M   =  .7900   CETA= 0.0000
CL  = -.1448   CD  =  .0074   CM  =  .0087
```

```
NACA 0012
TIME= 125.66   ALFA= 720.00   M FA=   0.00   CEFA=   0.(
AL  = -.0000   M   =  .7900   CETA= 0.0000
CL  = -.0603   CD  =  .0054   CM  =  .0024
```

Figure 9c



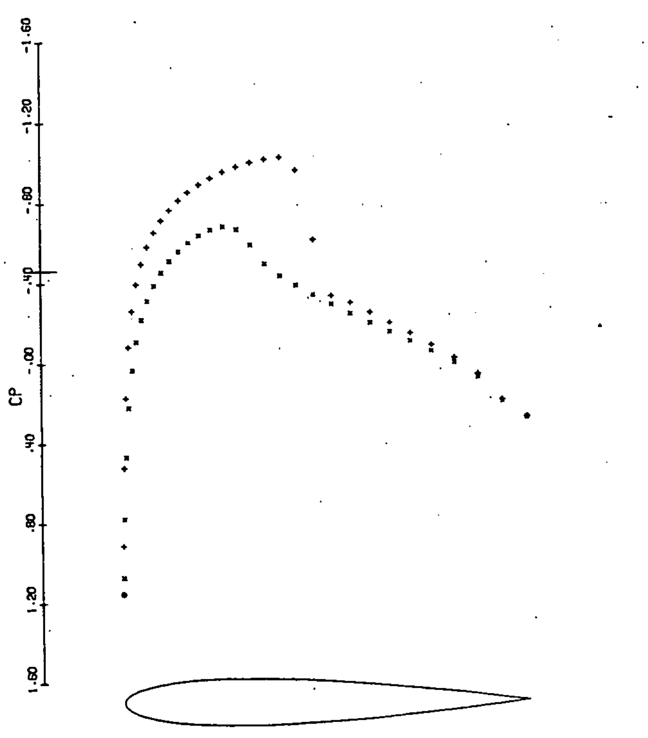
```
NACA 0012
TIME=  62.83   ALFA= 360.00   M FA=   0.00   CEFA=   0.00
AL   = -.0000  M    =  .7900  CETA= 0.0000
CL   = -.0599  CD   =  .0054  CM   =  .0024
```

```
NACA 0012
TIME=  68.07   ALFA= 390.00   M FA=   0.00   CEFA=   0.00
AL   =  .5000  M    =  .7900  CETA= 0.0000
CL   =  .0410  CD   =  .0056  CM   = -.0044
```

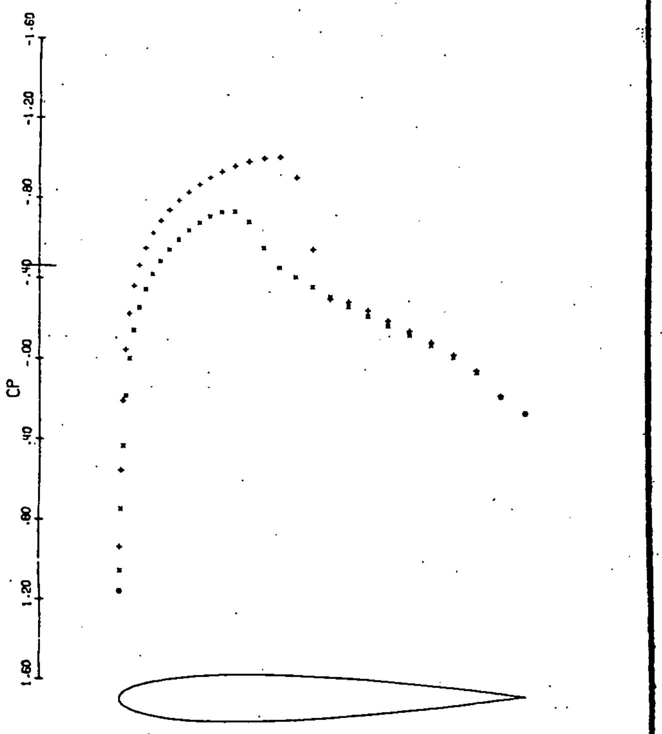
```
NACA 0012
TIME=  73.30   ALFA= 420.00   M FA=   0.00   CEFA=   0.00
AL   =  .8660  M    =  .7900  CETA= 0.0000
CL   =  .1297  CD   =  .0077  CM   = -.0100
```

```
NACA 0012
TIME=  78.54   ALFA= 450.00   M FA=   0.00   CEFA=   0.00
AL   = 1.0000  M    =  .7900  CETA= 0.0000
CL   =  .1846  CD   =  .0095  CM   = -.0129
```

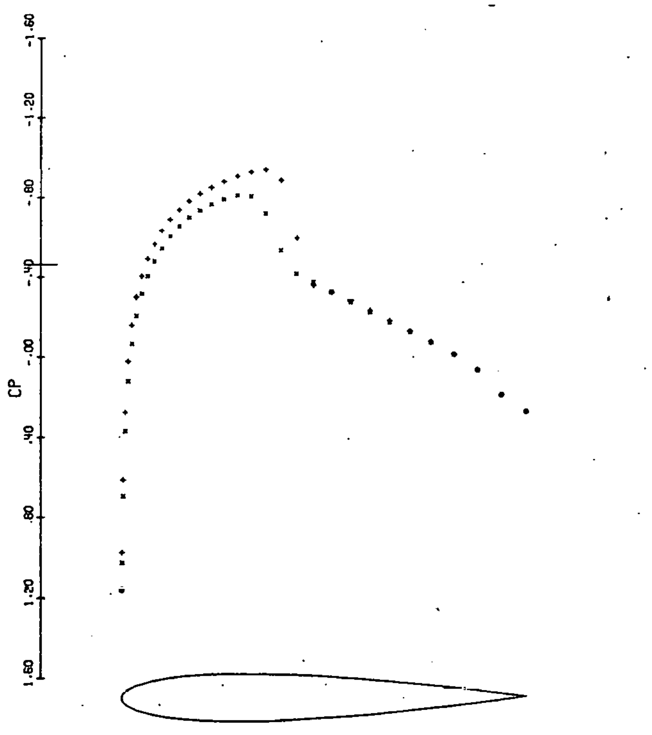
Figure 9d



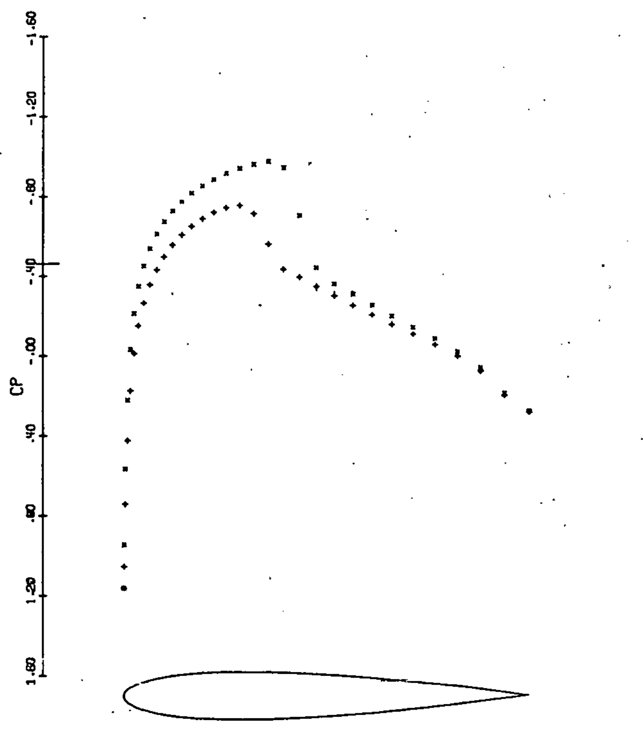
```
NACA 0012
TIME=  83.78    ALFA= 480.00    M FA=    0.00    CEFA=    0.0
AL   =  .8660   M    =  .7900   CETA= 0.0000
CL   =  .1892   CD   =  .0094   CM   = -.0126
```



```
NACA 0012
TIME=  89.01    ALFA= 510.00    M FA=    0.00    CEFA=    0.00
AL   =  .5000   M    =  .7900   CETA= 0.0000
CL   =  .1449   CD   =  .0074   CM   = -.0087
```

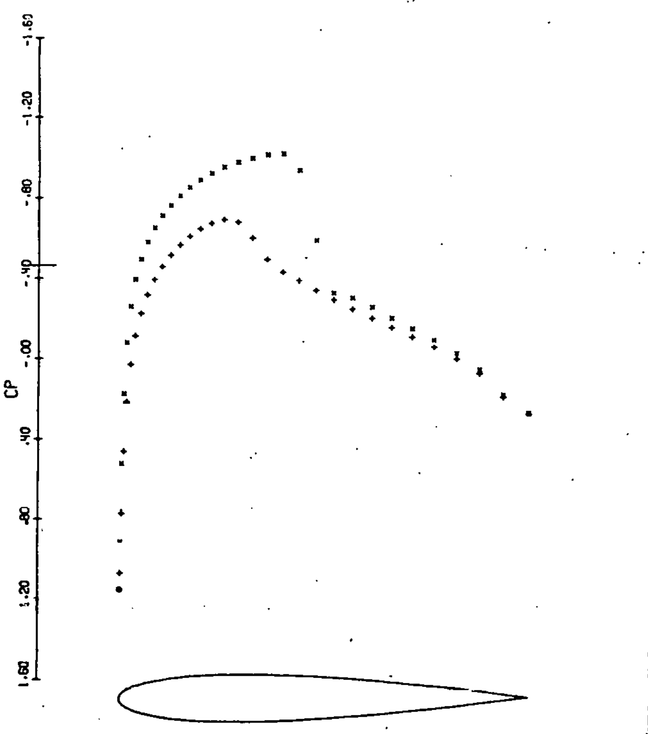


```
NACA 0012
TIME=  94.25    ALFA= 540.00    M FA=    0.00    CEFA=    0.00
AL   =  .0000   M    =  .7900   CETA= 0.0000
CL   =  .0604   CD   =  .0054   CM   = -.0024
```

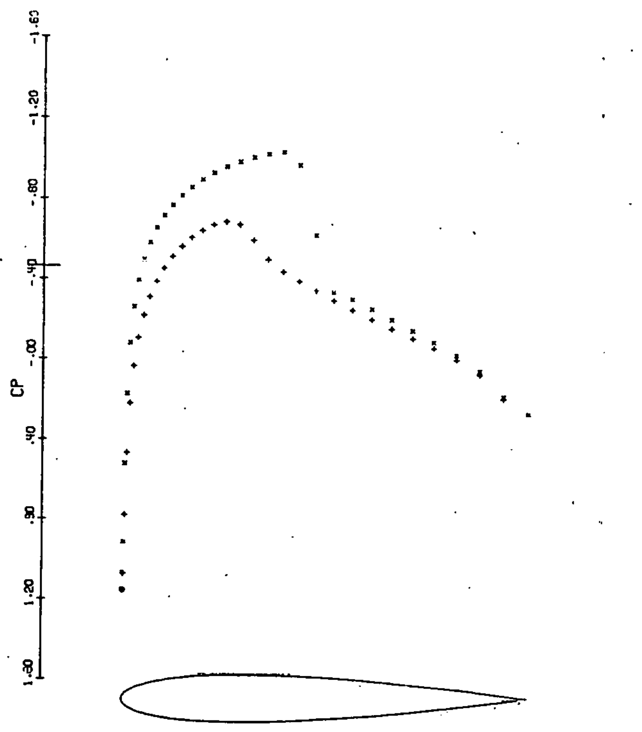


```
NACA 0012
TIME= 104.72    ALFA= 600.00    M FA=    0.00    CEFA=    0.00
AL   = -.8660   M    =  .7900   CETA= 0.0000
CL   = -.1294   CD   =  .0077   CM   =  .0099
```

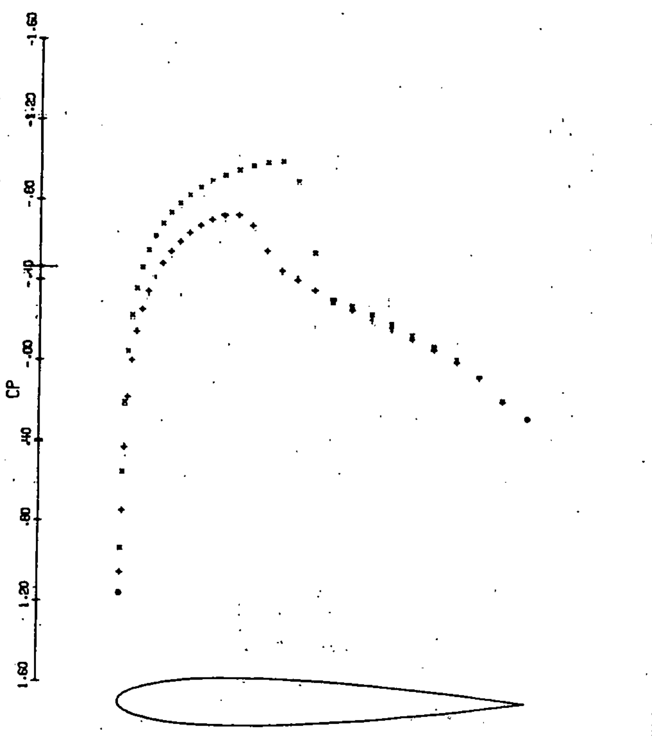
Figure 9e



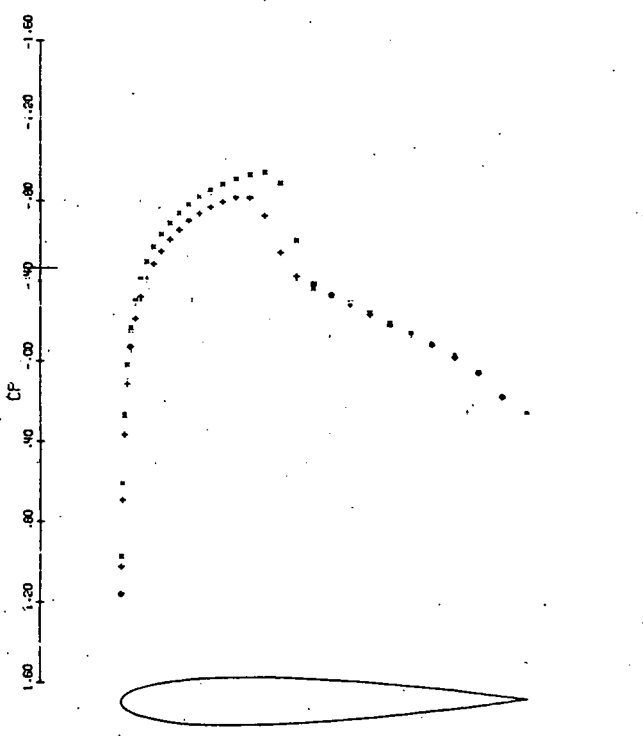
```
NACA 0012
TIME= 109.96   ALFA= 630.00   M FA=   0.00   CEFA=   0.00
AL   =-1.0000  M    =  .7900  CETA= 0.0000
CL   = -.1844  CD   =  .0095  CM   =  .0129
```



```
NACA 0012
TIME= 115.19   ALFA= 660.00   M FA=   0.00   CEFA=   0.00
AL   = -.8660  M    =  .7900  CETA= 0.0000
CL   = -.1890  CD   =  .0094  CM   =  .0126
```

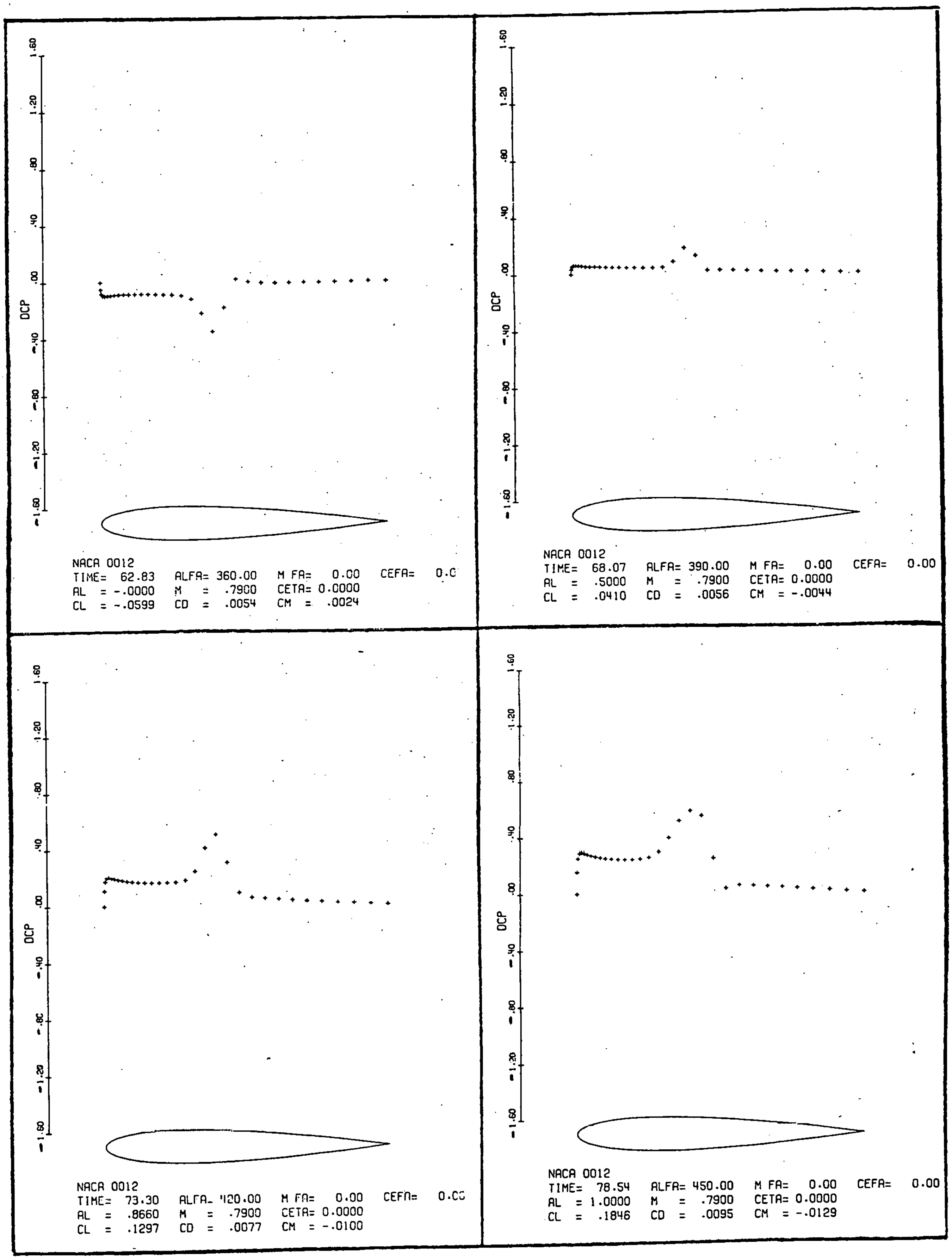


```
NACA 0012
TIME= 120.43   ALFA= 690.00   M FA=   0.00   CEFA=   0.00
AL   = -.5000  M    =  .7900  CETA= 0.0000
CL   = -.1448  CD   =  .0074  CM   =  .0087
```



```
NACA 0012
TIME= 125.66   ALFA= 720.00   M FA=   0.00   CEFA=   0.00
AL   = -.0000  M    =  .7900  CETA= 0.0000
CL   = -.0603  CD   =  .0054  CM   =  .0024
```

Figure 9f



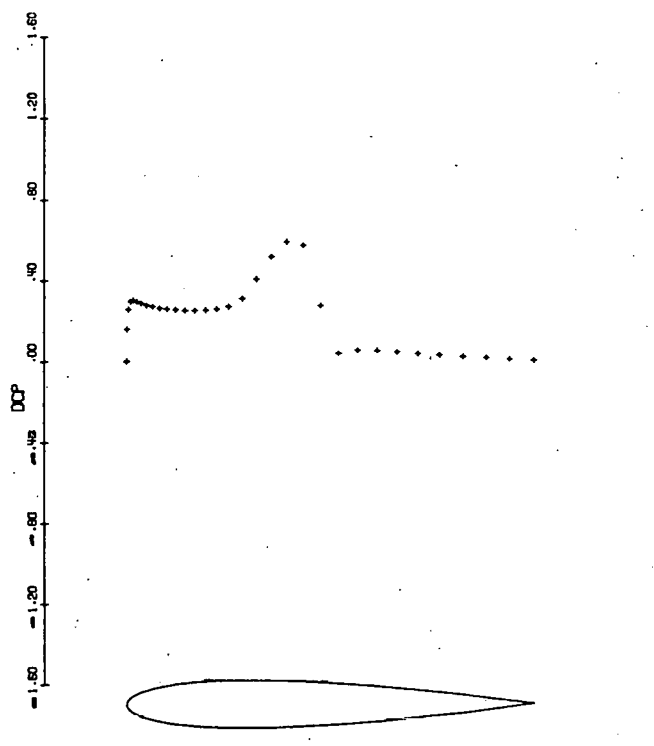
```
NACA 0012
TIME=  62.83   ALFA= 360.00   M FA=   0.00   CEFA=   0.C
AL   = -.0000  M    =  .7900  CETA= 0.0000
CL   = -.0599  CD   =  .0054  CM   =  .0024
```

```
NACA 0012
TIME=  68.07   ALFA= 390.00   M FA=   0.00   CEFA=   0.00
AL   =  .5000  M    =  .7900  CETA= 0.0000
CL   =  .0410  CD   =  .0056  CM   = -.0044
```

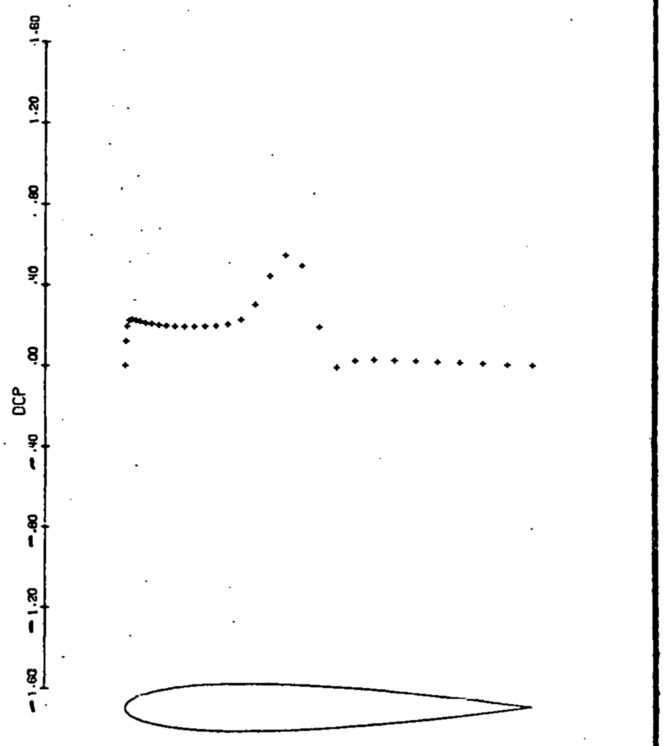
```
NACA 0012
TIME=  73.30   ALFA- 420.00   M FA=   0.00   CEFA=   0.CC
AL   =  .8660  M    =  .7900  CETA= 0.0000
CL   =  .1297  CD   =  .0077  CM   = -.0100
```

```
NACA 0012
TIME=  78.54   ALFA= 450.00   M FA=   0.00   CEFA=   0.00
AL   = 1.0000  M    =  .7900  CETA= 0.0000
CL   =  .1846  CD   =  .0095  CM   = -.0129
```

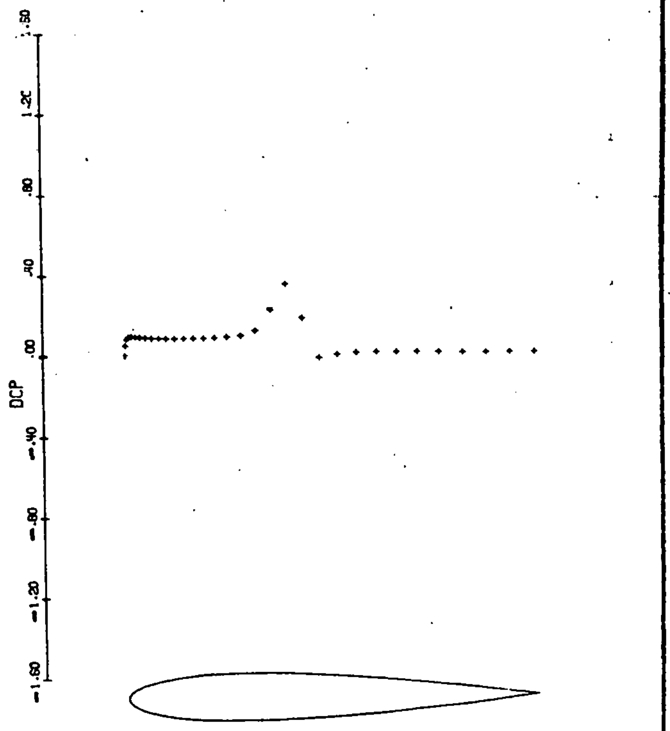
Figure 9g



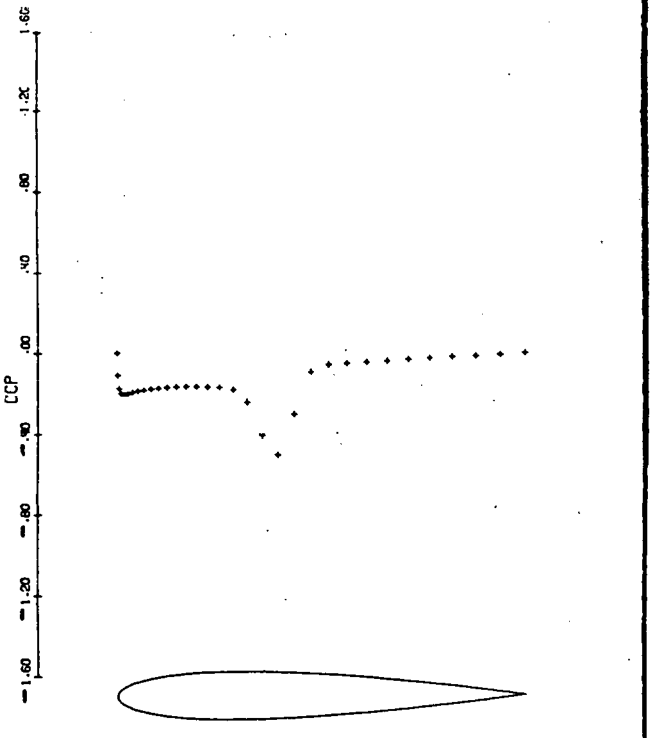
```
NACA 0012
TIME=  83.78   ALFA= 480.00   M FA=   0.00   CEFA=   0.00
AL   =  .8660   M    =  .7900  CETA= 0.0000
CL   =  .1892   CD   =  .0094  CM   = -.0126
```



```
NACA 0012
TIME=  89.01   ALFA= 510.00   M FA=   0.00   CEFA=   0.00
AL   =  .5000   M    =  .7900  CETA= 0.0000
CL   =  .1449   CD   =  .0074  CM   = -.0087
```

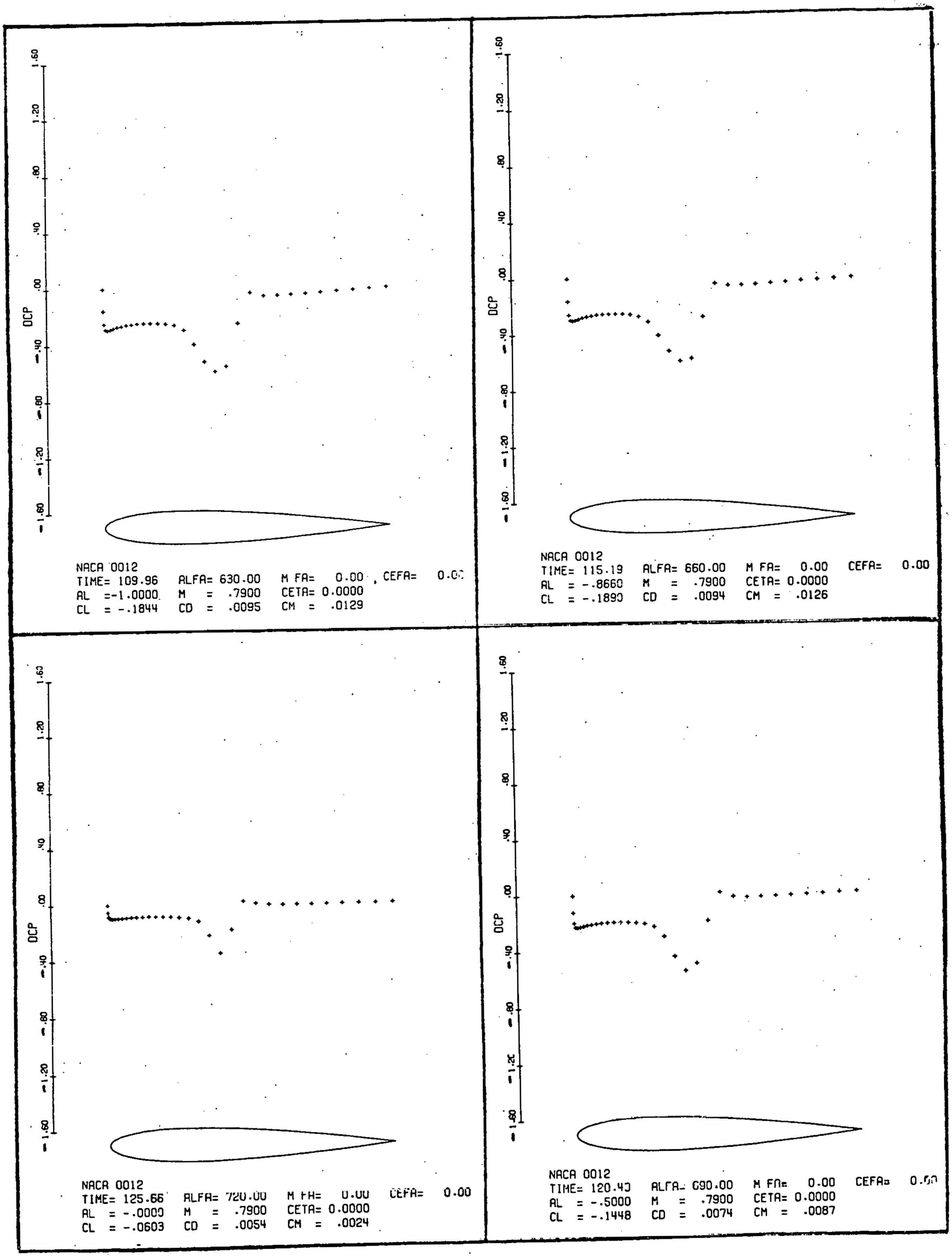


```
NACA 0012
TIME=  94.25   ALFA= 540.00   M FA=   0.00   CEFA=   0.00
AL   =  .0000   M    =  .7900  CETA= 0.0000
CL   =  .0604   CD   =  .0054  CM   = -.0024
```



```
NACA 0012
TIME= 104.72   ALFA= 600.00   M FA=   0.00   CEFA=   0.00
AL   = -.8660   M    =  .7900  CETA= 0.0000
CL   = -.1294   CD   =  .0077  CM   =  .0099
```

Figure 9h



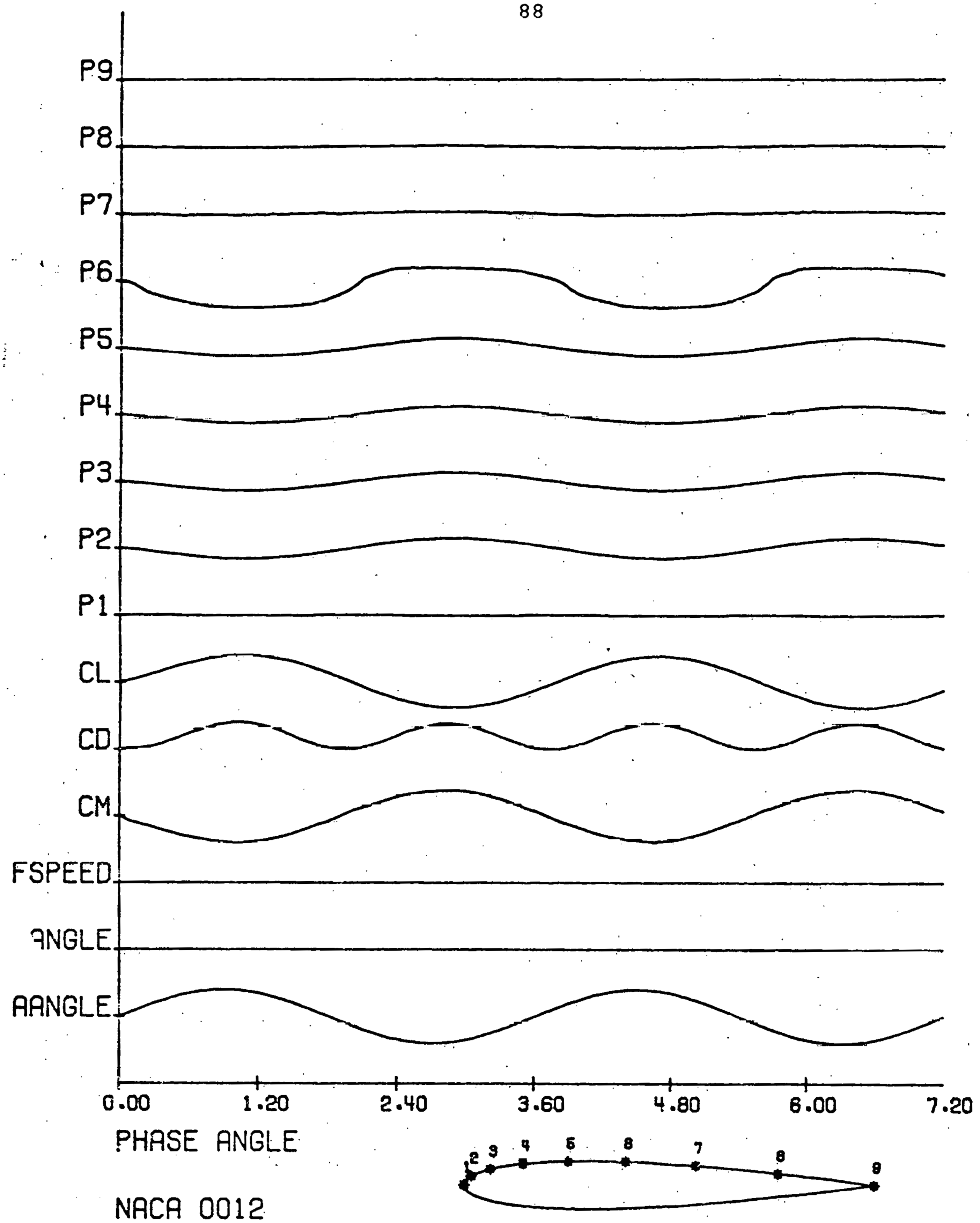
```
NACA 0012
TIME= 109.96   ALFA= 630.00   M FA=   0.00  , CEFA=   0.00
AL  =-1.0000   M   =  .7900   CETA= 0.0000
CL  = -.1844   CD  =  .0095   CM  =  .0129
```

```
NACA 0012
TIME= 115.19   ALFA= 660.00   M FA=   0.00   CEFA=   0.00
AL  = -.8660   M   =  .7900   CETA= 0.0000
CL  = -.1890   CD  =  .0094   CM  =  .0126
```

```
NACA 0012
TIME= 125.66   ALFA= 720.00   M FA=   0.00   CEFA=   0.00
AL  = -.0000   M   =  .7900   CETA= 0.0000
CL  = -.0603   CD  =  .0054   CM  =  .0024
```

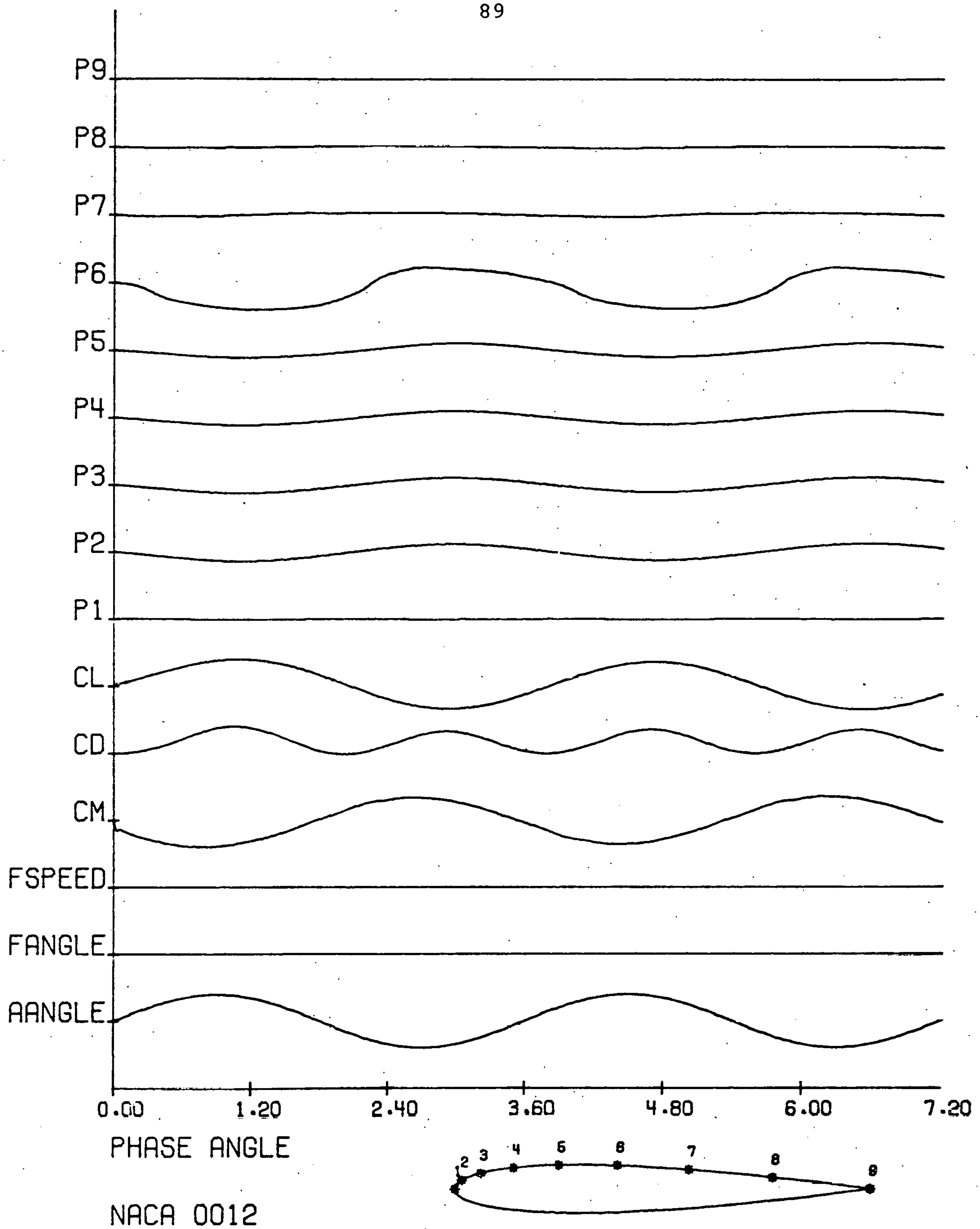
```
NACA 0012
TIME= 120.43   ALFA= 690.00   M FA=   0.00   CEFA=   0.00
AL  = -.5000   M   =  .7900   CETA= 0.0000
CL  = -.1448   CD  =  .0074   CM  =  .0087
```

Figure 9i



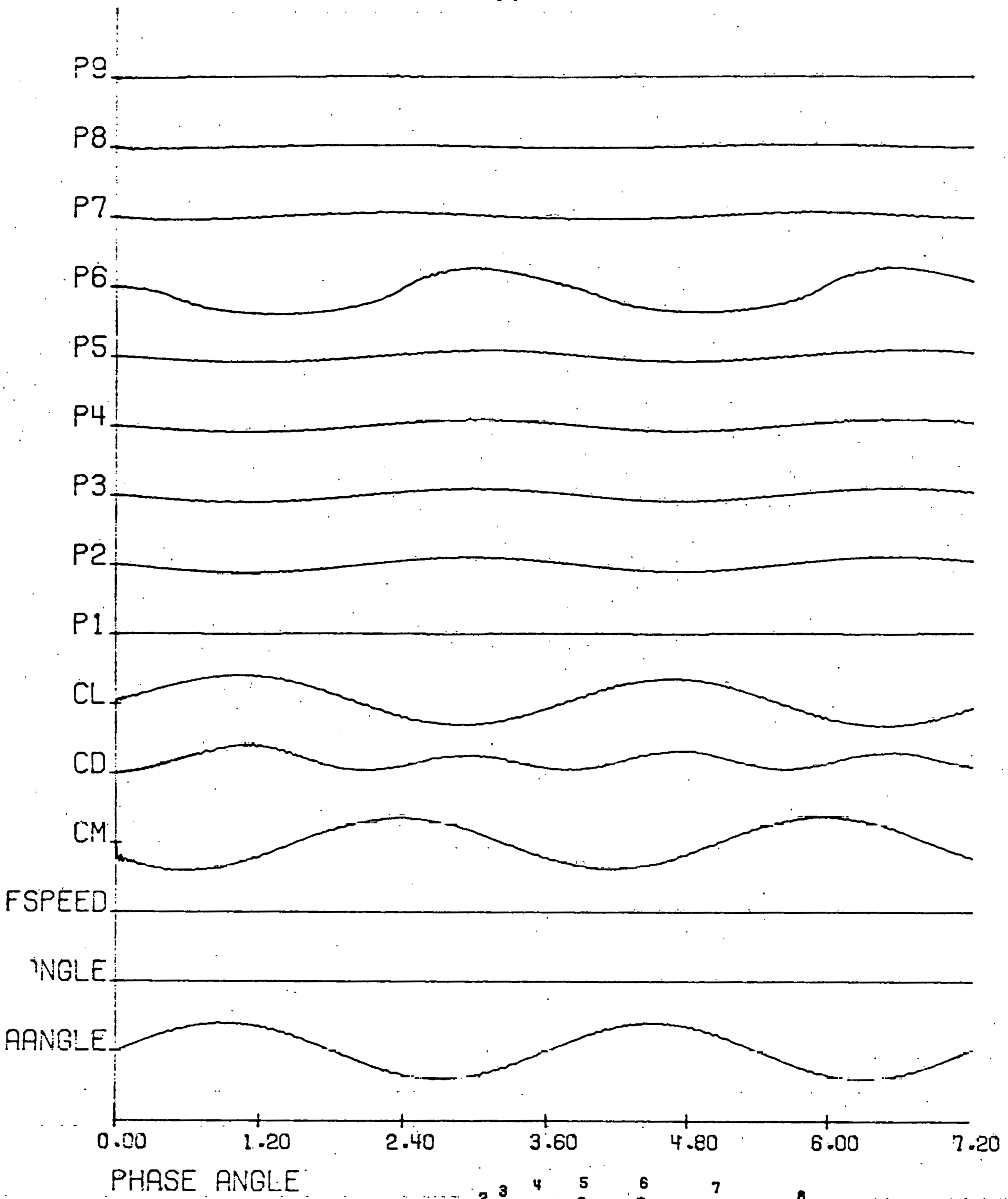
NACA 0012
UNSTEADY TRACES OF AIRFOIL IN SINUSOIDAL RIGID BODY MOTION
MEAN ATTACK ANGLE= 0.00 AMP= 1.00 FREQ RATE= .10
MEAN FLIGHT SPEED= .79 AMP= 0.00 FREQ RATE= 0.
MEAN FLIGHT ANGLE= 0.00 AMP= 0.00 FREQ RATE= 0.00

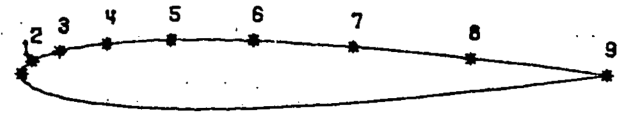
Figure 9j



NACA 0012
UNSTEADY TRACES OF AIRFOIL IN SINUSOIDAL RIGID BODY MOTI
MEAN ATTACK ANGLE= 0.00 AMP= 1.00 FREQ RATE= .3
MEAN FLIGHT SPEED= .79 AMP= 0.00 FREQ RATE= 0.0
MEAN FLIGHT ANGLE= 0.00 AMP= 0.00 FREQ RATE= 0.0

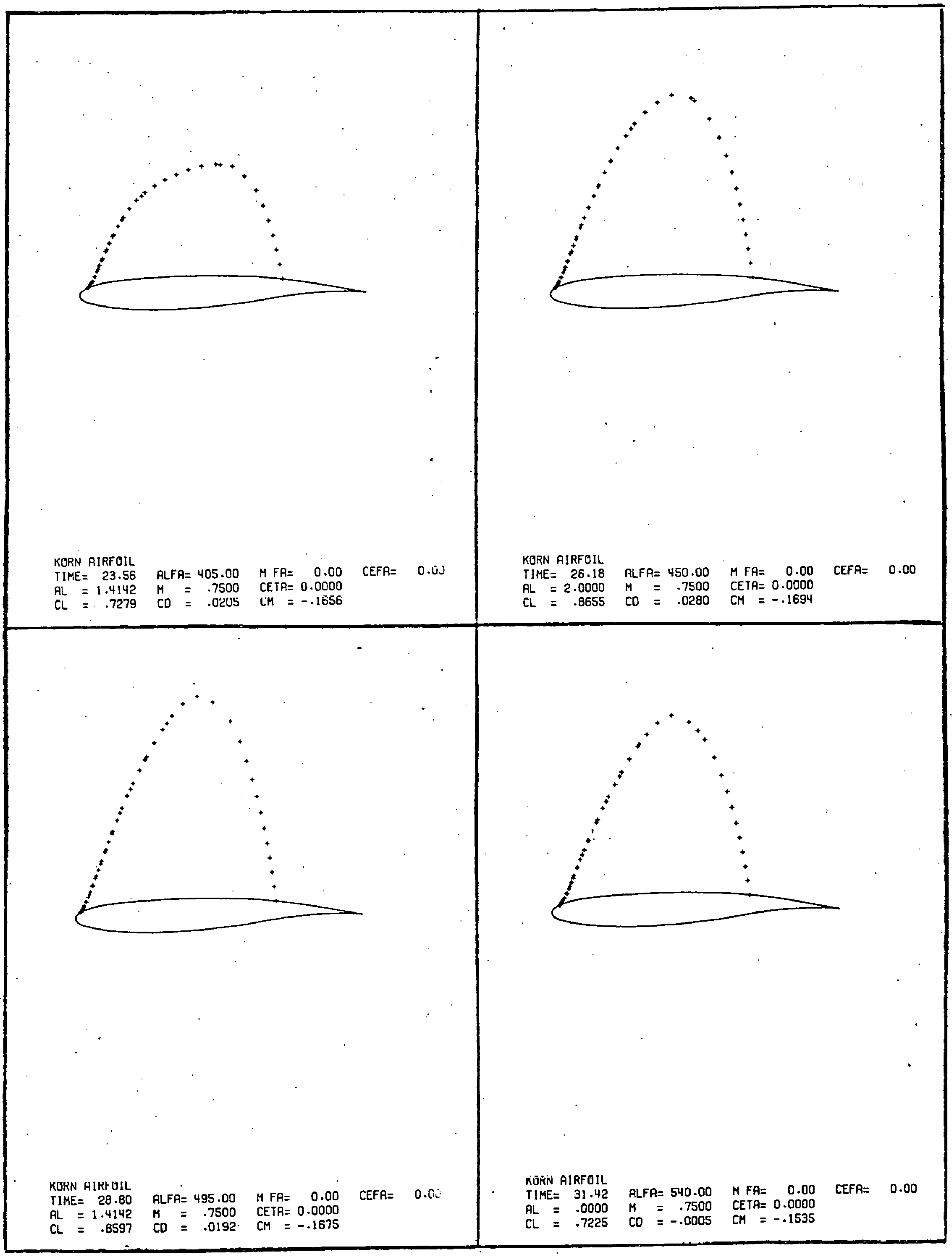
Figure 9k





NACA 0012
UNSTEADY TRACES OF AIRFOIL IN SINUSOIDAL RIGID BODY MOTIO
MEAN ATTACK ANGLE= 0.00 AMP= 1.00 FREQ RATE= .6
MEAN FLIGHT SPEED= .79 AMP= 0.00 FREQ RATE= 0.0
MEAN FLIGHT ANGLE= 0.00 AMP= 0.00 FREQ RATE= 0.0

Figure 91



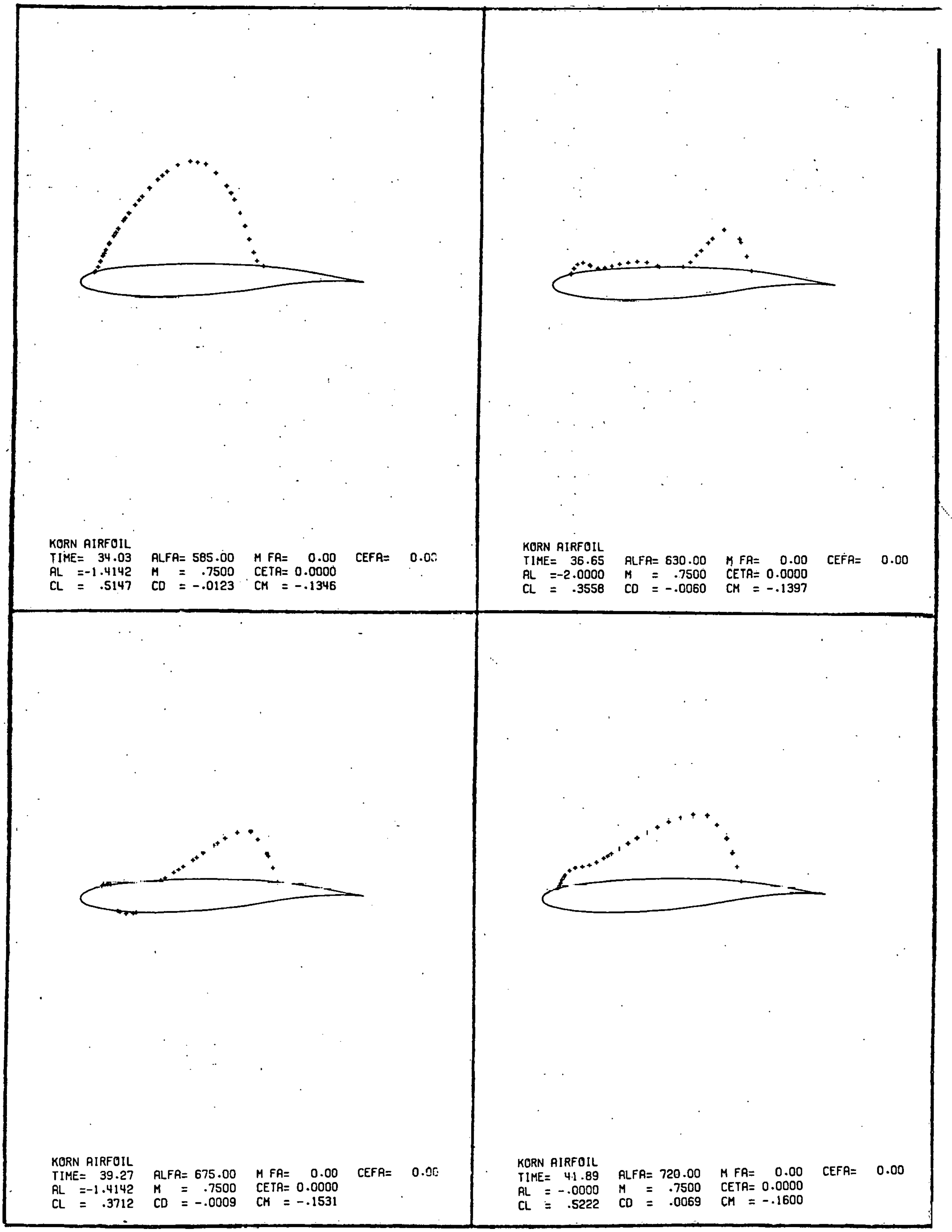
```
KORN AIRFOIL
TIME=  23.56   ALFA= 405.00   M FA=    0.00   CEFA=    0.0J
AL   = 1.4142  M    =   .7500  CETA= 0.0000
CL   =  .7279  CD   =   .0205  CM   = -.1656
```

```
KORN AIRFOIL
TIME=  26.18   ALFA= 450.00   M FA=    0.00   CEFA=    0.00
AL   = 2.0000  M    =   .7500  CETA= 0.0000
CL   =  .8655  CD   =   .0280  CM   = -.1694
```

```
KORN AIRFOIL
TIME=  28.80   ALFA= 495.00   M FA=    0.00   CEFA=    0.00
AL   = 1.4142  M    =   .7500  CETA= 0.0000
CL   =  .8597  CD   =   .0192  CM   = -.1675
```

```
KORN AIRFOIL
TIME=  31.42   ALFA= 540.00   M FA=    0.00   CEFA=    0.00
AL   =  .0000  M    =   .7500  CETA= 0.0000
CL   =  .7225  CD   = -.0005  CM   = -.1535
```

Figure 10a



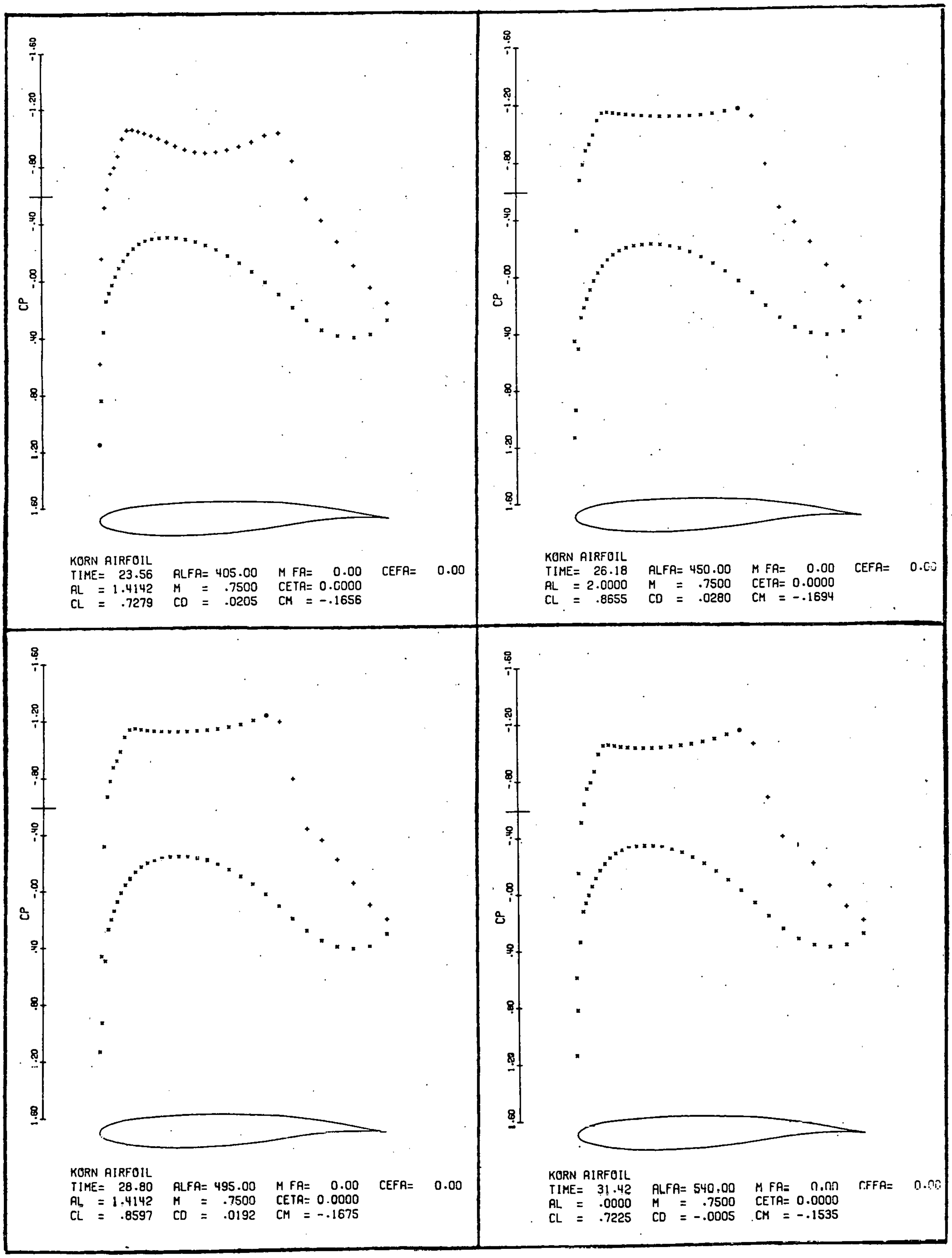
```
KORN AIRFOIL
TIME=  34.03   ALFA= 585.00   M FA=   0.00   CEFA=   0.00
AL  =-1.4142   M   =  .7500   CETA= 0.0000
CL  =  .5147   CD  = -.0123   CM  = -.1346
```

```
KORN AIRFOIL
TIME=  36.65   ALFA= 630.00   M FA=   0.00   CEFA=   0.00
AL  =-2.0000   M   =  .7500   CETA= 0.0000
CL  =  .3558   CD  = -.0060   CM  = -.1397
```

```
KORN AIRFOIL
TIME=  39.27   ALFA= 675.00   M FA=   0.00   CEFA=   0.00
AL  =-1.4142   M   =  .7500   CETA= 0.0000
CL  =  .3712   CD  = -.0009   CM  = -.1531
```

```
KORN AIRFOIL
TIME=  41.89   ALFA= 720.00   M FA=   0.00   CEFA=   0.00
AL  = -.0000   M   =  .7500   CETA= 0.0000
CL  =  .5222   CD  =  .0069   CM  = -.1600
```

Figure 10b



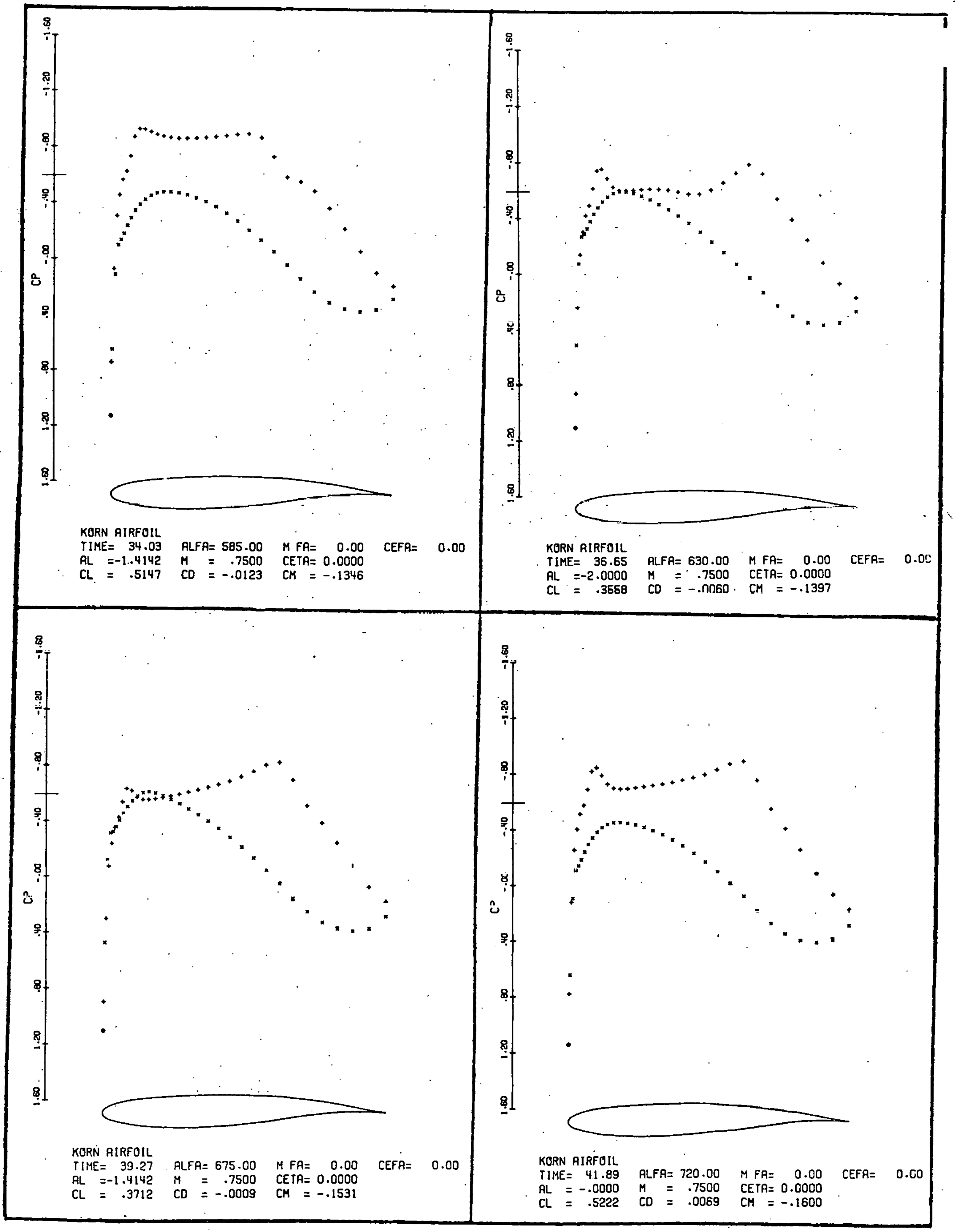
```
KORN AIRFOIL
TIME=  23.56   ALFA= 405.00   M FA=   0.00   CEFA=    0.00
AL   = 1.4142  M    =  .7500  CETA= 0.0000
CL   =  .7279  CD   =  .0205  CM   = -.1656
```

```
KORN AIRFOIL
TIME=  26.18   ALFA= 450.00   M FA=   0.00   CEFA=    0.00
AL   = 2.0000  M    =  .7500  CETA= 0.0000
CL   =  .8655  CD   =  .0280  CM   = -.1694
```

```
KORN AIRFOIL
TIME=  28.80   ALFA= 495.00   M FA=   0.00   CEFA=    0.00
AL   = 1.4142  M    =  .7500  CETA= 0.0000
CL   =  .8597  CD   =  .0192  CM   = -.1675
```

```
KORN AIRFOIL
TIME=  31.42   ALFA= 540.00   M FA=   0.00   CEFA=    0.00
AL   =  .0000  M    =  .7500  CETA= 0.0000
CL   =  .7225  CD   = -.0005  CM   = -.1535
```

Figure 10c



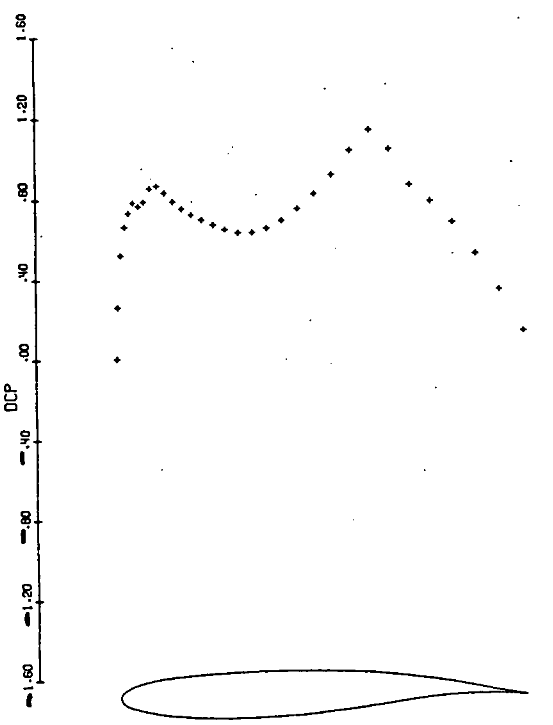
```
KORN AIRFOIL
TIME=  34.03   ALFA= 585.00   M FA=    0.00   CEFA=    0.00
AL   =-1.4142  M    =  .7500  CETA= 0.0000
CL   =  .5147  CD   = -.0123  CM   = -.1346
```

```
KORN AIRFOIL
TIME=  36.65   ALFA= 630.00   M FA=    0.00   CEFA=    0.00
AL   =-2.0000  M    =  .7500  CETA= 0.0000
CL   =  .3668  CD   = -.0060  CM   = -.1397
```

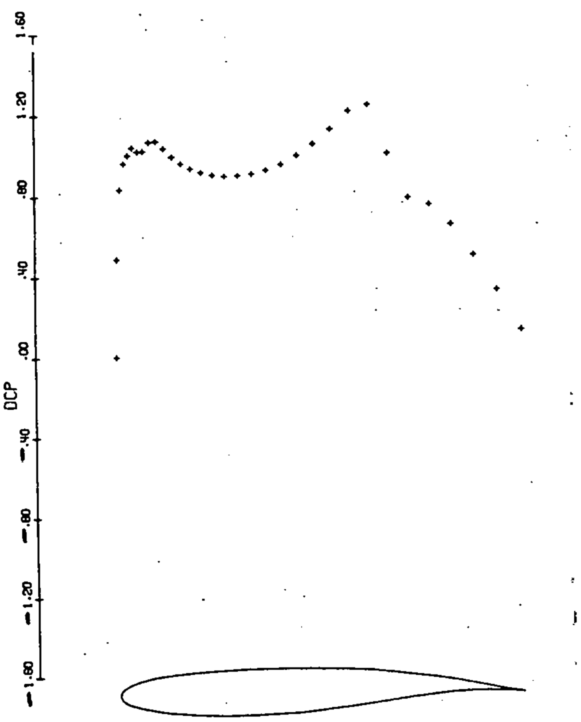
```
KORN AIRFOIL
TIME=  39.27   ALFA= 675.00   M FA=    0.00   CEFA=    0.00
AL   =-1.4142  M    =  .7500  CETA= 0.0000
CL   =  .3712  CD   = -.0009  CM   = -.1531
```

```
KORN AIRFOIL
TIME=  41.89   ALFA= 720.00   M FA=    0.00   CEFA=    0.00
AL   = -.0000  M    =  .7500  CETA= 0.0000
CL   =  .5222  CD   =  .0069  CM   = -.1600
```

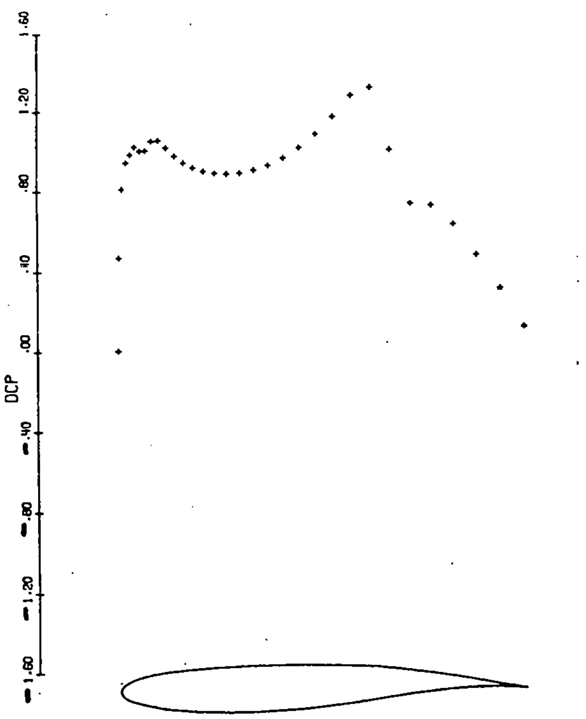
Figure 10d



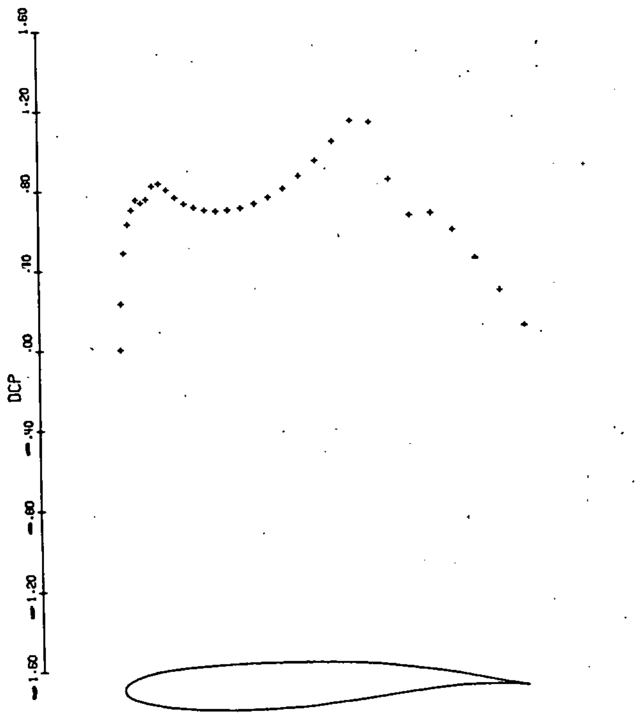
```
KORN AIRFOIL
TIME=  23.56   ALFA= 405.00   M FA=   0.00   CEFA=    0.00
AL   = 1.4142  M    =  .7500  CETA= 0.0000
CL   =  .7279  CD   =  .0205  CM   = -.1656
```



```
KORN AIRFOIL
TIME=  26.18   ALFA= 450.00   M FA=   0.00   CEFA=    0.00
AL   = 2.0000  M    =  .7500  CETA= 0.0000
CL   =  .8655  CD   =  .0280  CM   = -.1694
```

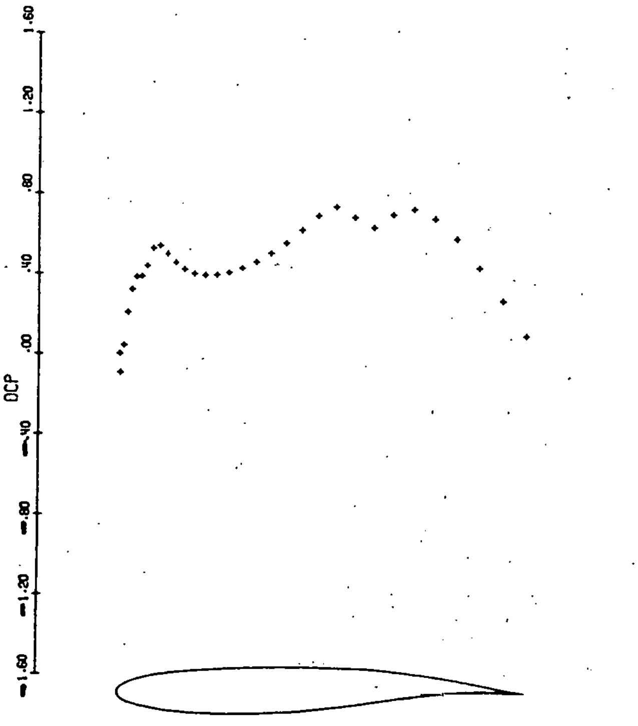


```
KORN AIRFOIL
TIME=  28.80   ALFA= 495.00   M FA=   0.00   CEFA=    0.00
AL   = 1.4142  M    =  .7500  CETA= 0.0000
CL   =  .8597  CD   =  .0192  CM   = -.1675
```

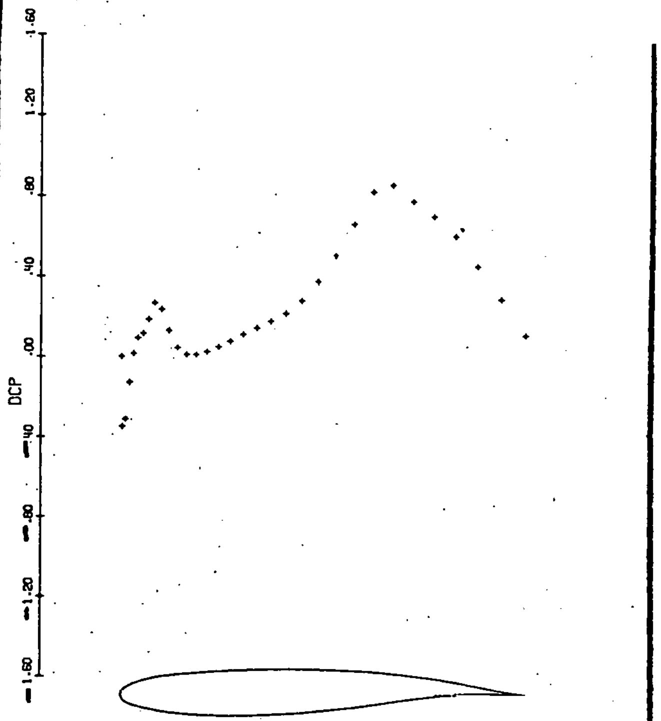


```
KORN AIRFOIL
TIME=  31.42   ALFA= 540.00   M FA=   0.00   CEFA=    0.00
AL   =  .0000  M    =  .7500  CETA= 0.0000
CL   =  .7225  CD   = -.0005  CM   = -.1535
```

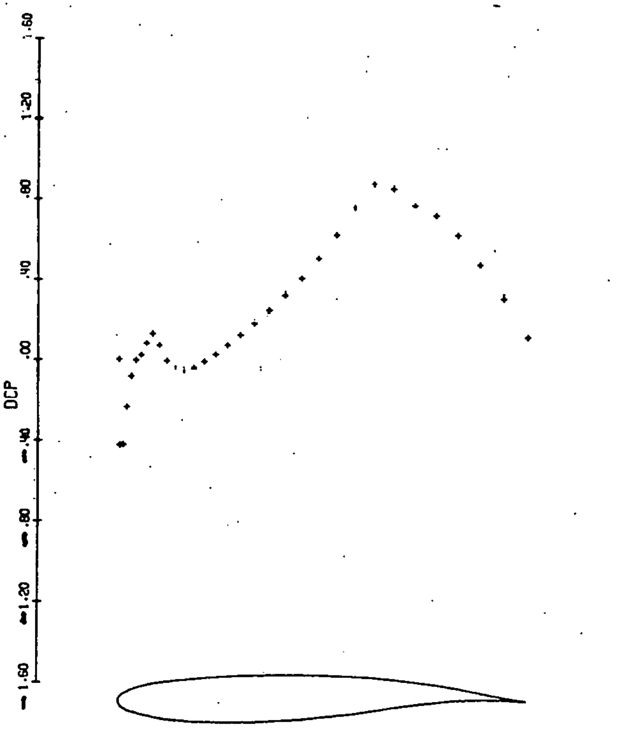
Figure 10e



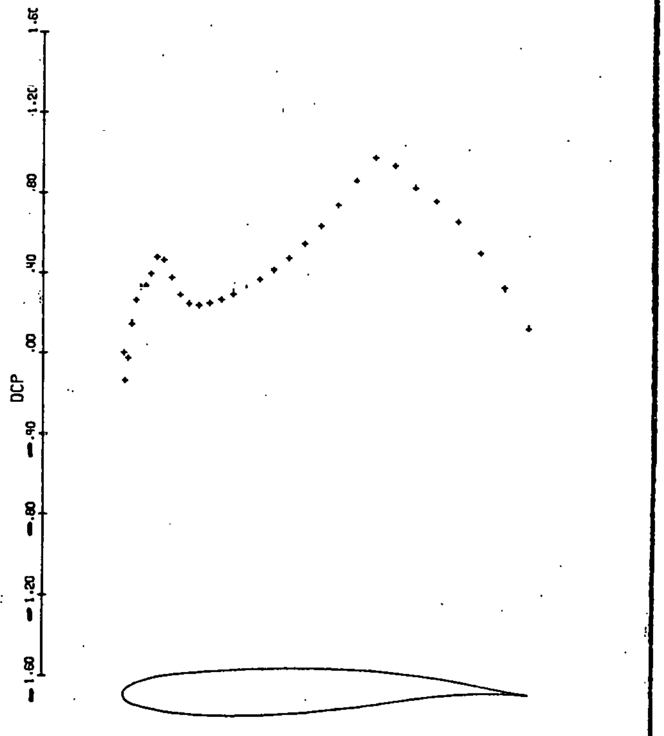
```
KORN AIRFOIL
TIME=  34.03    ALFA= 585.00    M FA=   0.00    CEFA=    0.00
AL  =-1.4142    M    =  .7500   CETA= 0.0000
CL  =   .5147   CD   = -.0123   CM   = -.1346
```



```
KORN AIRFOIL
TIME=  36.65    ALFA= 630.00    M FA=   0.00    CEFA=    0.00
AL  =-2.0000    M    =  .7500   CETA= 0.0000
CL  =  .3558    CD   = -.0060   CM   = -.1397
```

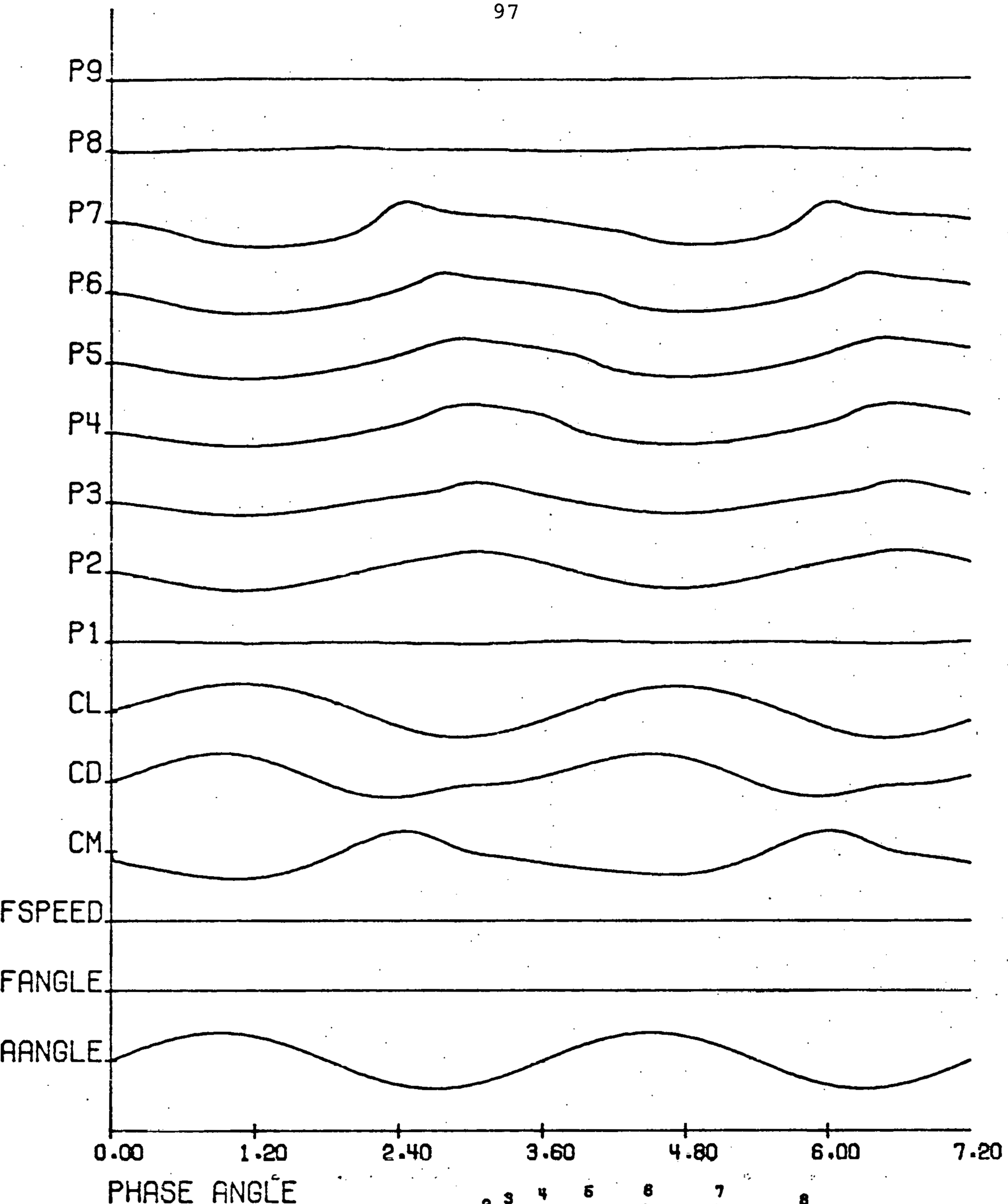


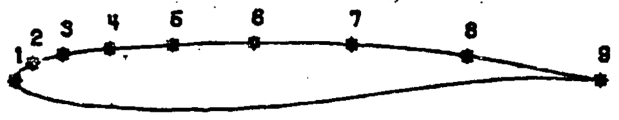
```
KORN AIRFOIL
TIME=  39.27    ALFA= 675.00    M FA=   0.00    CEFA=    0.00
AL  =-1.4142    M    =  .7500   CETA= 0.0000
CL  =  .3712    CD   = -.0009   CM   = -.1531
```



```
KORN AIRFOIL
TIME=  41.89    ALFA= 720.00    M FA=   0.00    CEFA=    0.00
AL  = -.0000    M    =  .7500   CETA= 0.0000
CL  =  .5222    CD   =  .0069   CM   = -.1600
```

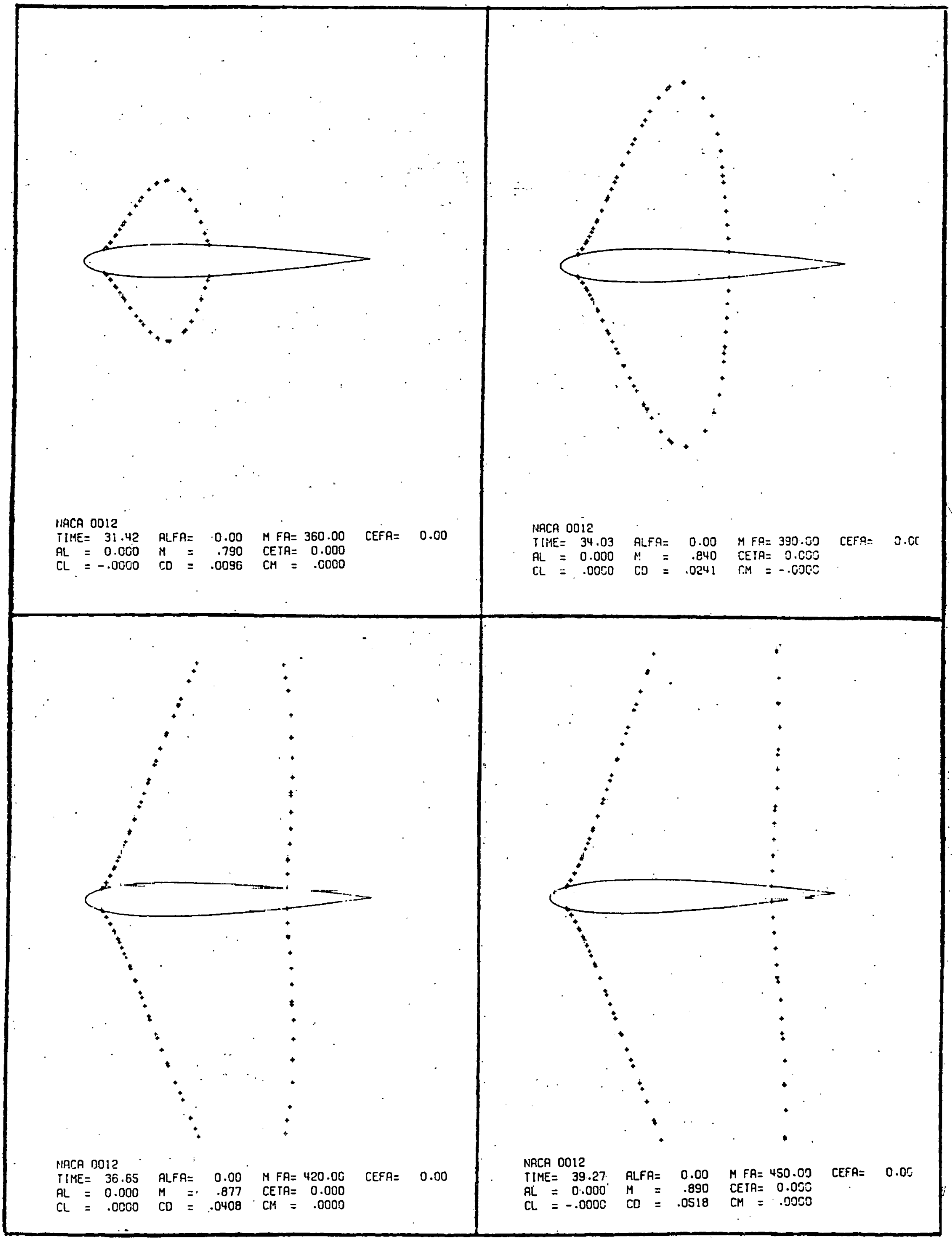
Figure 10f





KØRN AIRFØIL
UNSTEADY TRACES OF AIRFOIL IN SINUSOIDAL RIGID BODY MOTIO
MEAN ATTACK ANGLE= 0.00 AMP= 2.00 FREQ RATE= .30
MEAN FLIGHT SPEED= .75 AMP= 0.00 FREQ RATE= 0.00
MEAN FLIGHT ANGLE= 0.00 AMP= 0.00 FREQ RATE= 0.00

Figure 10q



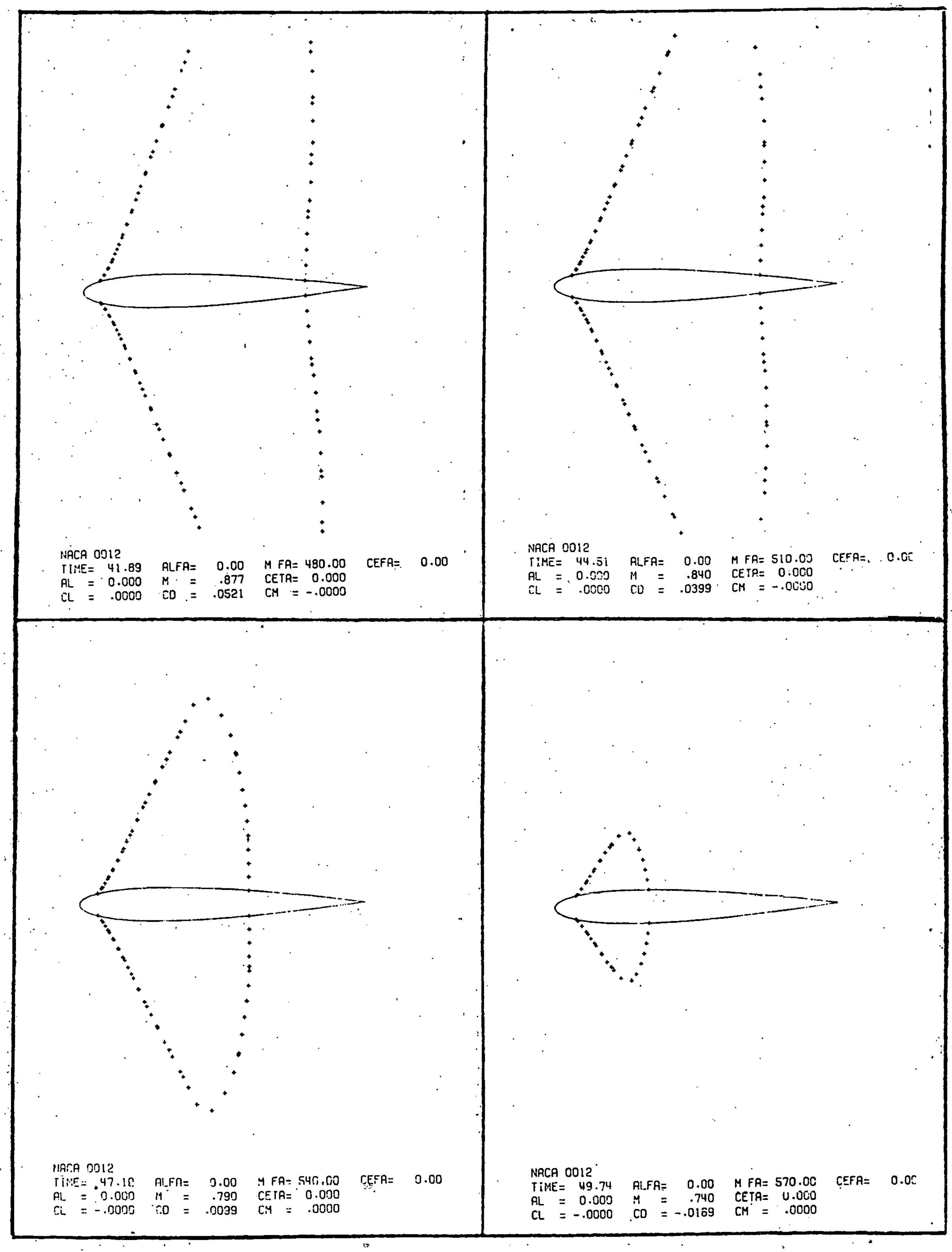
```
NACA 0012
TIME=  31.42   ALFA=   0.00   M FA= 360.00   CEFA=   0.00
AL  =  0.000   M    =   .790   CETA=  0.000
CL  = -.0000   CD   =  .0096   CM  =   .0000
```

```
NACA 0012
TIME=  34.03   ALFA=   0.00   M FA= 390.00   CEFA=   0.0
AL  =  0.000   M    =   .840   CETA=  0.000
CL  =  .0000   CD   =  .0241   CM  =  -.0000
```

```
NACA 0012
TIME=  36.65   ALFA=   0.00   M FA= 420.00   CEFA=   0.00
AL  =  0.000   M    =   .877   CETA=  0.000
CL  =  .0000   CD   =  .0408   CM  =   .0000
```

```
NACA 0012
TIME=  39.27   ALFA=   0.00   M FA= 450.00   CEFA=   0.00
AL  =  0.000   M    =   .890   CETA=  0.000
CL  = -.0000   CD   =  .0518   CM  =   .0000
```

Figure 11a



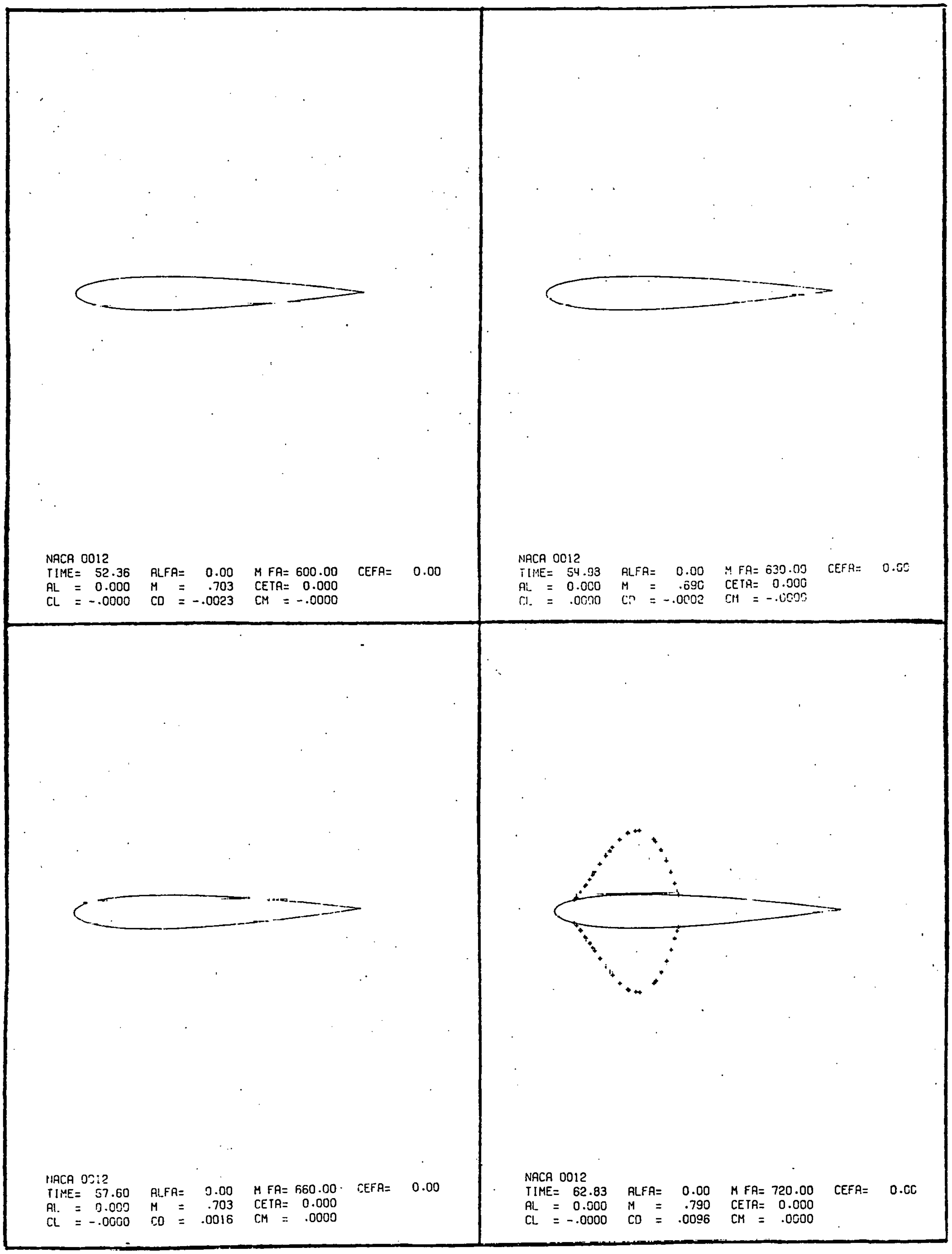
```
NACA 0012
TIME=  41.89   ALFA=   0.00   M FA= 480.00   CEFA=   0.00
AL  =  0.000   M   =   .877   CETA=  0.000
CL  =  .0000   CD  =  .0521   CM  = -.0000
```

```
NACA 0012
TIME=  44.51   ALFA=   0.00   M FA= 510.00   CEFA=   0.00
AL  =  0.000   M   =   .840   CETA=  0.000
CL  =  .0000   CD  =  .0399   CM  = -.0000
```

```
NACA 0012
TIME=  47.10   ALFA=   0.00   M FA= 540.00   CEFA=   0.00
AL  =  0.000   M   =   .790   CETA=  0.000
CL  = -.0000   CD  =  .0039   CM  =  .0000
```

```
NACA 0012
TIME=  49.74   ALFA=   0.00   M FA= 570.00   CEFA=   0.00
AL  =  0.000   M   =   .740   CETA=  0.000
CL  = -.0000   CD  = -.0169   CM  =  .0000
```

Figure 11b



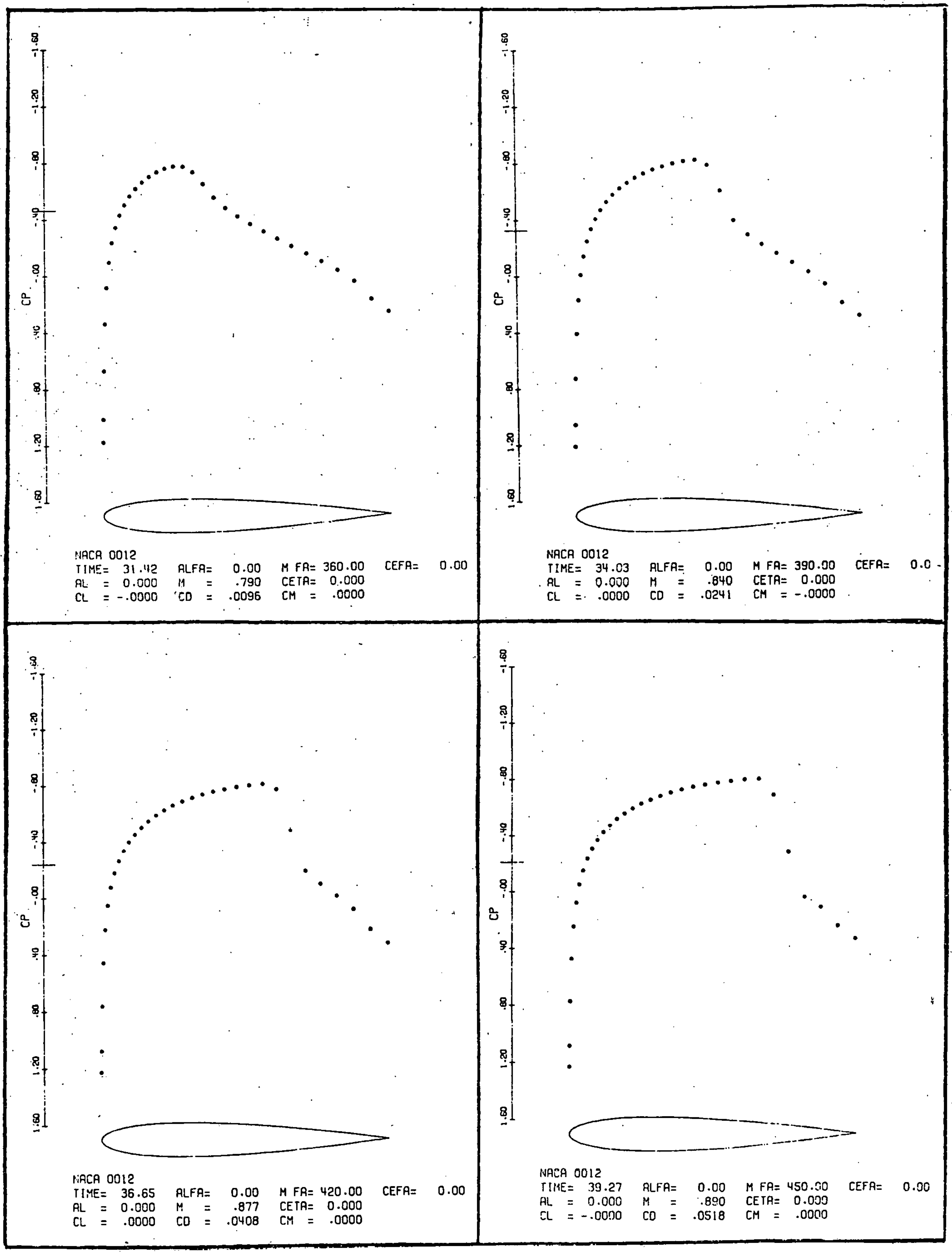
```
NACA 0012
TIME=  52.36   ALFA=    0.00   M FA= 600.00   CEFA=    0.00
AL   =  0.000   M    =    .703   CETA=  0.000
CL   = -.0000   CD   = -.0023   CM   = -.0000
```

```
NACA 0012
TIME=  54.93   ALFA=    0.00   M FA= 630.00   CEFA=    0.00
AL   =  0.000   M    =    .690   CETA=  0.000
CL   =  .0000   CD   = -.0002   CM   = -.0000
```

```
NACA 0012
TIME=  57.60   ALFA=    0.00   M FA= 660.00   CEFA=    0.00
AL   =  0.000   M    =    .703   CETA=  0.000
CL   = -.0000   CD   =   .0016   CM   =   .0000
```

```
NACA 0012
TIME=  62.83   ALFA=    0.00   M FA= 720.00   CEFA=    0.00
AL   =  0.000   M    =    .790   CETA=  0.000
CL   = -.0000   CD   =   .0096   CM   =   .0000
```

Figure 11c



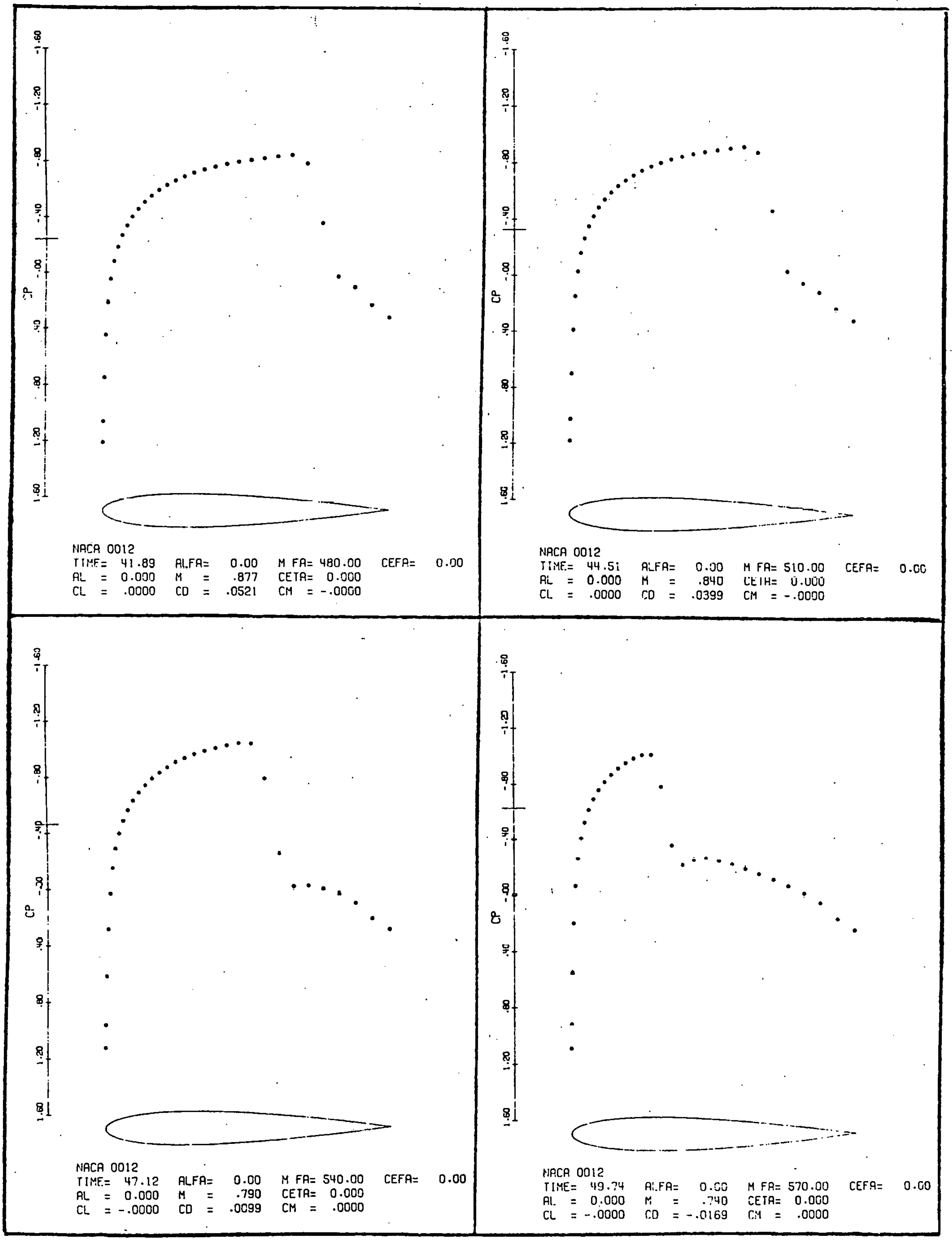
```
NACA 0012
TIME=  31.42   ALFA=   0.00   M FA= 360.00   CEFA=   0.00
AL   =  0.000  M    =   .790  CETA=  0.000
CL   = -.0000  CD   =  .0096  CM   =  .0000

NACA 0012
TIME=  34.03   ALFA=   0.00   M FA= 390.00   CEFA=   0.0
AL   =  0.000  M    =   .840  CETA=  0.000
CL   =  .0000  CD   =  .0241  CM   = -.0000

NACA 0012
TIME=  36.65   ALFA=   0.00   M FA= 420.00   CEFA=   0.00
AL   =  0.000  M    =   .877  CETA=  0.000
CL   =  .0000  CD   =  .0408  CM   =  .0000

NACA 0012
TIME=  39.27   ALFA=   0.00   M FA= 450.00   CEFA=   0.00
AL   =  0.000  M    =   .890  CETA=  0.000
CL   = -.0000  CD   =  .0518  CM   =  .0000
```

Figure 11d



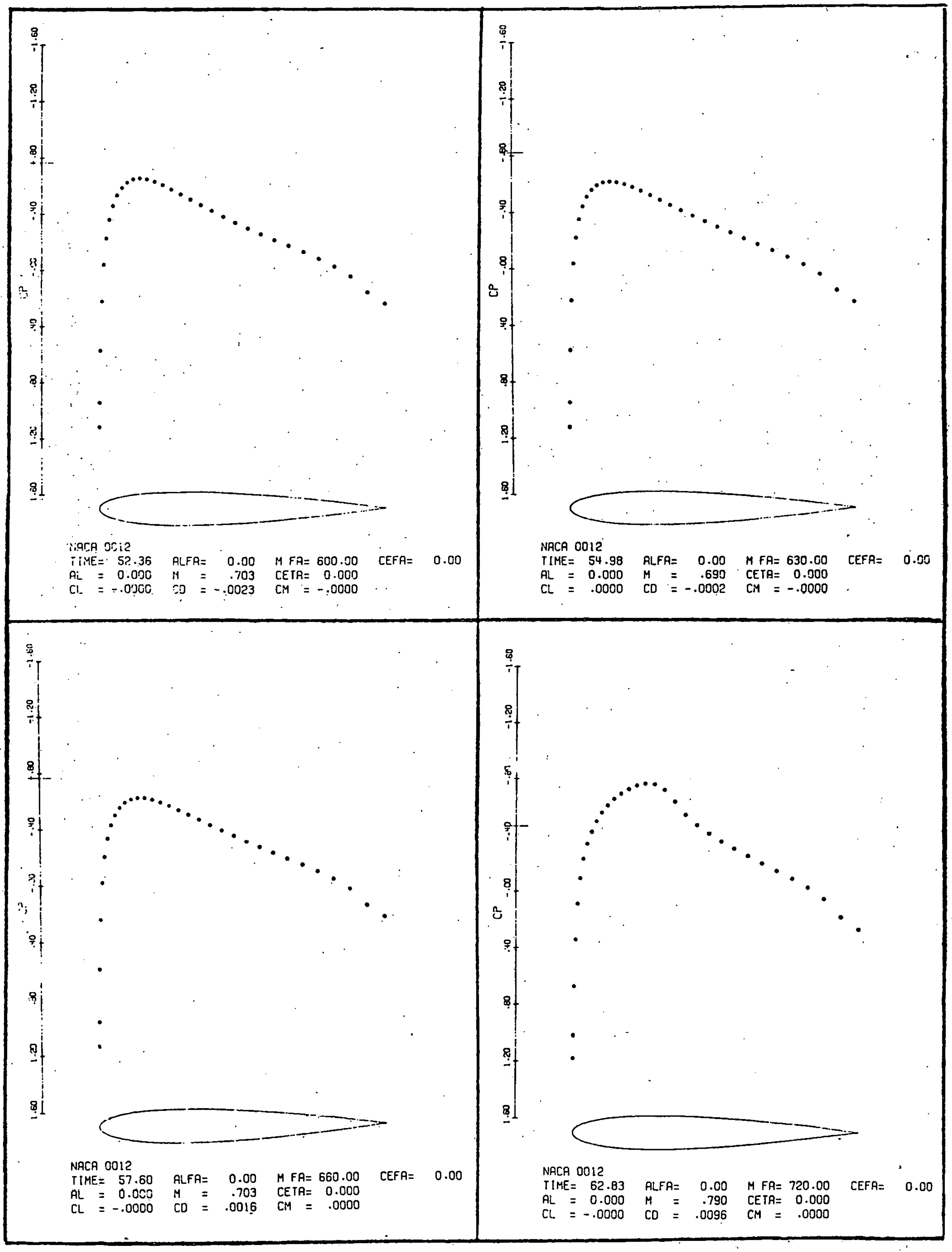
```
NACA 0012
TIME=  41.89   ALFA=    0.00   M FA= 480.00   CEFA=    0.00
AL   =  0.000   M    =    .877   CETA=  0.000
CL   =   .0000  CD   =   .0521   CM   = -.0000
```

```
NACA 0012
TIME=  44.51   ALFA=    0.00   M FA= 510.00   CEFA=    0.00
AL   =  0.000   M    =    .840   CETA=  0.000
CL   =   .0000  CD   =   .0399   CM   = -.0000
```

```
NACA 0012
TIME=  47.12   ALFA=    0.00   M FA= 540.00   CEFA=    0.00
AL   =  0.000   M    =    .790   CETA=  0.000
CL   = -.0000   CD   =   .0099   CM   =   .0000
```

```
NACA 0012
TIME=  49.74   ALFA=    0.00   M FA= 570.00   CEFA=    0.00
AL   =  0.000   M    =    .740   CETA=  0.000
CL   = -.0000   CD   = -.0169   CM   =   .0000
```

Figure 11e



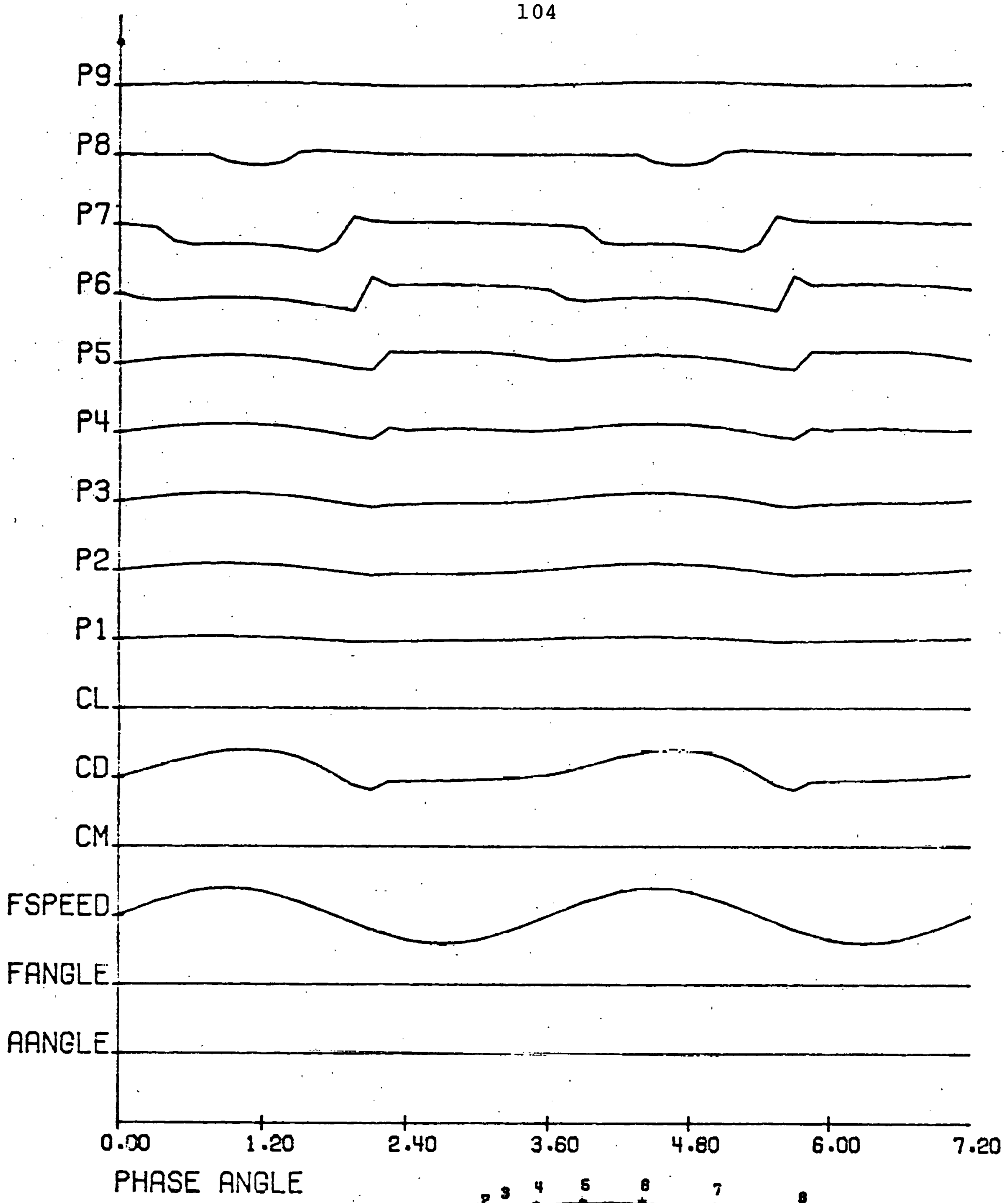
```
NACA 0012
TIME=  52.36   ALFA=   0.00   M FA= 600.00   CEFA=   0.00
AL  =  0.000   M    =  .703   CETA=  0.000
CL  = -.0000   CD   = -.0023  CM   = -.0000
```

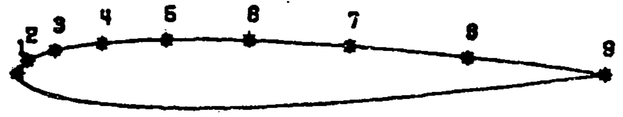
```
NACA 0012
TIME=  54.98   ALFA=   0.00   M FA= 630.00   CEFA=   0.00
AL  =  0.000   M    =  .690   CETA=  0.000
CL  =  .0000   CD   = -.0002  CM   = -.0000
```

```
NACA 0012
TIME=  57.60   ALFA=   0.00   M FA= 660.00   CEFA=   0.00
AL  =  0.000   M    =  .703   CETA=  0.000
CL  = -.0000   CD   =  .0016  CM   =  .0000
```

```
NACA 0012
TIME=  62.83   ALFA=   0.00   M FA= 720.00   CEFA=   0.00
AL  =  0.000   M    =  .790   CETA=  0.000
CL  = -.0000   CD   =  .0096  CM   =  .0000
```

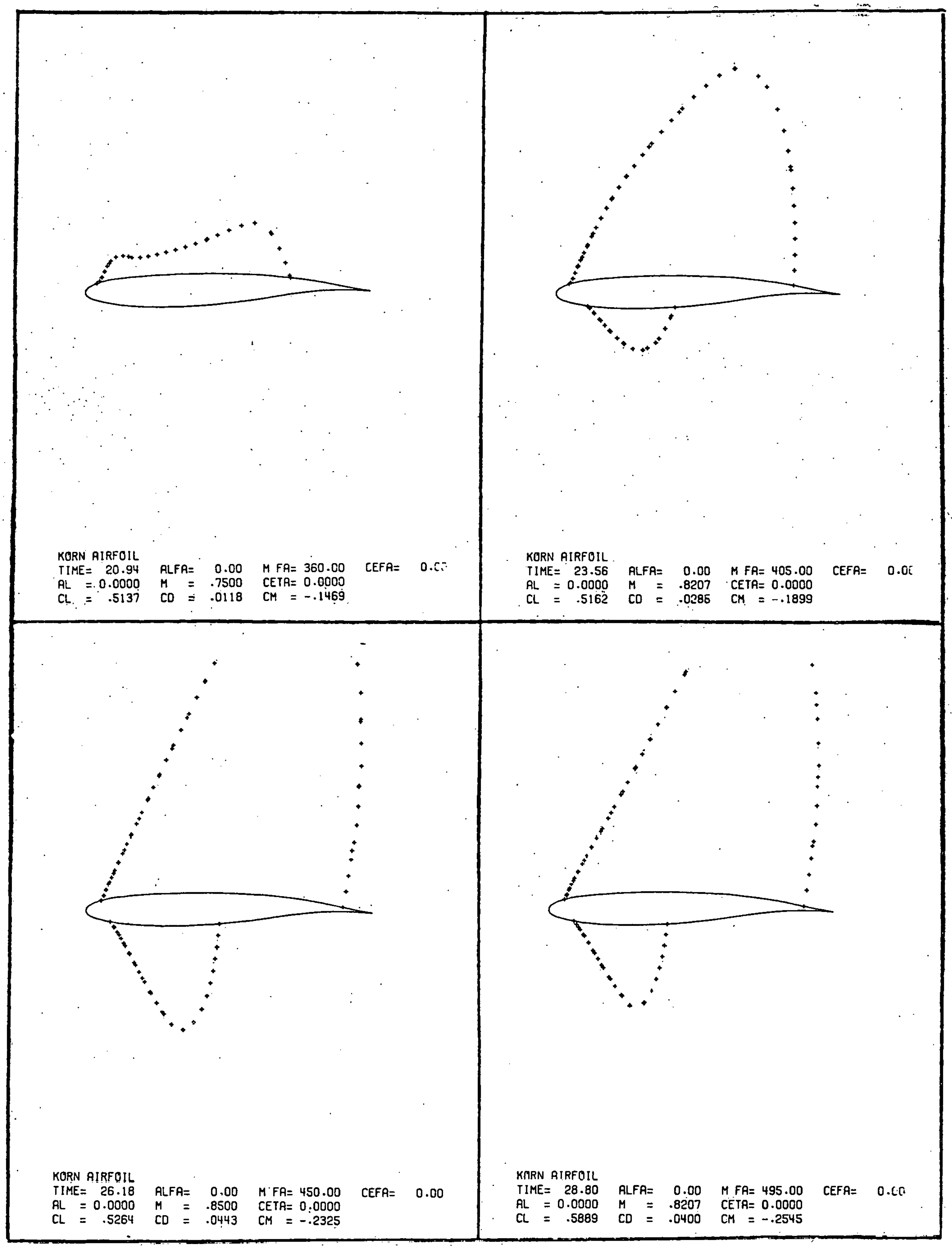
Figure 11f





NACA 0012
UNSTEADY TRACES OF AIRFOIL IN SINUSOIDAL RIGID BODY MOTIO
MEAN ATTACK ANGLE= 0.00 AMP= 0.00 FREQ RATE= 0.00
MEAN FLIGHT SPEED= .79 AMP= .10 FREQ RATE= .
MEAN FLIGHT ANGLE= 0.00 AMP= 0.00 FREQ RATE= 0.00

Figure 11q



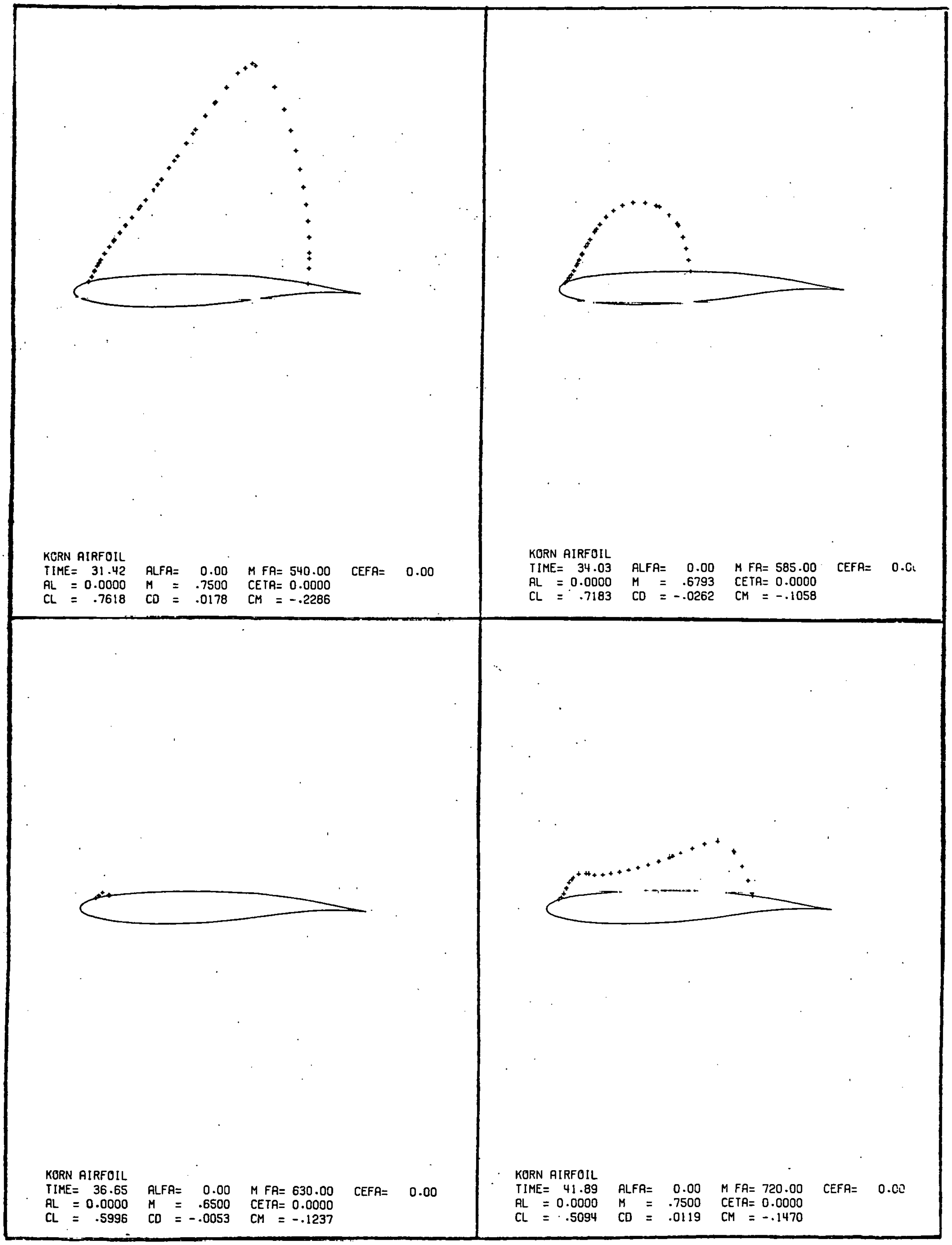
```
KORN AIRFOIL
TIME=  20.94   ALFA=   0.00   M FA= 360.00   CEFA=   0.00
AL  = 0.0000   M   =  .7500   CETA= 0.0000
CL  =  .5137   CD  =  .0118   CM  = -.1469
```

```
KORN AIRFOIL
TIME=  23.56   ALFA=   0.00   M FA= 405.00   CEFA=   0.0
AL  = 0.0000   M   =  .8207   CETA= 0.0000
CL  =  .5162   CD  =  .0286   CM  = -.1899
```

```
KORN AIRFOIL
TIME=  26.18   ALFA=   0.00   M FA= 450.00   CEFA=   0.00
AL  = 0.0000   M   =  .8500   CETA= 0.0000
CL  =  .5264   CD  =  .0443   CM  = -.2325
```

```
KORN AIRFOIL
TIME=  28.80   ALFA=   0.00   M FA= 495.00   CEFA=   0.00
AL  = 0.0000   M   =  .8207   CETA= 0.0000
CL  =  .5889   CD  =  .0400   CM  = -.2545
```

Figure 12a



```
KORN AIRFOIL
TIME=  31.42   ALFA=   0.00   M FA= 540.00   CEFA=   0.00
AL  = 0.0000   M    =  .7500  CETA= 0.0000
CL  =  .7618   CD   =  .0178  CM   = -.2286
```

```
KORN AIRFOIL
TIME=  34.03   ALFA=   0.00   M FA= 585.00   CEFA=   0.0
AL  = 0.0000   M    =  .6793  CETA= 0.0000
CL  =  .7183   CD   = -.0262  CM   = -.1058
```

```
KORN AIRFOIL
TIME=  36.65   ALFA=   0.00   M FA= 630.00   CEFA=   0.00
AL  = 0.0000   M    =  .6500  CETA= 0.0000
CL  =  .5996   CD   = -.0053  CM   = -.1237
```

```
KORN AIRFOIL
TIME=  41.89   ALFA=   0.00   M FA= 720.00   CEFA=   0.00
AL  = 0.0000   M    =  .7500  CETA= 0.0000
CL  =  .5094   CD   =  .0119  CM   = -.1470
```

Figure 12b

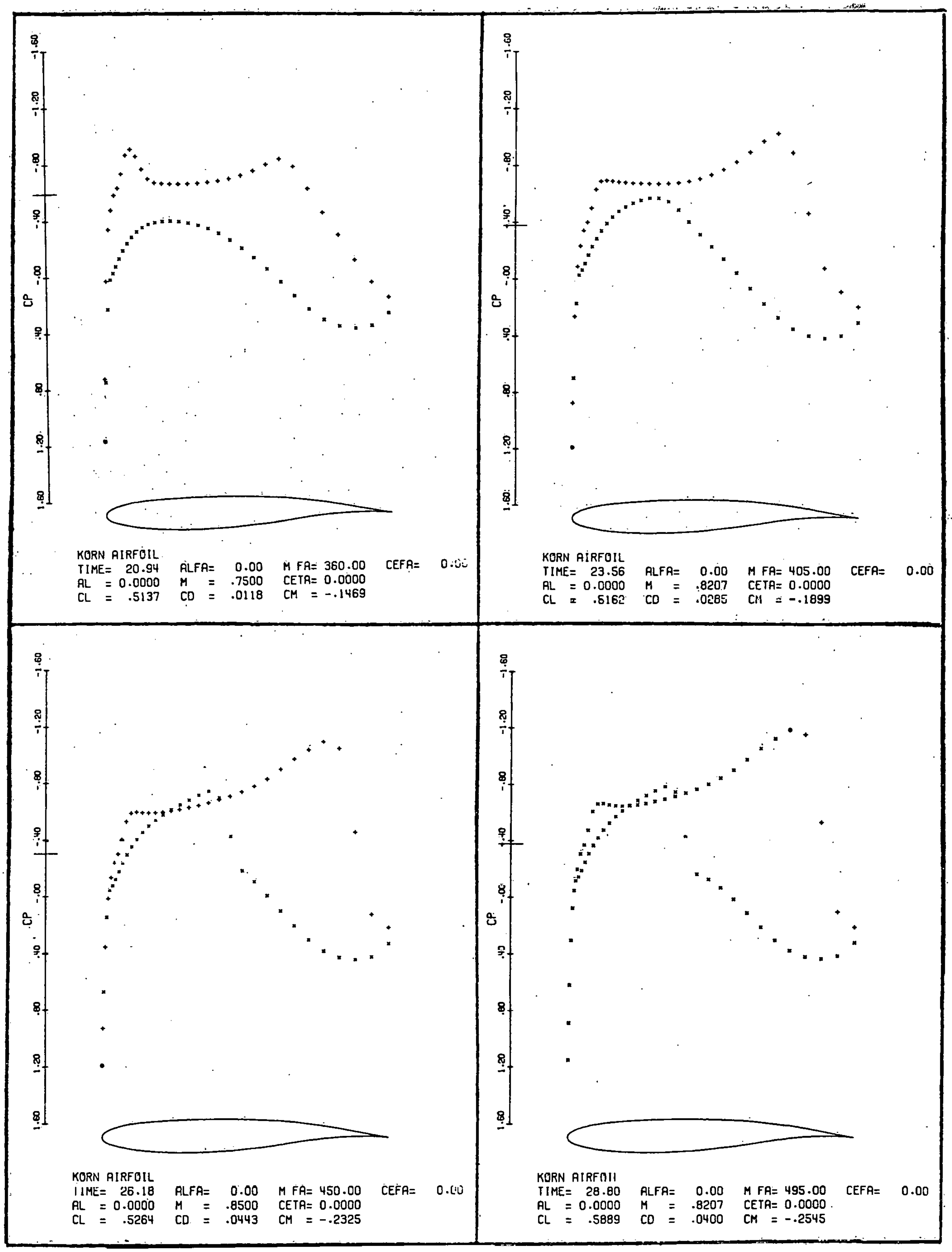
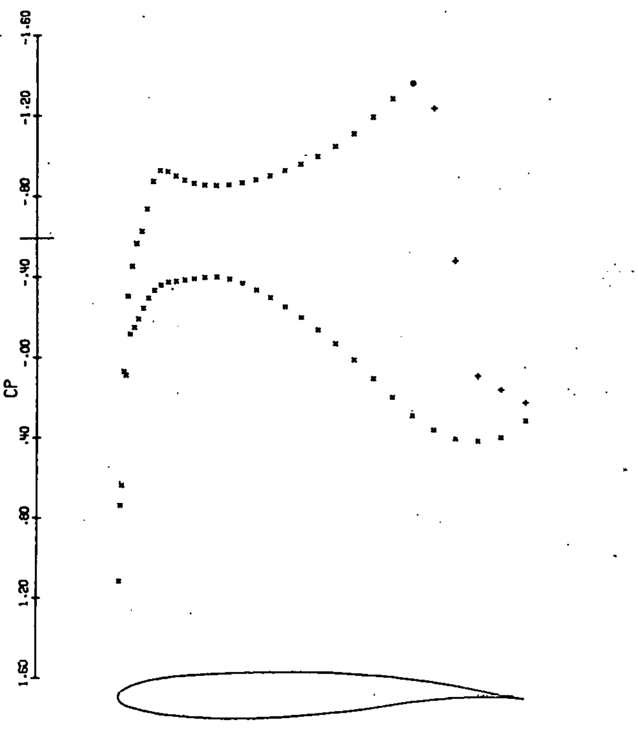
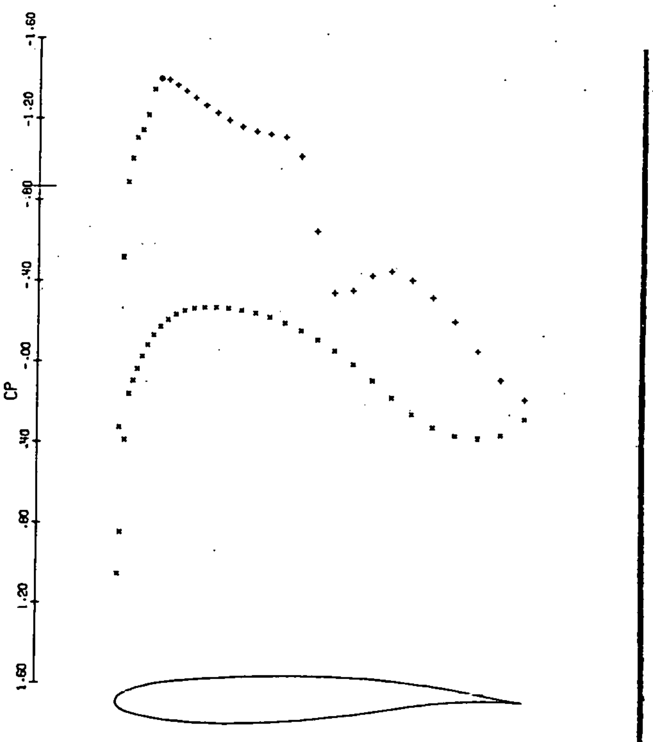

-1.60
-1.20
-.80
-.40
.00
.40
.80
1.20
1.60
CP
KORN AIRFOIL
TIME= 20.94 ALFA= 0.00 M FA= 360.00 CEFA= 0.00
AL = 0.0000 M = .7500 CETA= 0.0000
CL = .5137 CD = .0118 CM = -.1469
KORN AIRFOIL
TIME= 23.56 ALFA= 0.00 M FA= 405.00 CEFA= 0.00
AL = 0.0000 M = .8207 CETA= 0.0000
CL = .5162 CD = .0285 CM = -.1899
KORN AIRFOIL
TIME= 26.18 ALFA= 0.00 M FA= 450.00 CEFA= 0.00
AL = 0.0000 M = .8500 CETA= 0.0000
CL = .5264 CD = .0443 CM = -.2325
KORN AIRFOIL
TIME= 28.80 ALFA= 0.00 M FA= 495.00 CEFA= 0.00
AL = 0.0000 M = .8207 CETA= 0.0000
CL = .5889 CD = .0400 CM = -.2545

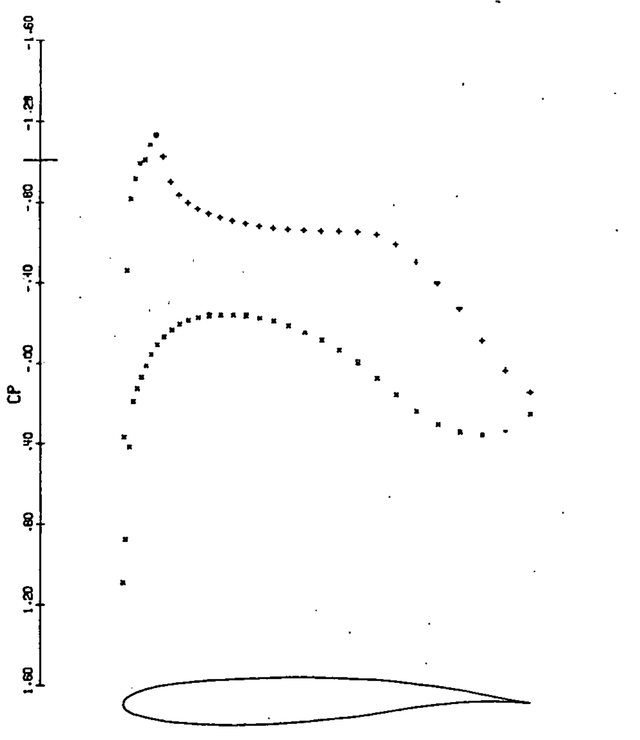
Figure 12c



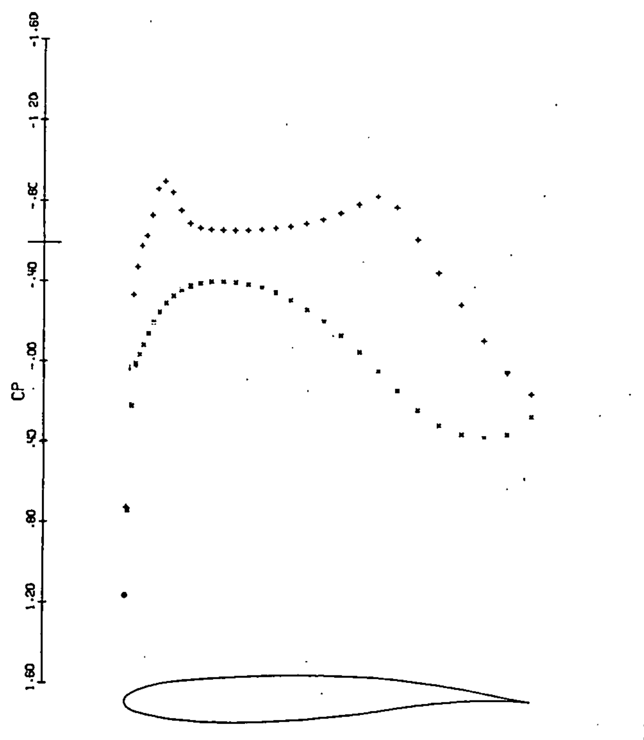
```
KORN AIRFOIL
TIME=  31.42   ALFA=    0.00   M FA= 540.00   CEFA=    0.00
AL   = 0.0000  M    =  .7500   CETA= 0.0000
CL   =   .7618  CD   =  .0178   CM   = -.2286
```



```
KORN AIRFOIL
TIME=  34.03   ALFA=    0.00   M FA= 585.00   CEFA=    0.00
AL   = 0.0000  M    =  .6793   CETA= 0.0000
CL   =   .7183  CD   = -.0262   CM   = -.1058
```

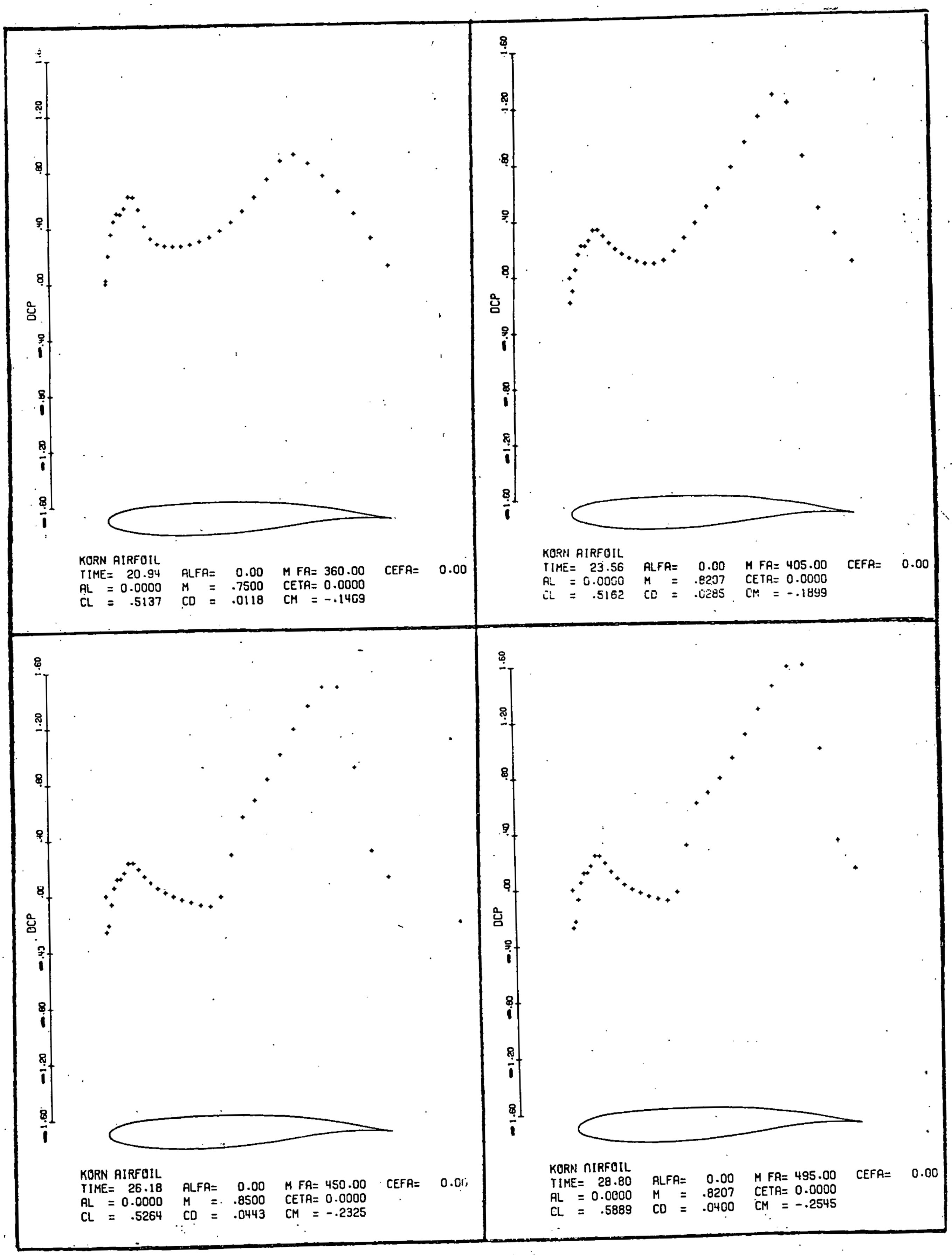


```
KORN AIRFOIL
TIME=  36.65   ALFA=    0.00   M FA= 630.00   CEFA=    0.00
AL   = 0.0000  M    =  .6500   CETA= 0.0000
CL   =   .5996  CD   = -.0053   CM   = -.1237
```



```
KORN AIRFOIL
TIME=  41.89   ALFA=    0.00   M FA= 720.00   CEFA=    0.00
AL   = 0.0000  M    =  .7500   CETA= 0.0000
CL   =   .5094  CD   =  .0119   CM   = -.1470
```

Figure 12d



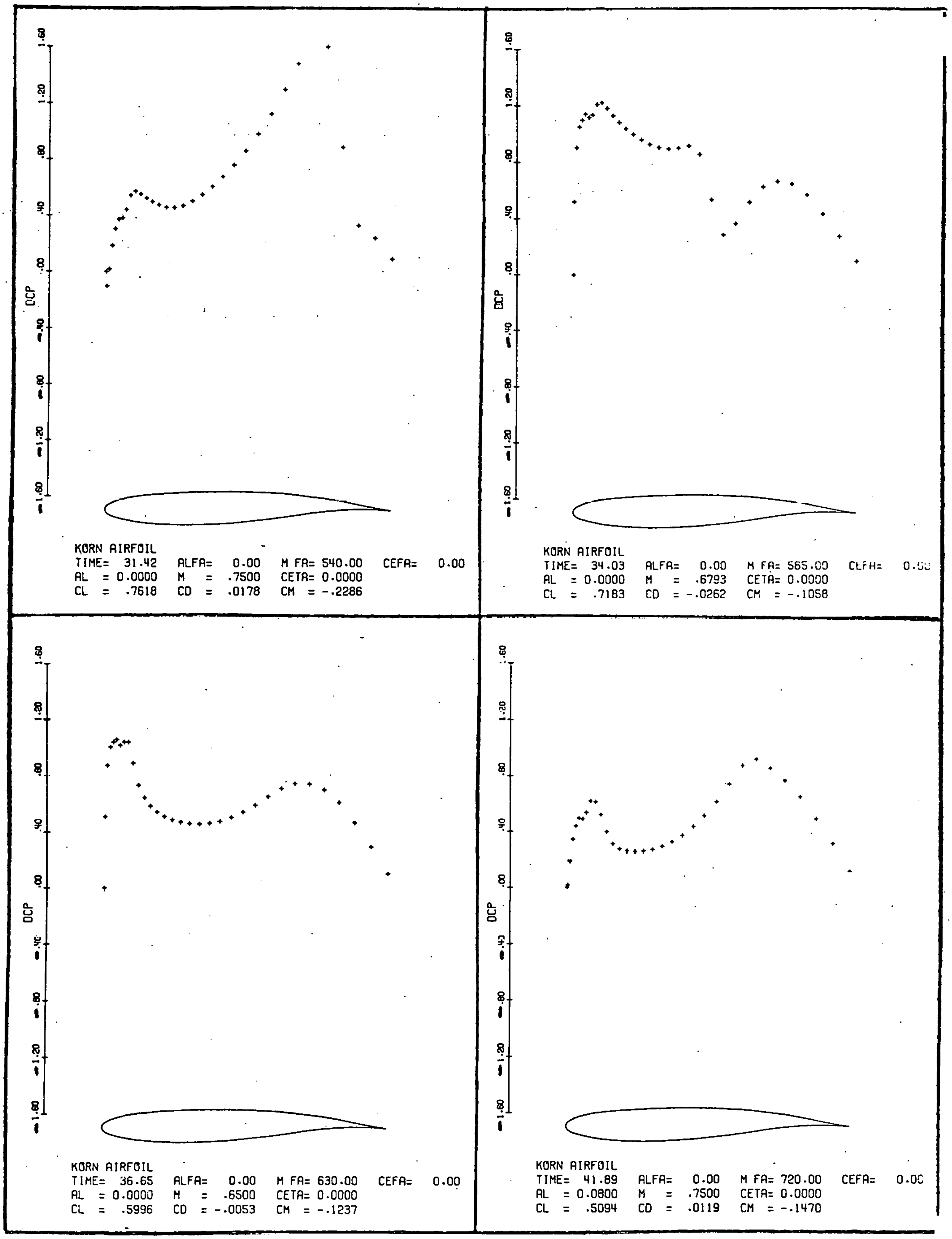
```
KORN AIRFOIL
TIME=  20.94   ALFA=    0.00   M FA= 360.00   CEFA=    0.00
AL   = 0.0000  M    =  .7500   CETA= 0.0000
CL   =  .5137  CD   =  .0118   CM   = -.1409
```

```
KORN AIRFOIL
TIME=  23.56   ALFA=    0.00   M FA= 405.00   CEFA=    0.00
AL   = 0.0000  M    =  .8207   CETA= 0.0000
CL   =  .5162  CD   =  .0285   CM   = -.1899
```

```
KORN AIRFOIL
TIME=  26.18   ALFA=    0.00   M FA= 450.00   CEFA=    0.00
AL   = 0.0000  M    =  .8500   CETA= 0.0000
CL   =  .5264  CD   =  .0443   CM   = -.2325
```

```
KORN AIRFOIL
TIME=  28.80   ALFA=    0.00   M FA= 495.00   CEFA=    0.00
AL   = 0.0000  M    =  .8207   CETA= 0.0000
CL   =  .5889  CD   =  .0400   CM   = -.2545
```

Figure 12e



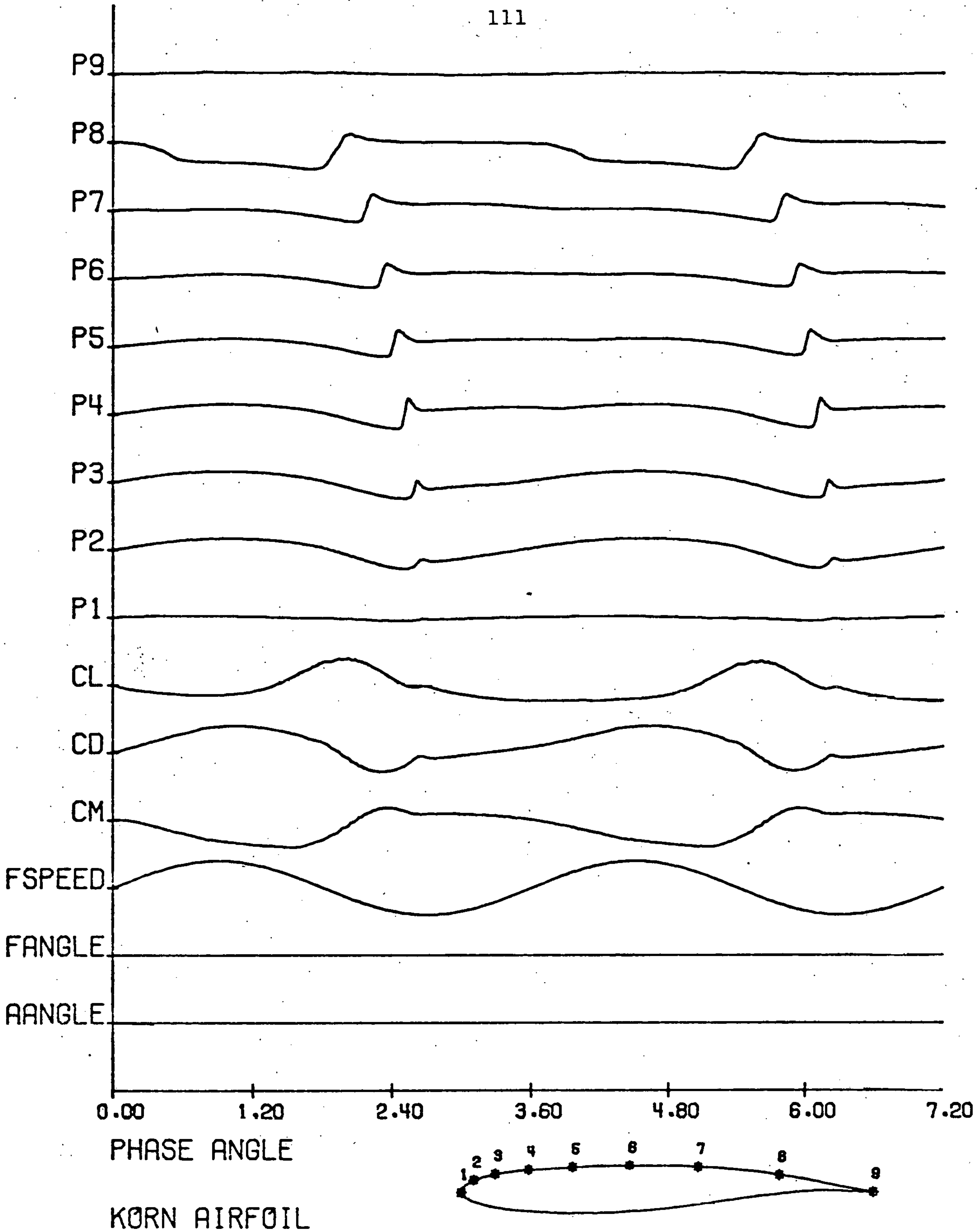
```
KORN AIRFOIL
TIME=  31.42   ALFA=   0.00   M FA= 540.00   CEFA=   0.00
AL  = 0.0000   M   =  .7500   CETA= 0.0000
CL  =  .7618   CD  =  .0178   CM  = -.2286
```

```
KORN AIRFOIL
TIME=  34.03   ALFA=   0.00   M FA= 565.00   CEFA=   0.00
AL  = 0.0000   M   =  .6793   CETA= 0.0000
CL  =  .7183   CD  = -.0262   CM  = -.1058
```

```
KORN AIRFOIL
TIME=  36.65   ALFA=   0.00   M FA= 630.00   CEFA=   0.00
AL  = 0.0000   M   =  .6500   CETA= 0.0000
CL  =  .5996   CD  = -.0053   CM  = -.1237
```

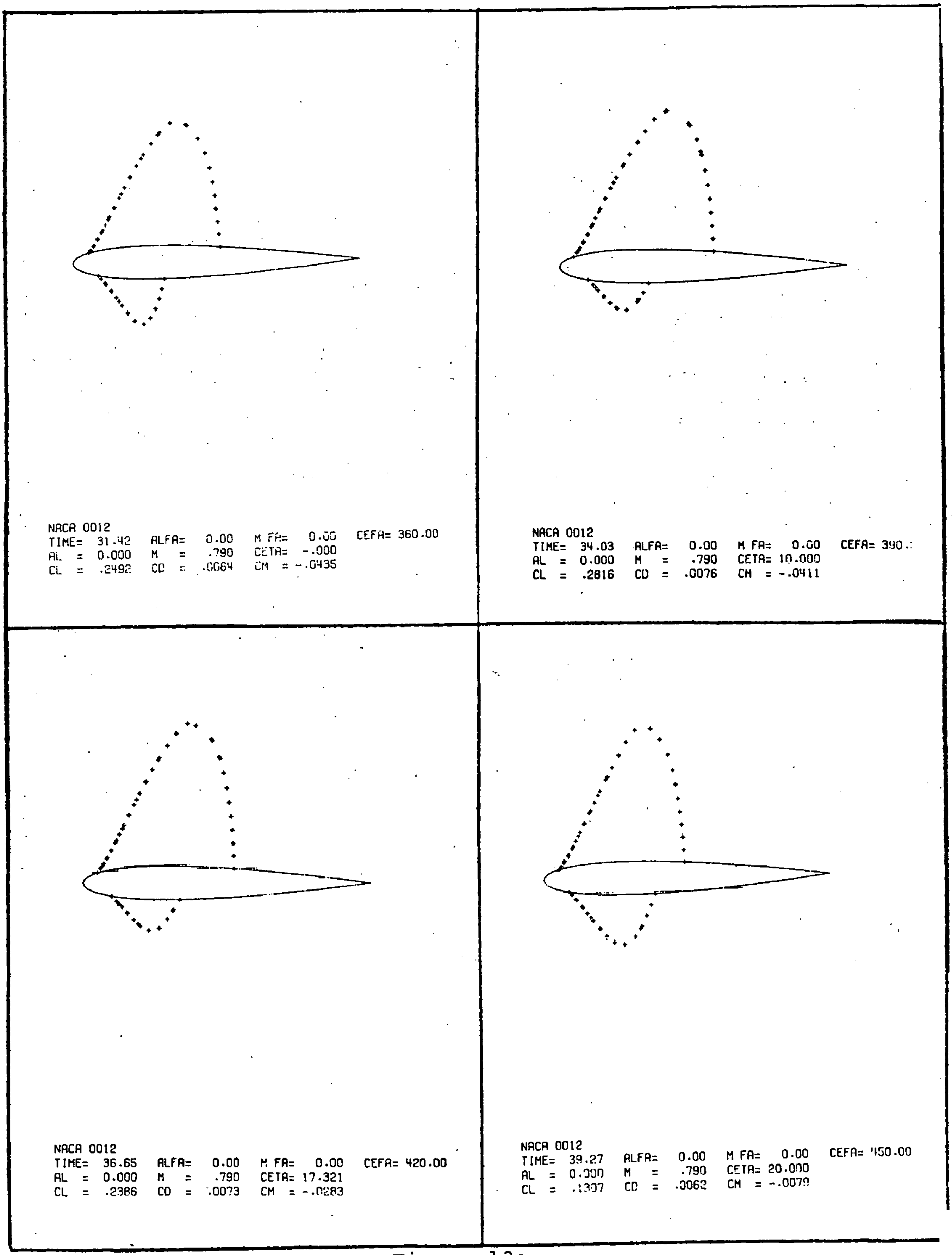
```
KORN AIRFOIL
TIME=  41.89   ALFA=   0.00   M FA= 720.00   CEFA=   0.00
AL  = 0.0800   M   =  .7500   CETA= 0.0000
CL  =  .5094   CD  =  .0119   CM  = -.1470
```

Figure 12f



KØRN AIRFØIL
UNSTEADY TRACES OF AIRFOIL IN SINUSOIDAL RIGID BODY MOTION
MEAN ATTACK ANGLE= 0.00 AMP= 0.00 FREQ RATE= 0.00
MEAN FLIGHT SPEED= .75 AMP= .10 FREQ RATE= .30
MEAN FLIGHT ANGLE= 0.00 AMP= 0.00 FREQ RATE= 0.00

Figure 12g



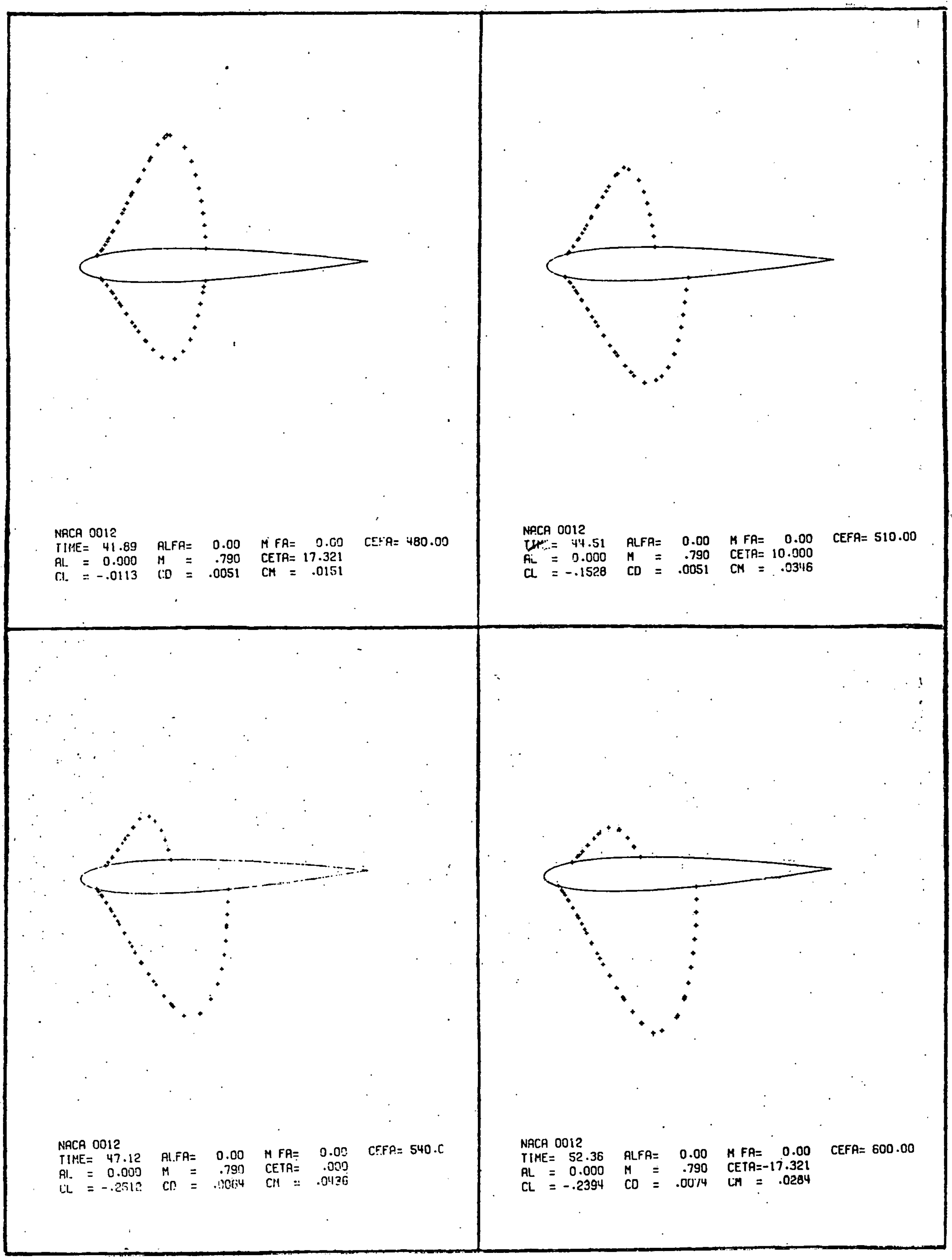
```
NACA 0012
TIME=  31.42   ALFA=   0.00   M FA=   0.00   CEFA= 360.00
AL   =  0.000  M    =   .790  CETA=  -.000
CL   =  .2492  CD   =  .0064  CM   = -.0435
```

```
NACA 0012
TIME=  34.03   ALFA=   0.00   M FA=   0.00   CEFA= 390.
AL   =  0.000  M    =   .790  CETA= 10.000
CL   =  .2816  CD   =  .0076  CM   = -.0411
```

```
NACA 0012
TIME=  36.65   ALFA=   0.00   M FA=   0.00   CEFA= 420.00
AL   =  0.000  M    =   .790  CETA= 17.321
CL   =  .2386  CD   =  .0073  CM   = -.0283
```

```
NACA 0012
TIME=  39.27   ALFA=   0.00   M FA=   0.00   CEFA= 450.00
AL   =  0.000  M    =   .790  CETA= 20.000
CL   =  .1307  CD   =  .0062  CM   = -.0079
```

Figure 13a



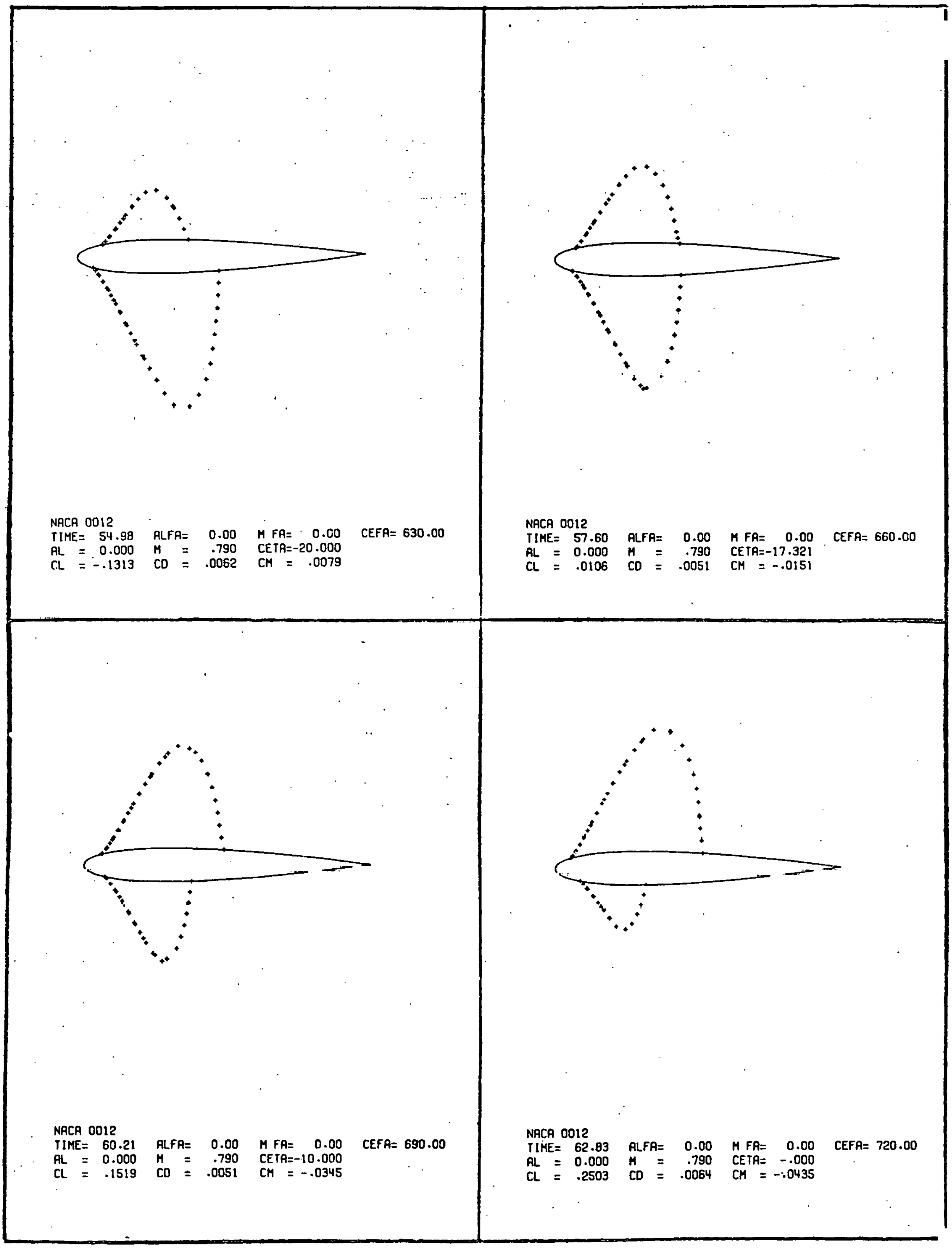
```
NACA 0012
TIME=  41.89   ALFA=   0.00   M FA=   0.00   CEFA= 480.00
AL   =  0.000  M    =   .790  CETA= 17.321
CL   = -.0113  CD   =  .0051  CM   =  .0151
```

```
NACA 0012
TIME=  44.51   ALFA=   0.00   M FA=   0.00   CEFA= 510.00
AL   =  0.000  M    =   .790  CETA= 10.000
CL   = -.1528  CD   =  .0051  CM   =  .0346
```

```
NACA 0012
TIME=  47.12   ALFA=   0.00   M FA=   0.00   CEFA= 540.C
AL   =  0.000  M    =   .790  CETA=   .000
CL   = -.2512  CD   =  .0064  CM   =  .0436
```

```
NACA 0012
TIME=  52.36   ALFA=   0.00   M FA=   0.00   CEFA= 600.00
AL   =  0.000  M    =   .790  CETA=-17.321
CL   = -.2394  CD   =  .0074  CM   =  .0284
```

Figure 13b



```
NACA 0012
TIME=  54.98   ALFA=    0.00   M FA=   0.00   CEFA= 630.00
AL  =  0.000   M    =   .790   CETA=-20.000
CL  = -.1313   CD   =  .0062   CM   =  .0079
```

```
NACA 0012
TIME=  57.60   ALFA=    0.00   M FA=   0.00   CEFA= 660.00
AL  =  0.000   M    =   .790   CETA=-17.321
CL  =  .0106   CD   =  .0051   CM   = -.0151
```

```
NACA 0012
TIME=  60.21   ALFA=    0.00   M FA=   0.00   CEFA= 690.00
AL  =  0.000   M    =   .790   CETA=-10.000
CL  =  .1519   CD   =  .0051   CM   = -.0345
```

```
NACA 0012
TIME=  62.83   ALFA=    0.00   M FA=   0.00   CEFA= 720.00
AL  =  0.000   M    =   .790   CETA=  -.000
CL  =  .2503   CD   =  .0064   CM   = -.0435
```

Figure 13c

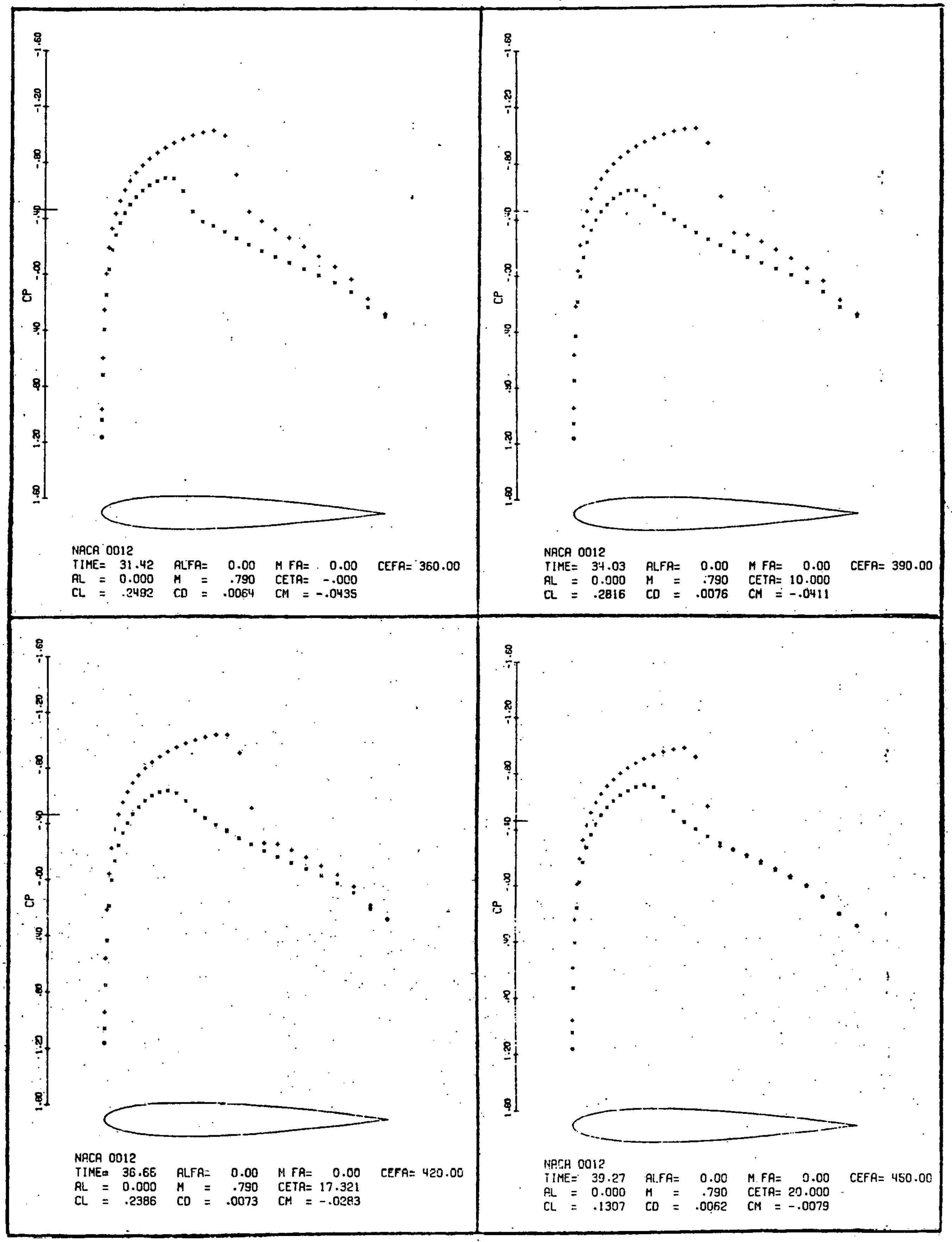
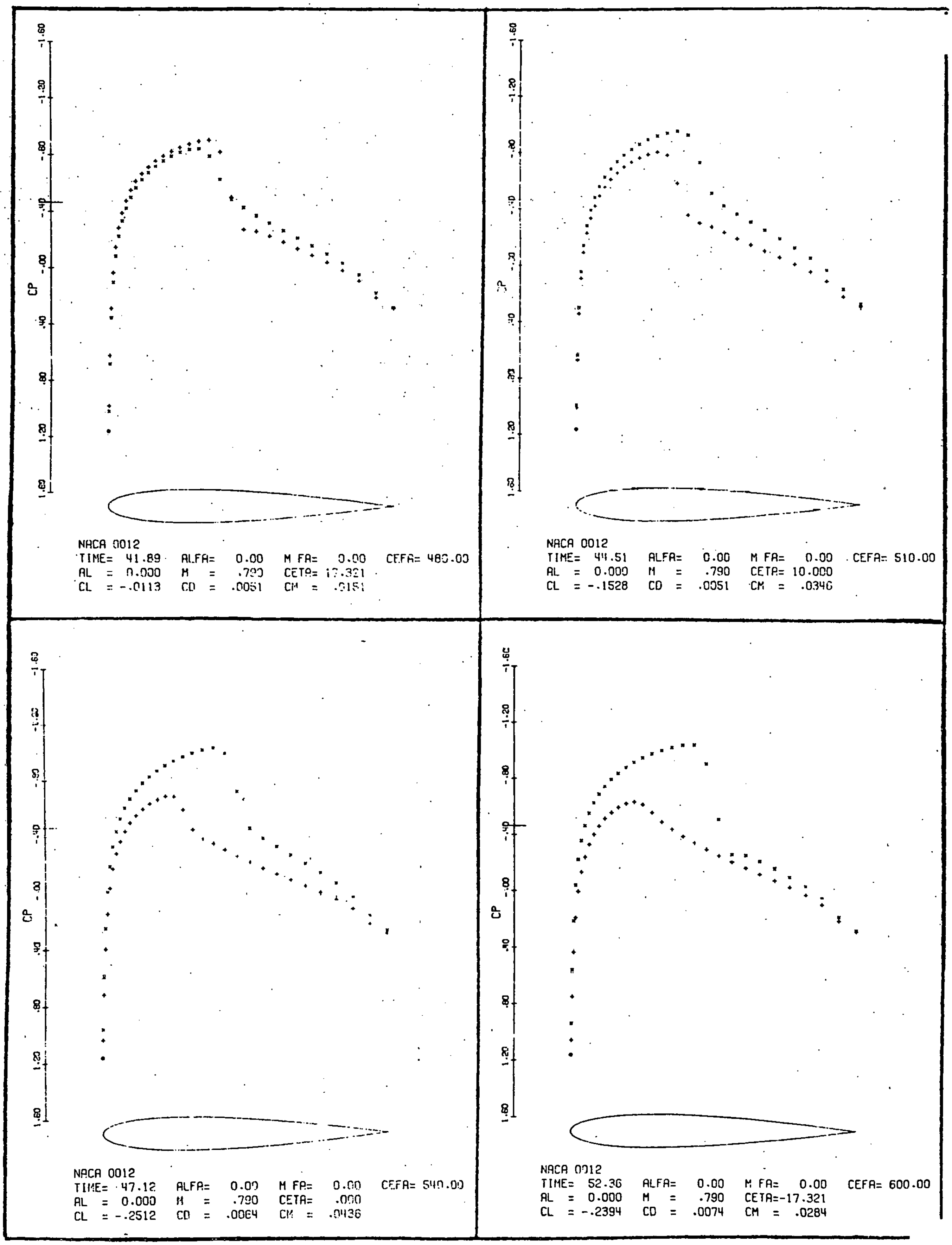

CP
-1.60
-1.20
-.80
-.40
.00
.40
.80
1.20
1.60
NACA 0012
TIME= 31.42 ALFA= 0.00 M FA= 0.00 CEFA= 360.00
AL = 0.000 M = .790 CETA= -.000
CL = .2492 CD = .0064 CM = -.0435
NACA 0012
TIME= 34.03 ALFA= 0.00 M FA= 0.00 CEFA= 390.00
AL = 0.000 M = .790 CETA= 10.000
CL = .2816 CD = .0076 CM = -.0411
NACA 0012
TIME= 36.66 ALFA= 0.00 M FA= 0.00 CEFA= 420.00
AL = 0.000 M = .790 CETA= 17.321
CL = .2386 CD = .0073 CM = -.0283
NACA 0012
TIME= 39.27 ALFA= 0.00 M FA= 0.00 CEFA= 450.00
AL = 0.000 M = .790 CETA= 20.000
CL = .1307 CD = .0062 CM = -.0079

Figure 13d



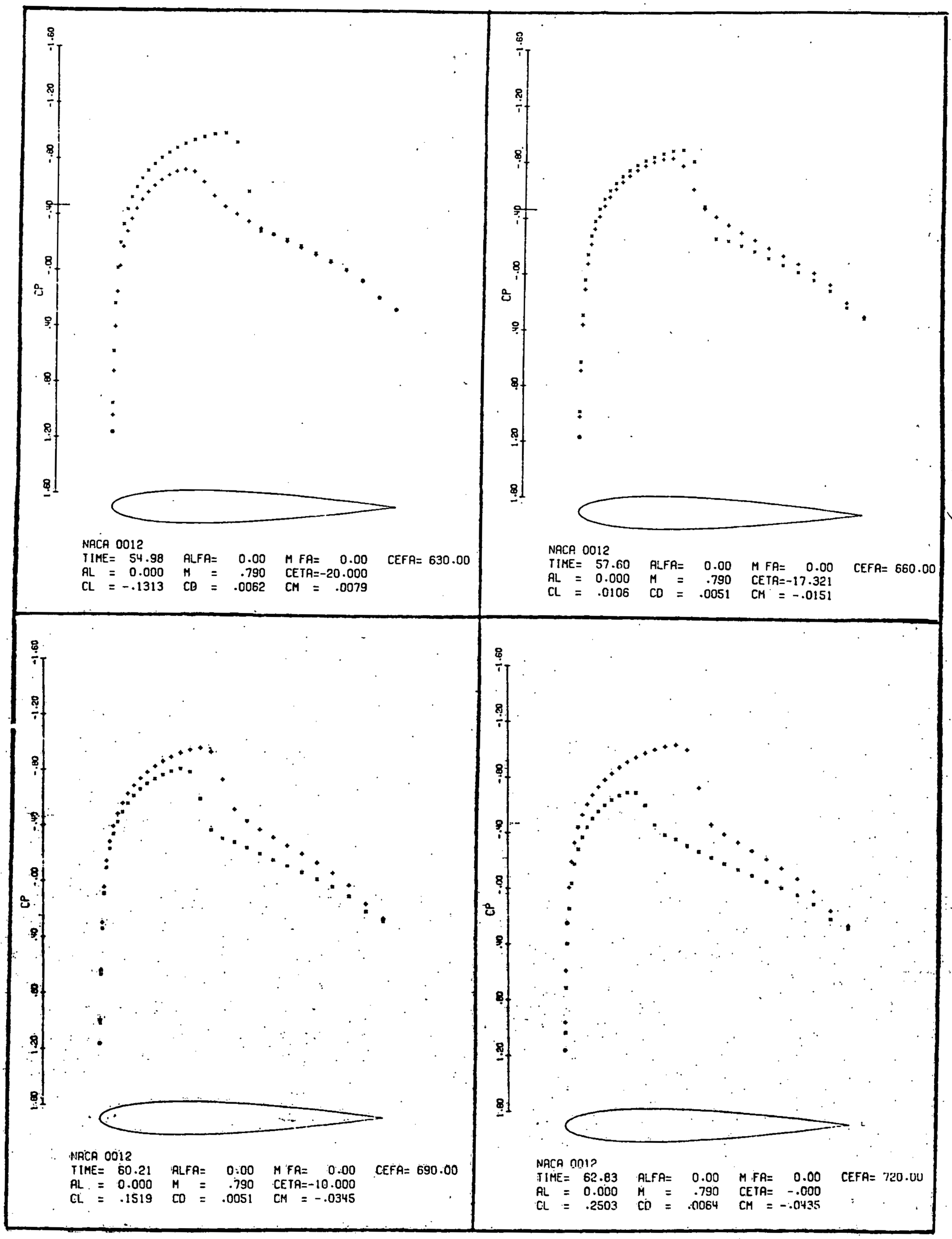
```
NACA 0012
TIME=  41.89   ALFA=   0.00   M FA=   0.00   CEFA= 480.00
AL   =  0.000   M    =   .790   CETA= 17.321
CL   = -.0113   CD   =  .0051   CM   =  .0151
```

```
NACA 0012
TIME=  44.51   ALFA=   0.00   M FA=   0.00   CEFA= 510.00
AL   =  0.000   M    =   .790   CETA= 10.000
CL   = -.1528   CD   =  .0051   CM   =  .0346
```

```
NACA 0012
TIME=  47.12   ALFA=   0.00   M FA=   0.00   CEFA= 540.00
AL   =  0.000   M    =   .790   CETA=   .000
CL   = -.2512   CD   =  .0064   CM   =  .0436
```

```
NACA 0012
TIME=  52.36   ALFA=   0.00   M FA=   0.00   CEFA= 600.00
AL   =  0.000   M    =   .790   CETA=-17.321
CL   = -.2394   CD   =  .0074   CM   =  .0284
```

Figure 13e



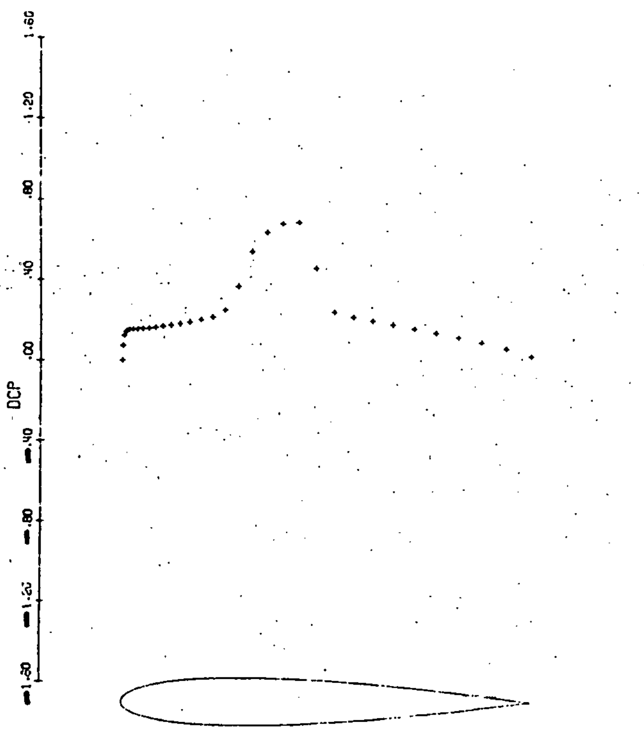
```
NACA 0012
TIME=  54.98   ALFA=    0.00   M FA=   0.00    CEFA= 630.00
AL   =  0.000   M    =    .790   CETA=-20.000
CL   = -.1313   CD   =   .0062   CM   =   .0079
```

```
NACA 0012
TIME=  57.60   ALFA=    0.00   M FA=   0.00    CEFA= 660.00
AL   =  0.000   M    =    .790   CETA=-17.321
CL   =  .0106   CD   =   .0051   CM   = -.0151
```

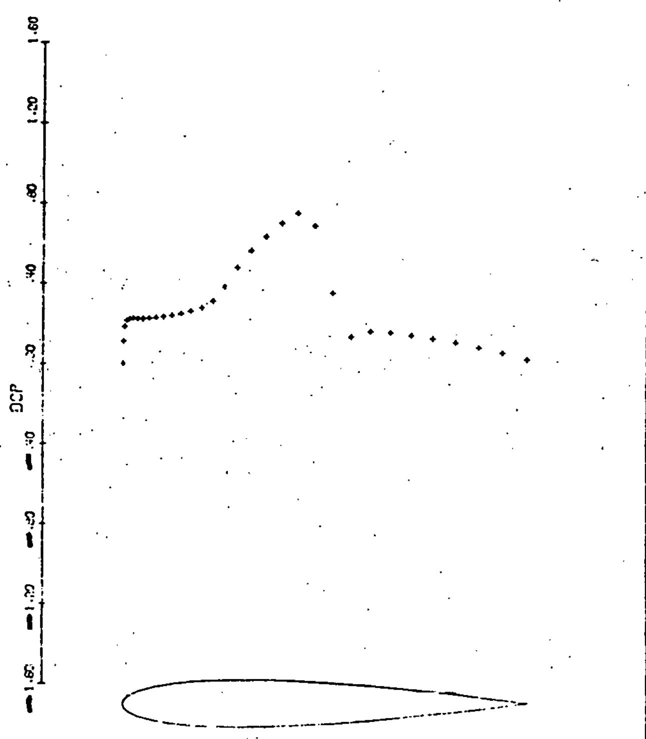
```
NACA 0012
TIME=  60.21   ALFA=    0.00   M FA=   0.00    CEFA= 690.00
AL   =  0.000   M    =    .790   CETA=-10.000
CL   =  .1519   CD   =   .0051   CM   = -.0345
```

```
NACA 0012
TIME=  62.83   ALFA=    0.00   M FA=   0.00    CEFA= 720.00
AL   =  0.000   M    =    .790   CETA=  -.000
CL   =  .2503   CD   =   .0064   CM   = -.0435
```

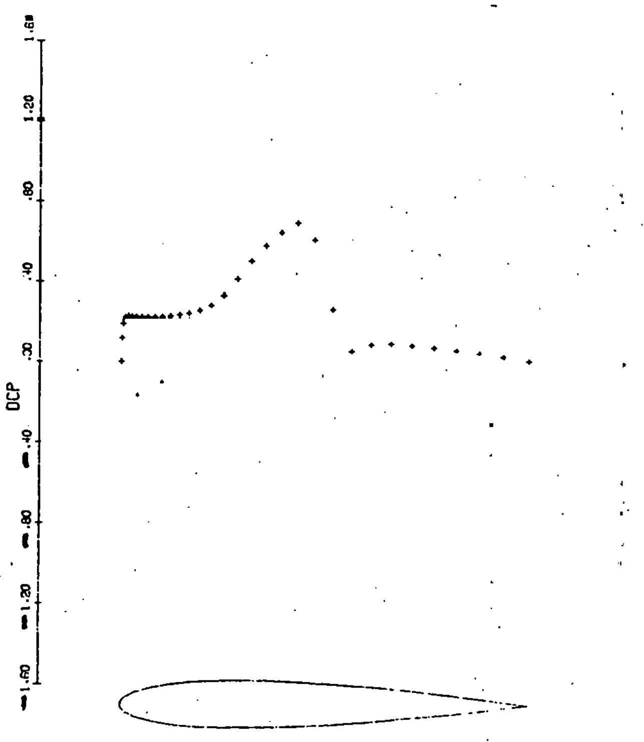
Figure 13f



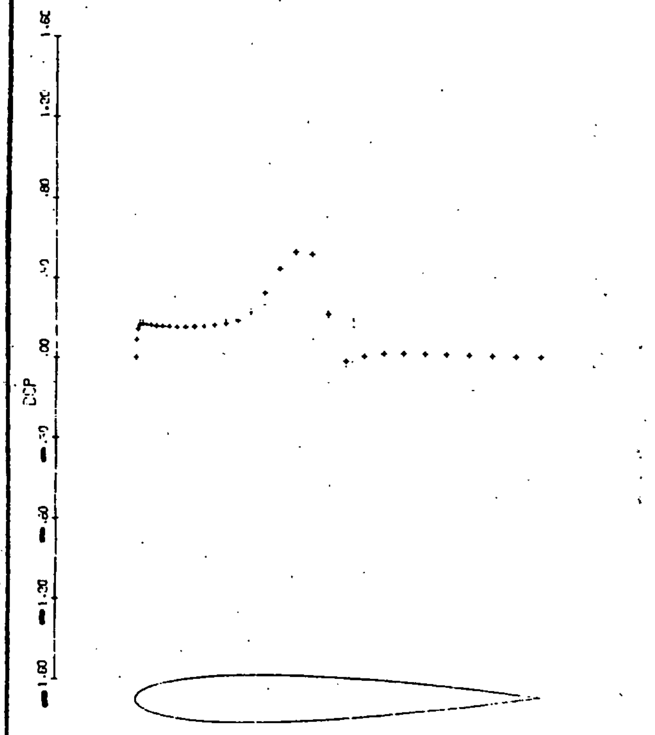
```
NAÇA 0012
TIME=  31.42   ALFA=   0.00   M FA=   0.00   CEFA= 360.00
AL   =  0.000   M    =   .790   CETA=  -.000
CL   =  .2492   CD   =  .0064   CM   = -.0435
```



```
NACA 0012
TIME=  34.03   ALFA=   0.00   M FA=   0.00   CEFA= 390.00
AL   =  0.000   M    =   .790   CETA= 10.000
CL   =  .2816   CD   =  .0076   CM   = -.0411
```

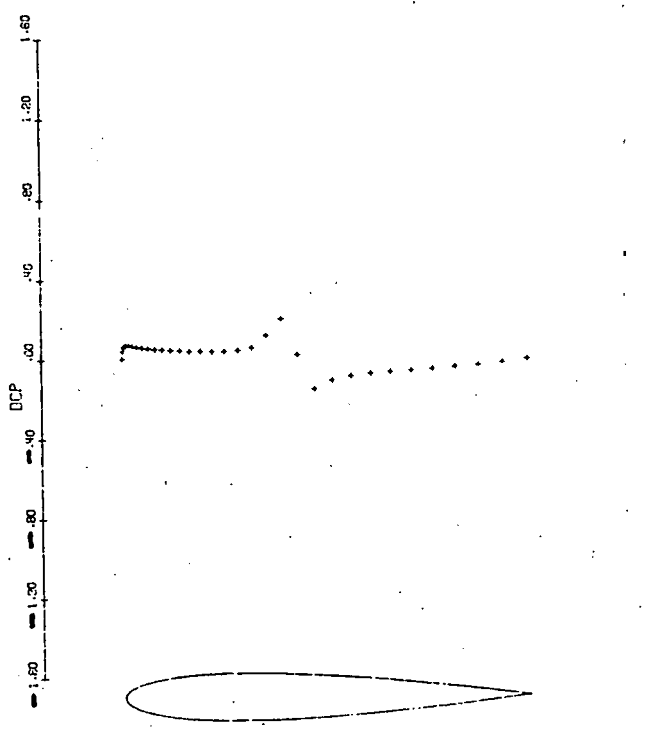


```
NACA 0012
TIME=  36.65   ALFA=   0.00   M FA=   0.00   CEFA= 420.00
AL   =  0.000   M    =   .790   CETA= 17.321
CL   =  .2386   CD   =  .0073   CM   = -.0263
```

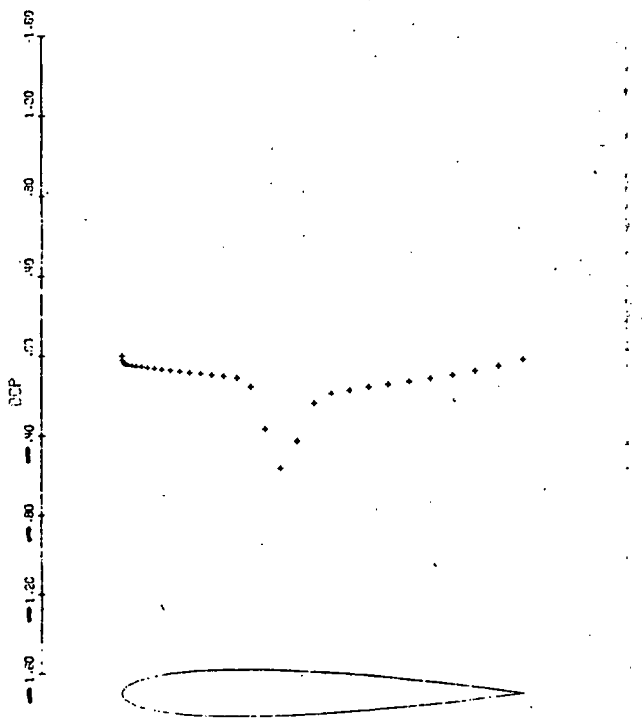


```
NACA 0012
TIME=  39.27   ALFA=   0.00   M FA=   0.00   CEFA= 450.00
AL   =  0.000   M    =   .790   CETA= 20.000
CL   =  .1307   CD   =  .0062   CM   = -.0079
```

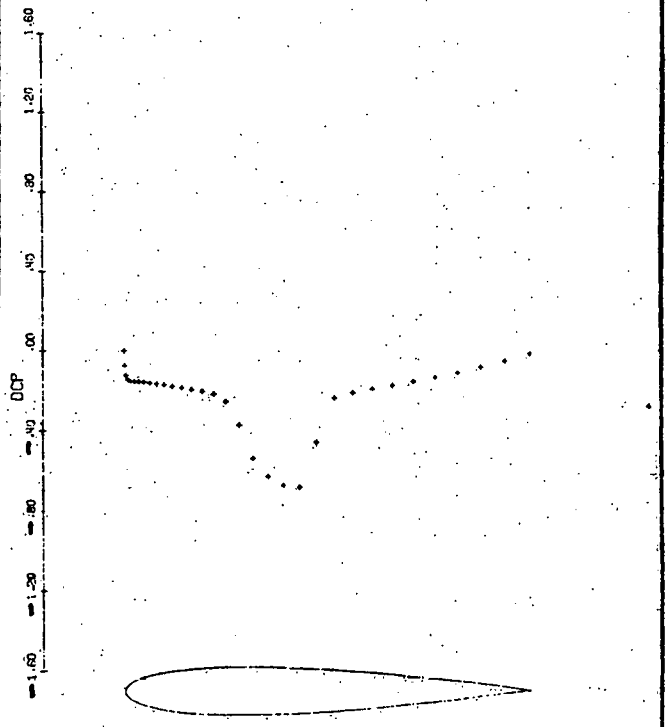
Figure 13g



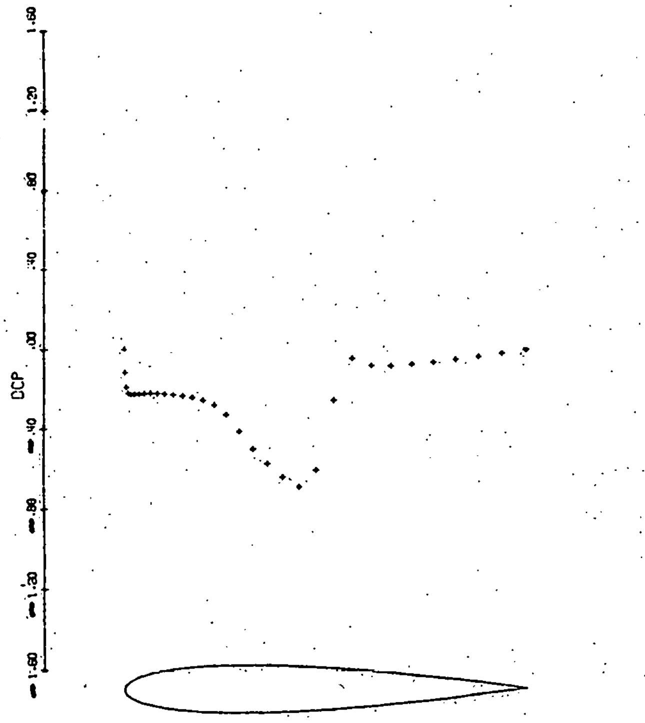
```
NACA 0012
TIME=  41.89   ALFA=   0.00   M FA=   0.00   CEFA= 480.00
AL  =  0.000   M   =   .790   CETA= 17.321
CL  = -.0113   CD  =  .0051   CM  =  .0151
```



```
NACA 0012
TIME=  47.51   ALFA=   0.00   M FA=   0.00   CEFA= 510.00
AL  =  0.000   M   =   .790   CETA= 10.000
CL  = -.1528   CD  =  .0051   CM  =  .0346
```

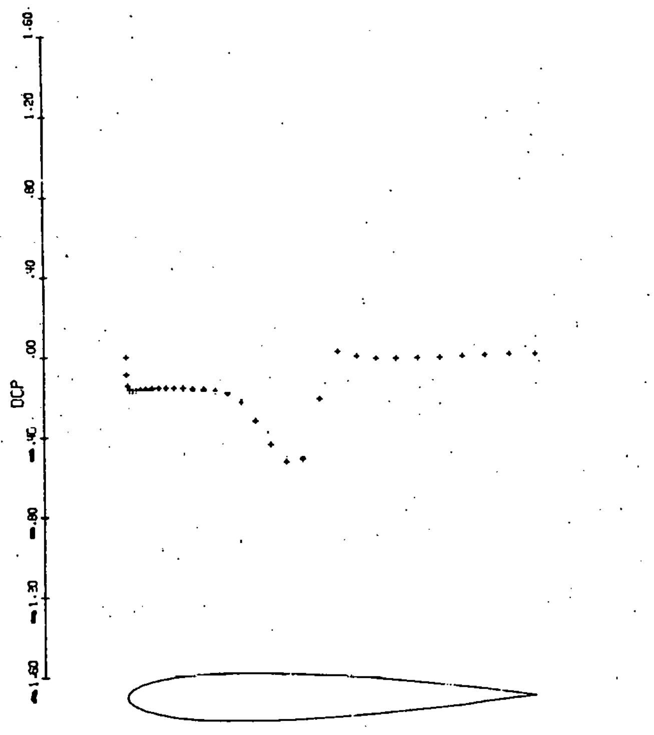


```
NACA 0012
TIME=  47.12   ALFA=   0.00   M FA=   0.00   CEFA= 540.00
AL  =  0.000   M   =   .790   CETA=   .000
CL  = -.2512   CD  =  .0064   CM  =  .0436
```

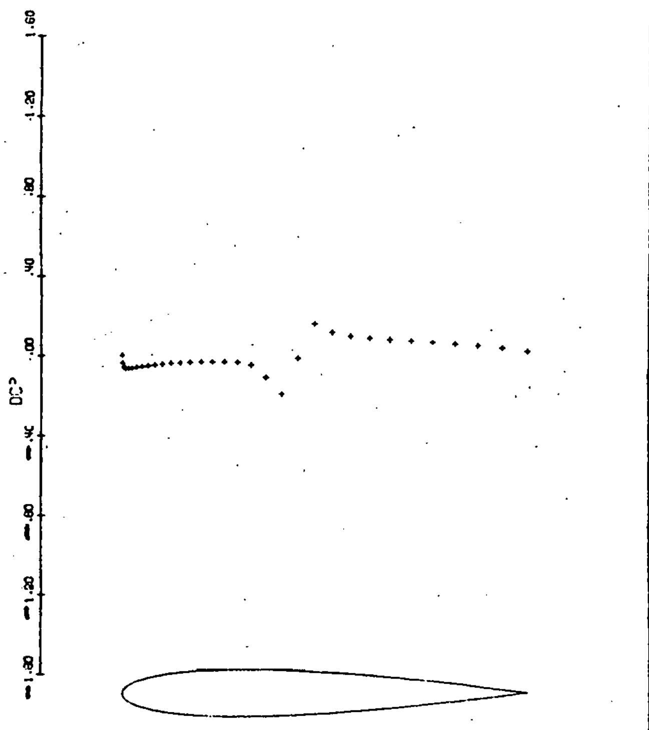


```
NACA 0012
TIME=  52.36   ALFA=   0.00   M FA=   0.00   CEFA= 600.00
AL  =  0.000   M   =   .790   CETA=-17.321
CL  = -.2394   CD  =  .0074   CM  =  .0284
```

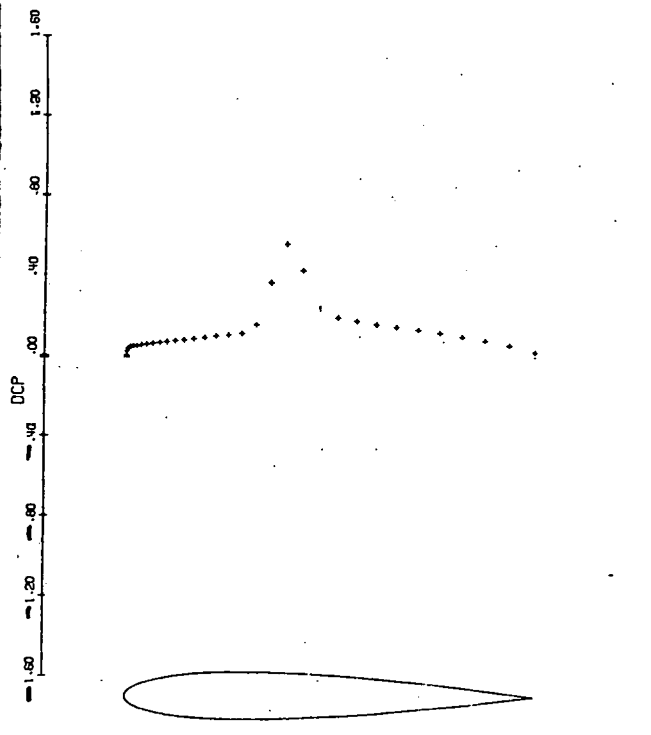
Figure 13h



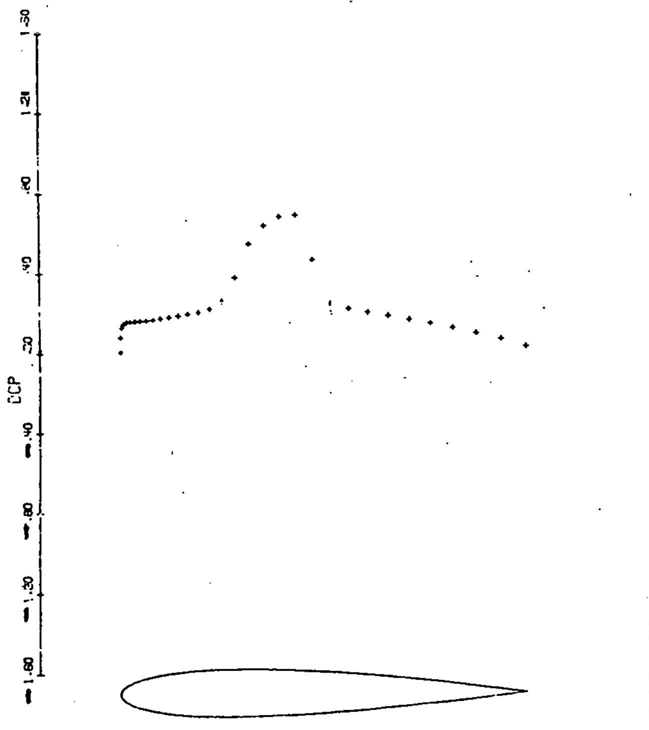
```
NACA 0012
TIME=  54.98   ALFA=   0.00   M FA=   0.00   CEFA= 630.00
AL   =  0.000  M    =   .790  CETA=-20.000
CL   = -.1313  CD   =  .0062  CM   =  .0079
```



```
NACA 0012
TIME=  57.60   ALFA=   0.00   M FA=   0.00   CEFA= 660.00
AL   =  0.000  M    =   .790  CETA=-17.321
CL   =  .0106  CD   =  .0051  CM   = -.0151
```

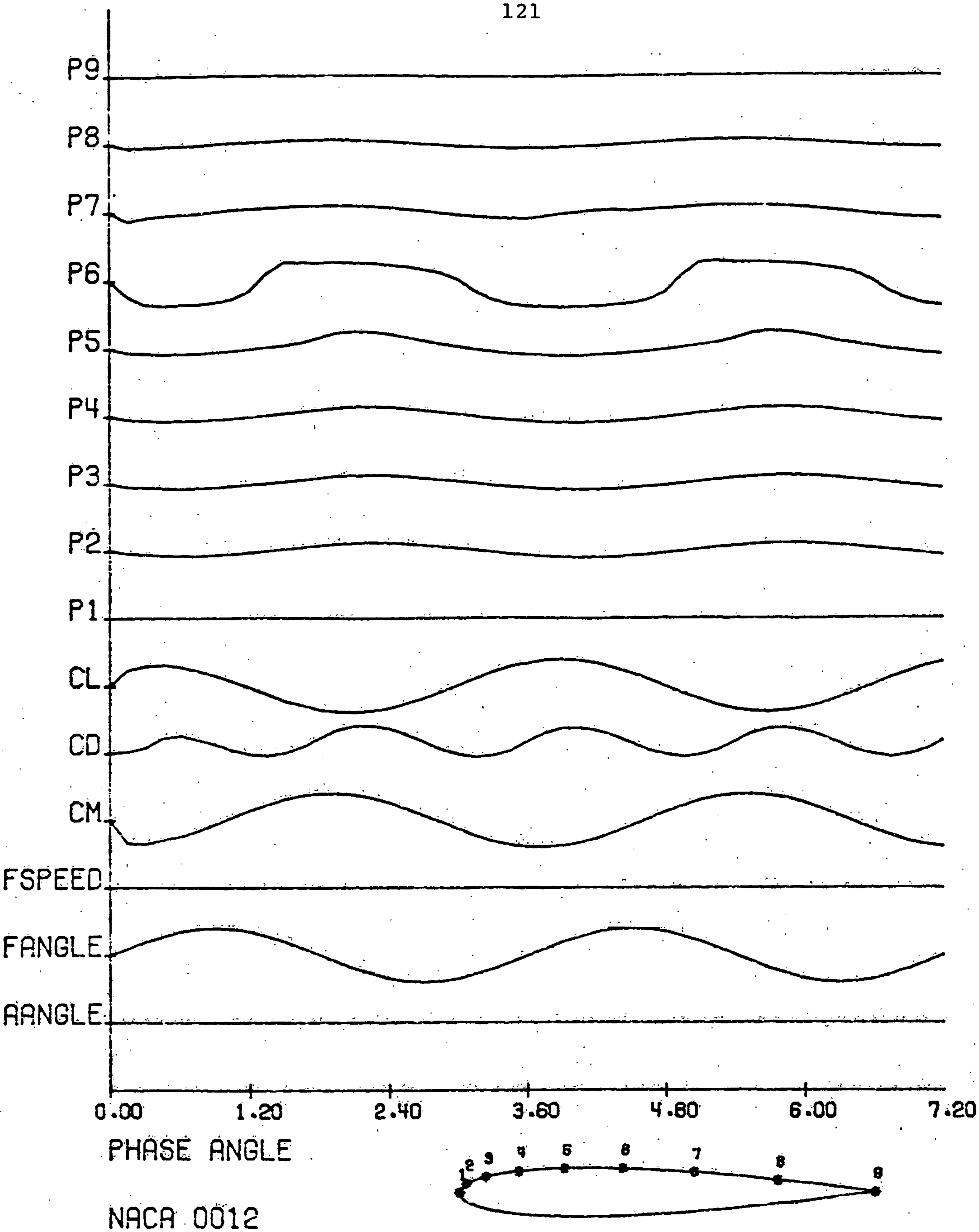


```
NACA 0012
TIME=  60.21   ALFA=   0.00   M FA=   0.00   CEFA= 690.00
AL   =  0.000  M    =   .790  CETA=-10.000
CL   =  .1519  CD   =  .0051  CM   = -.0345
```



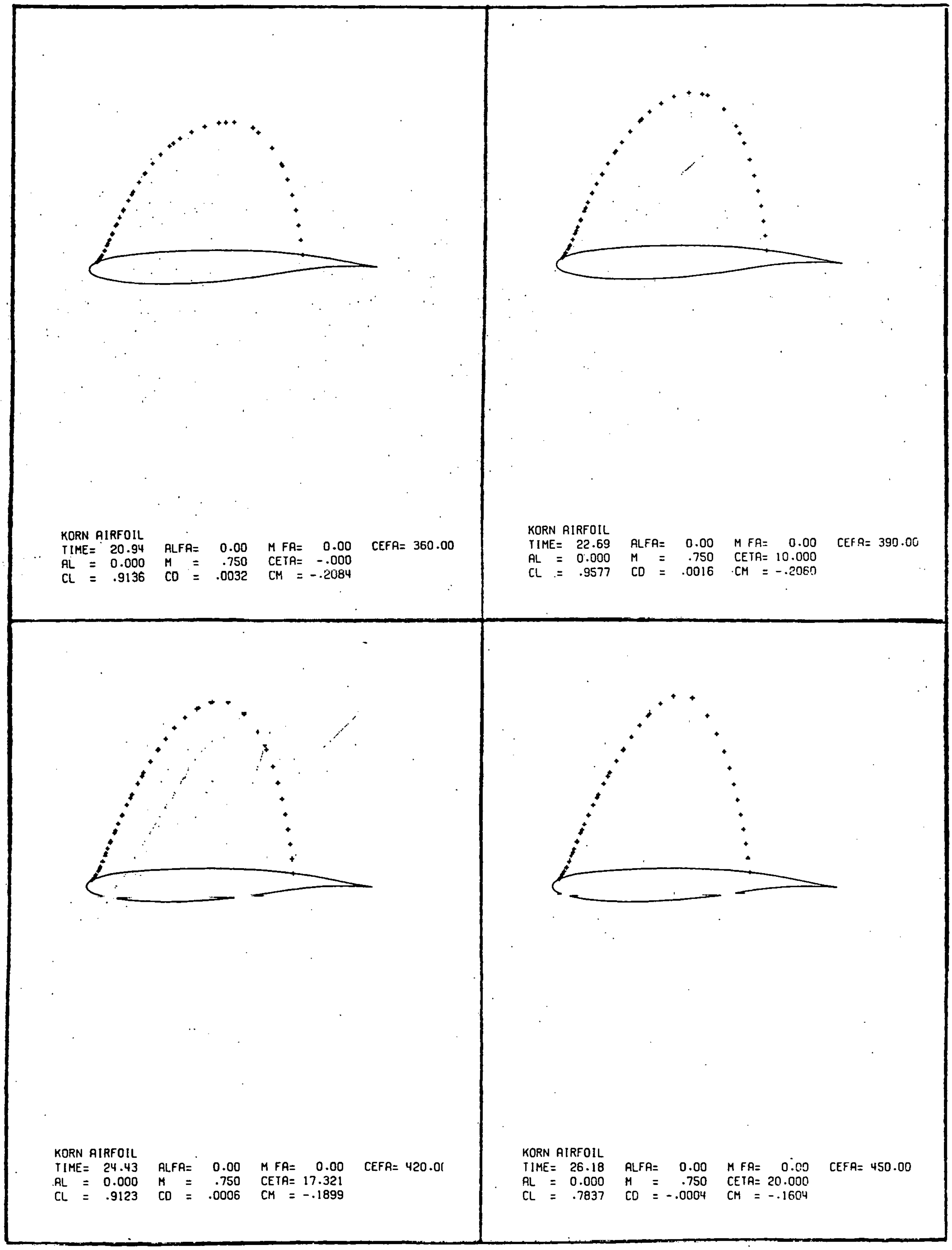
```
NACA 0012
TIME=  62.83   ALFA=   0.00   M FA=   0.00   CEFA= 720.00
AL   =  0.000  M    =   .790  CETA=  -.000
CL   =  .2503  CD   =  .0064  CM   = -.0435
```

Figure 13i



NACA 0012
UNSTEADY TRACES OF AIRFOIL IN SINUSOIDAL RIGID BODY MOTIO
MEAN ATTACK ANGLE= 0.00 AMP= 0.00 FREQ RATE= 0.00
MEAN FLIGHT SPEED= .79 AMP= 0.00 FREQ RATE= 0.00
MEAN FLIGHT ANGLE= 0.00 AMP=20.00 FREQ RATE= .20

Figure 13j



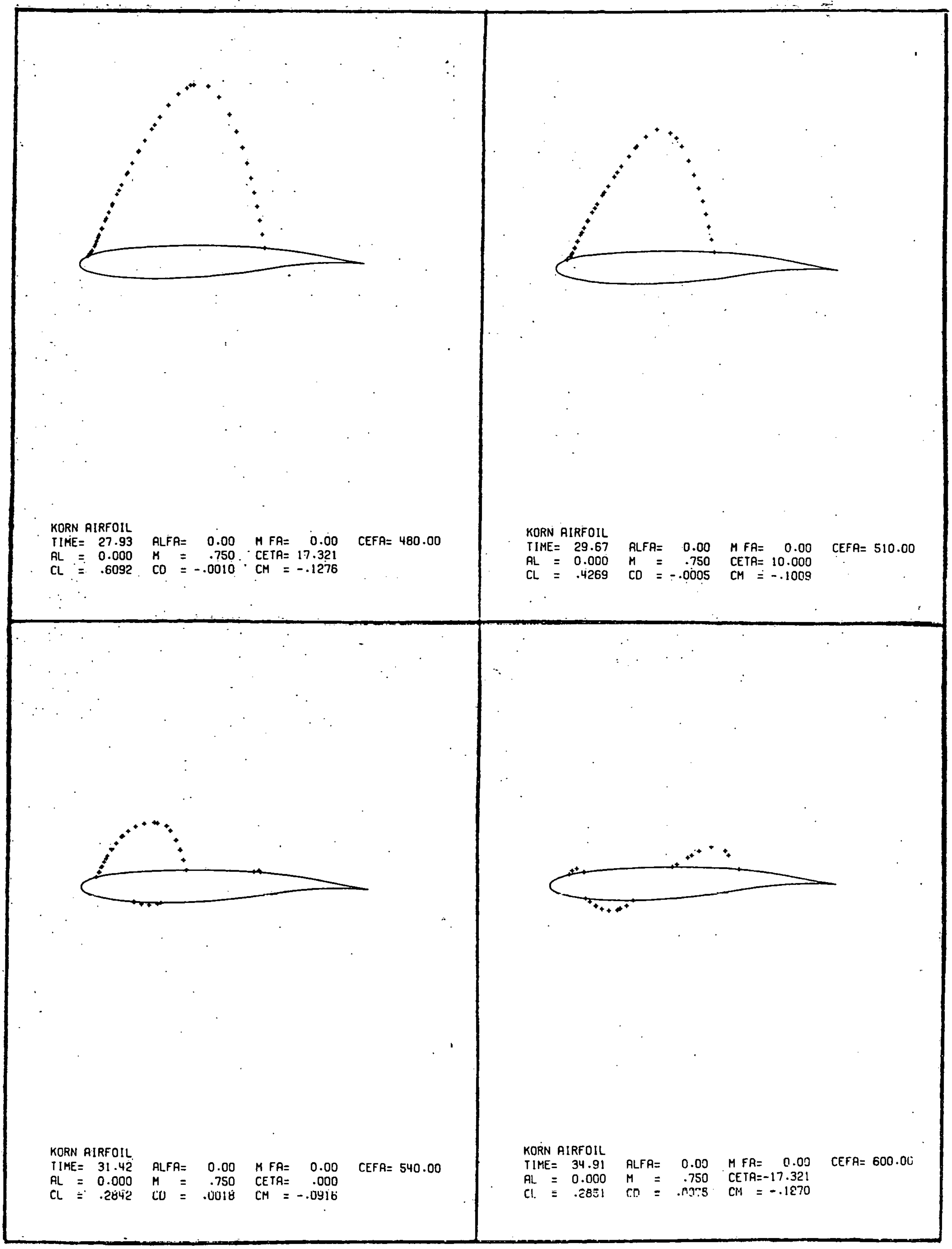
```
KORN AIRFOIL
TIME=  20.94   ALFA=   0.00   M FA=   0.00   CEFA= 360.00
AL  =  0.000   M   =   .750   CETA=  -.000
CL  =  .9136   CD  =  .0032   CM  = -.2084
```

```
KORN AIRFOIL
TIME=  22.69   ALFA=   0.00   M FA=   0.00   CEFA= 390.00
AL  =  0.000   M   =   .750   CETA= 10.000
CL  =  .9577   CD  =  .0016   CM  = -.2060
```

```
KORN AIRFOIL
TIME=  24.43   ALFA=   0.00   M FA=   0.00   CEFA= 420.0(
AL  =  0.000   M   =   .750   CETA= 17.321
CL  =  .9123   CD  =  .0006   CM  = -.1899
```

```
KORN AIRFOIL
TIME=  26.18   ALFA=   0.00   M FA=   0.00   CEFA= 450.00
AL  =  0.000   M   =   .750   CETA= 20.000
CL  =  .7837   CD  = -.0004   CM  = -.1604
```

Figure 14a



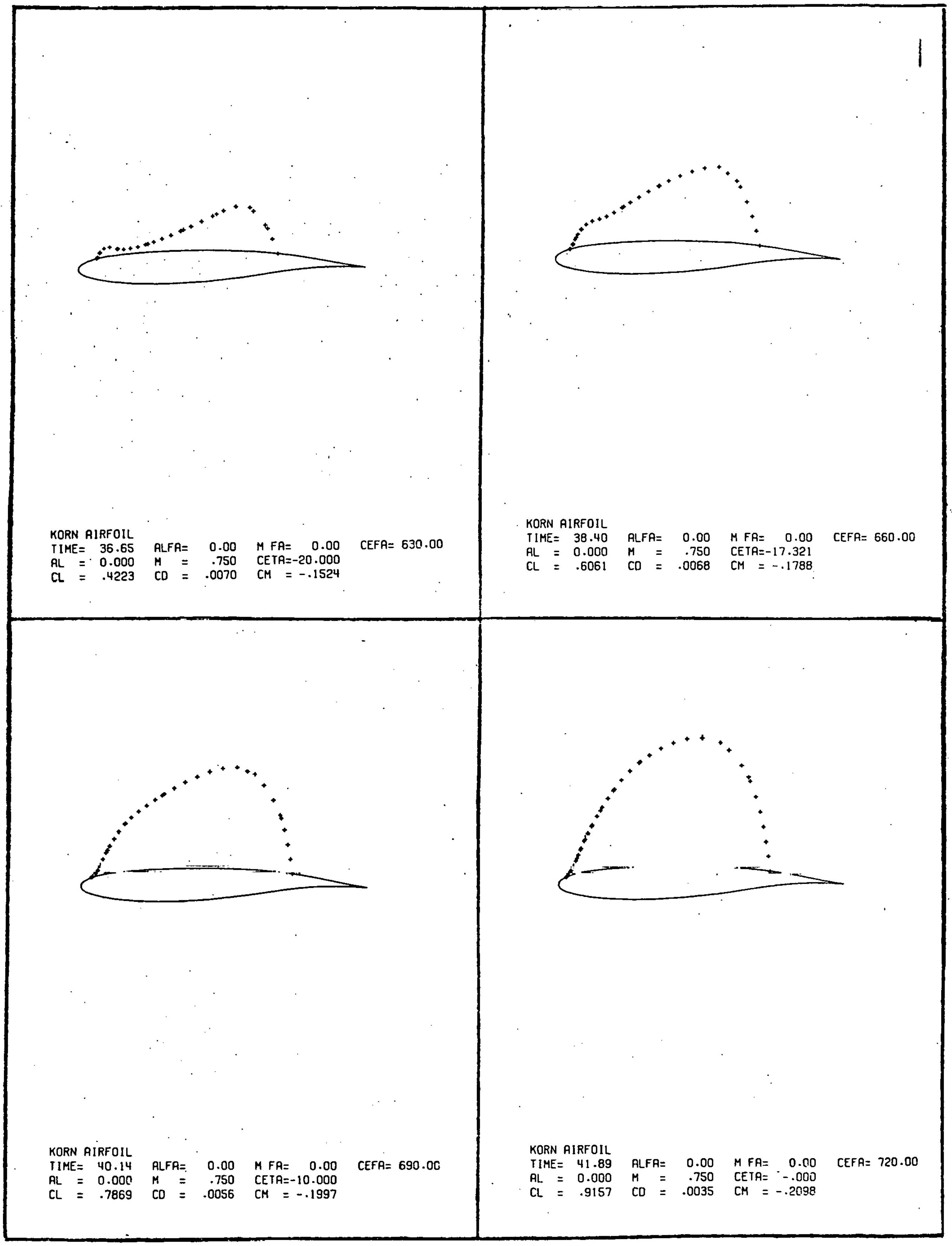
```
KORN AIRFOIL
TIME=  27.93   ALFA=   0.00   M FA=   0.00   CEFA= 480.00
AL  =  0.000   M   =   .750   CETA= 17.321
CL  =  .6092   CD  = -.0010   CM  = -.1276
```

```
KORN AIRFOIL
TIME=  29.67   ALFA=   0.00   M FA=   0.00   CEFA= 510.00
AL  =  0.000   M   =   .750   CETA= 10.000
CL  =  .4269   CD  = -.0005   CM  = -.1009
```

```
KORN AIRFOIL
TIME=  31.42   ALFA=   0.00   M FA=   0.00   CEFA= 540.00
AL  =  0.000   M   =   .750   CETA=   .000
CL  =  .2842   CD  =  .0018   CM  = -.0916
```

```
KORN AIRFOIL
TIME=  34.91   ALFA=   0.00   M FA=   0.00   CEFA= 600.00
AL  =  0.000   M   =   .750   CETA=-17.321
CL  =  .2851   CD  =  .0078   CM  = -.1270
```

Figure 14b



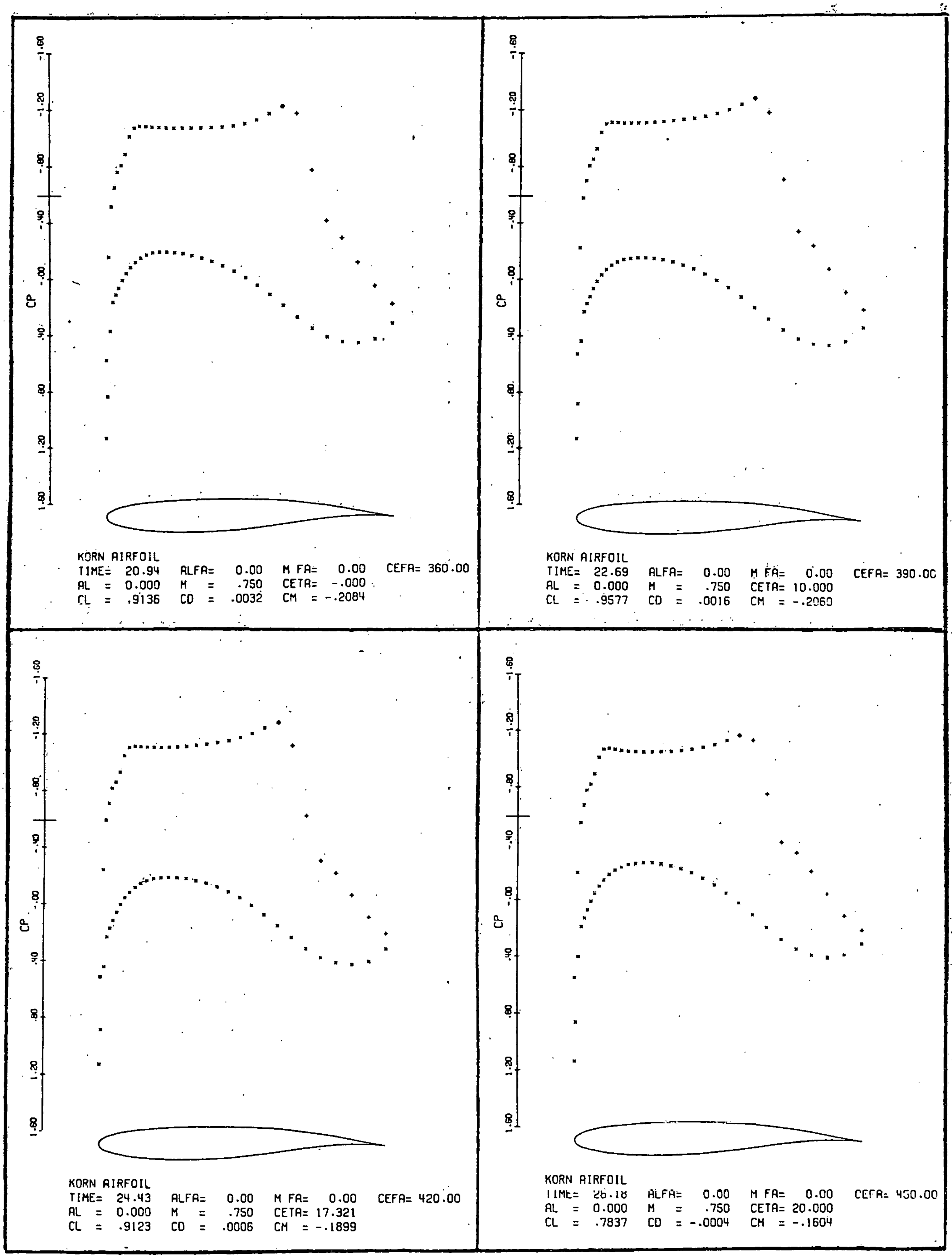
```
KORN AIRFOIL
TIME=  36.65   ALFA=    0.00   M FA=   0.00   CEFA= 630.00
AL  =  0.000   M    =   .750   CETA=-20.000
CL  =  .4223   CD   =  .0070   CM   = -.1524
```

```
KORN AIRFOIL
TIME=  38.40   ALFA=    0.00   M FA=   0.00   CEFA= 660.00
AL  =  0.000   M    =   .750   CETA=-17.321
CL  =  .6061   CD   =  .0068   CM   = -.1788
```

```
KORN AIRFOIL
TIME=  40.14   ALFA=    0.00   M FA=   0.00   CEFA= 690.00
AL  =  0.000   M    =   .750   CETA=-10.000
CL  =  .7869   CD   =  .0056   CM   = -.1997
```

```
KORN AIRFOIL
TIME=  41.89   ALFA=    0.00   M FA=   0.00   CEFA= 720.00
AL  =  0.000   M    =   .750   CETA=  -.000
CL  =  .9157   CD   =  .0035   CM   = -.2098
```

Figure 14c



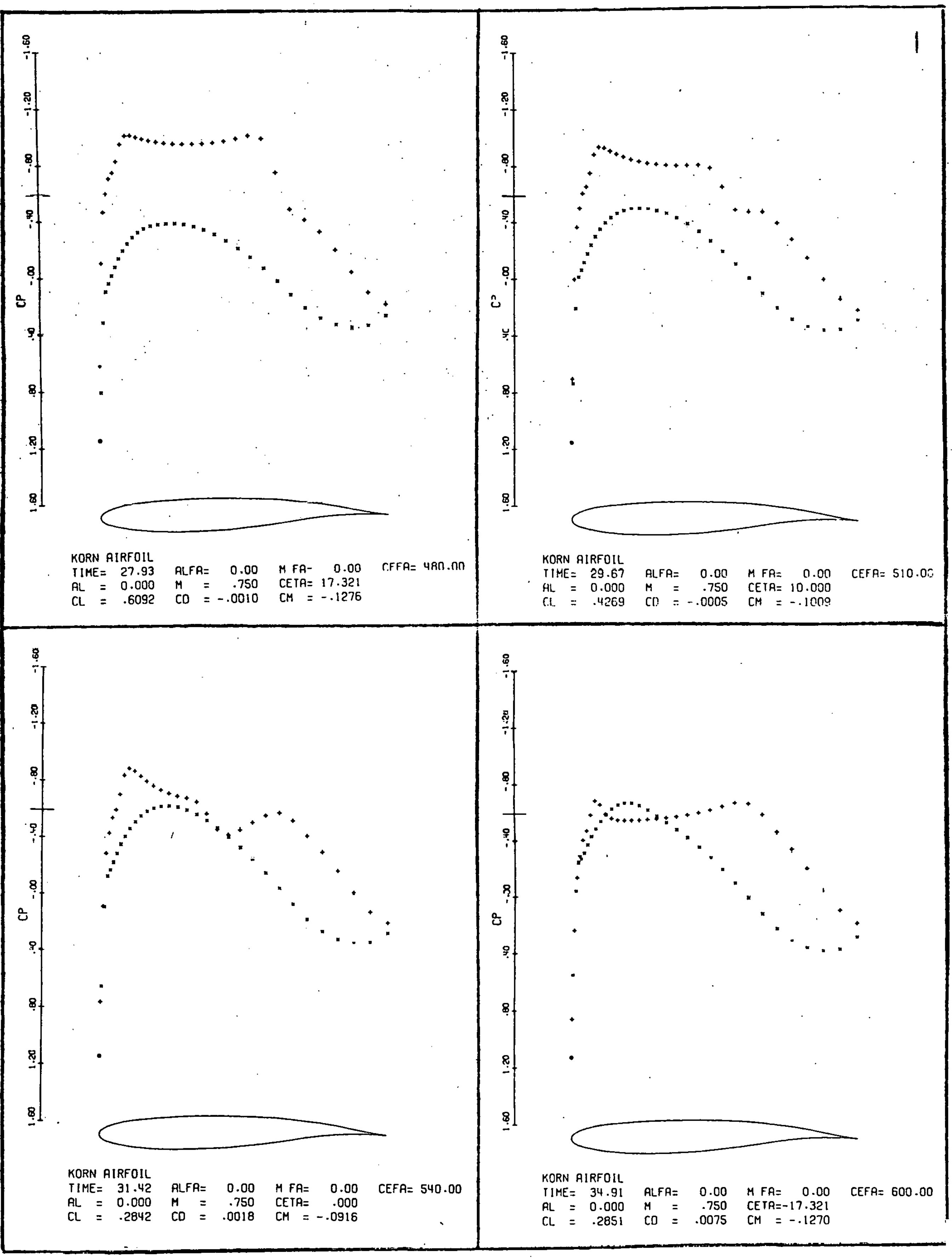
```
KORN AIRFOIL
TIME=  20.94   ALFA=    0.00   M FA=   0.00   CEFA= 360.00
AL  =  0.000   M    =   .750   CETA=  -.000
CL  =  .9136   CD   =  .0032   CM   = -.2084
```

```
KORN AIRFOIL
TIME=  22.69   ALFA=    0.00   M FA=   0.00   CEFA= 390.00
AL  =  0.000   M    =   .750   CETA= 10.000
CL  =  .9577   CD   =  .0016   CM   = -.2060
```

```
KORN AIRFOIL
TIME=  24.43   ALFA=    0.00   M FA=   0.00   CEFA= 420.00
AL  =  0.000   M    =   .750   CETA= 17.321
CL  =  .9123   CD   =  .0006   CM   = -.1899
```

```
KORN AIRFOIL
TIME=  26.18   ALFA=    0.00   M FA=   0.00   CEFA= 450.00
AL  =  0.000   M    =   .750   CETA= 20.000
CL  =  .7837   CD   = -.0004   CM   = -.1604
```

Figure 14d



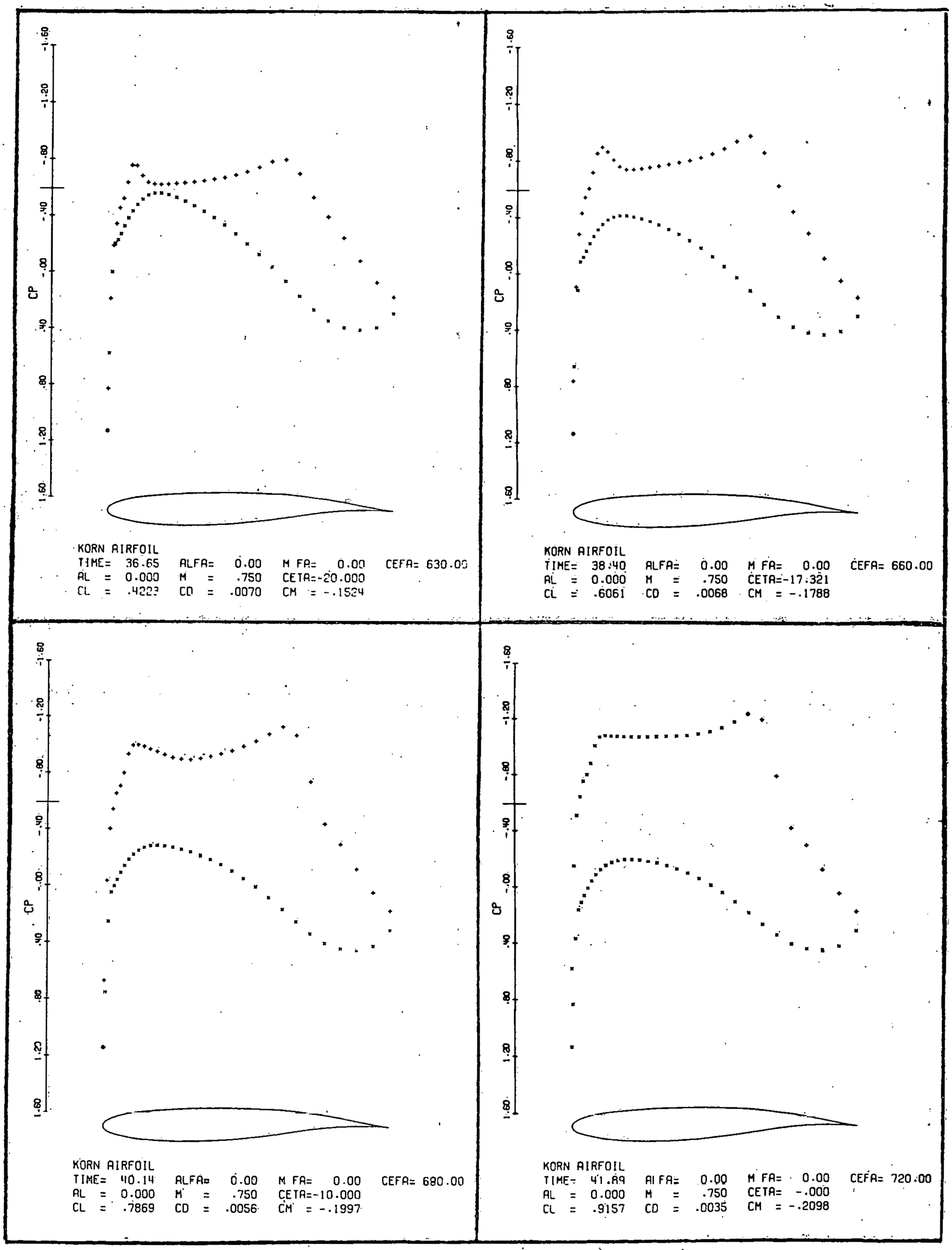
```
KORN AIRFOIL
TIME=  27.93   ALFA=   0.00   M FA-   0.00   CEFA= 480.00
AL  =  0.000   M   =   .750   CETA= 17.321
CL  =  .6092   CD  = -.0010   CM  = -.1276
```

```
KORN AIRFOIL
TIME=  29.67   ALFA=   0.00   M FA=   0.00   CEFA= 510.00
AL  =  0.000   M   =   .750   CETA= 10.000
CL  =  .4269   CD  = -.0005   CM  = -.1009
```

```
KORN AIRFOIL
TIME=  31.42   ALFA=   0.00   M FA=   0.00   CEFA= 540.00
AL  =  0.000   M   =   .750   CETA=    .000
CL  =  .2842   CD  =  .0018   CM  = -.0916
```

```
KORN AIRFOIL
TIME=  34.91   ALFA=   0.00   M FA=   0.00   CEFA= 600.00
AL  =  0.000   M   =   .750   CETA=-17.321
CL  =  .2851   CD  =  .0075   CM  = -.1270
```

Figure 14e



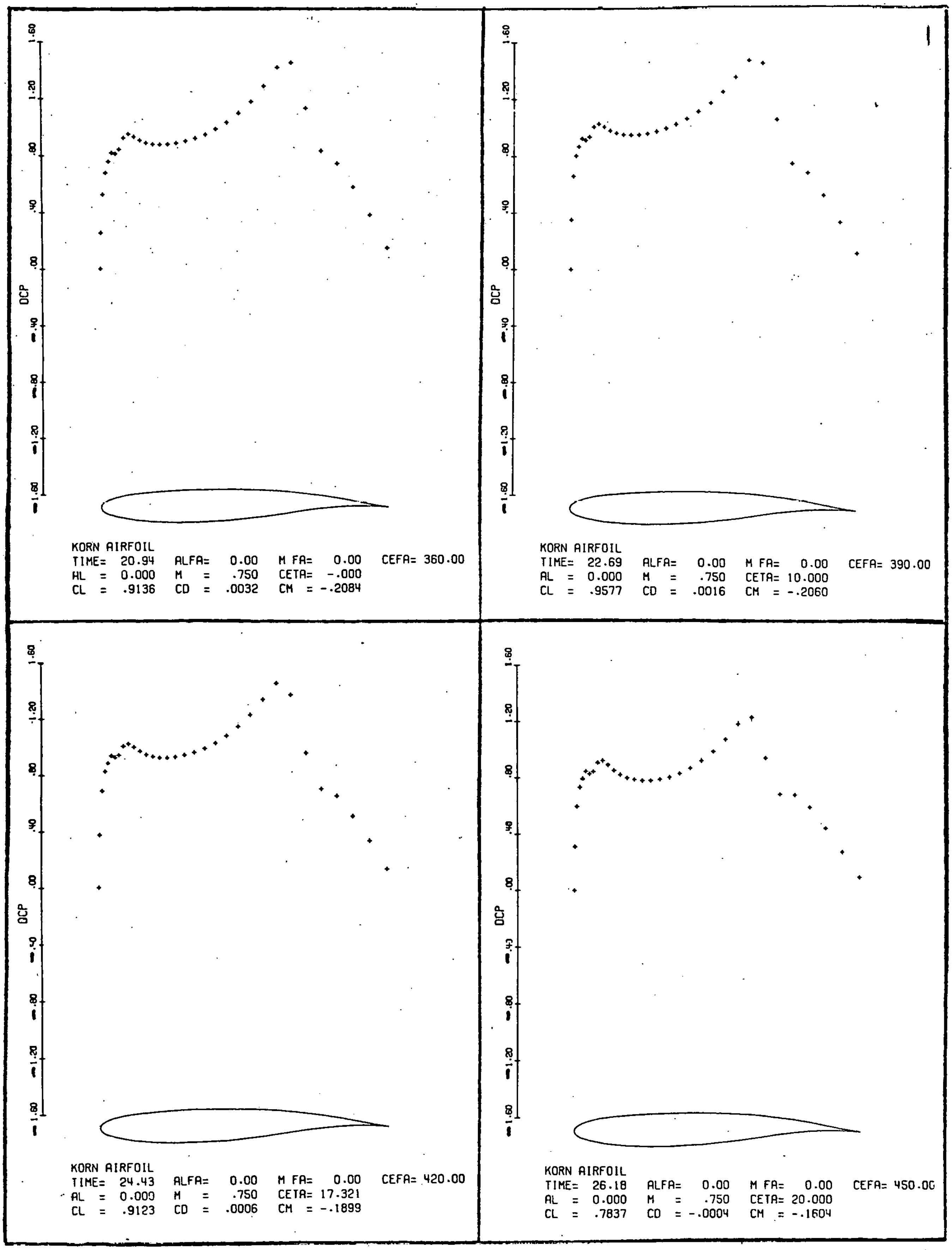
```
KORN AIRFOIL
TIME=  36.65   ALFA=   0.00   M FA=    0.00   CEFA= 630.00
AL  =  0.000   M    =  .750   CETA=-20.000
CL  =  .4223   CD   = .0070   CM   = -.1524
```

```
KORN AIRFOIL
TIME=  38.40   ALFA=   0.00   M FA=    0.00   CEFA= 660.00
AL  =  0.000   M    =  .750   CETA=-17.321
CL  =  .6061   CD   = .0068   CM   = -.1788
```

```
KORN AIRFOIL
TIME=  40.14   ALFA=   0.00   M FA=    0.00   CEFA= 680.00
AL  =  0.000   M    =  .750   CETA=-10.000
CL  =  .7869   CD   = .0056   CM   = -.1997
```

```
KORN AIRFOIL
TIME=  41.89   ALFA=   0.00   M FA=    0.00   CEFA= 720.00
AL  =  0.000   M    =  .750   CETA=  -.000
CL  =  .9157   CD   = .0035   CM   = -.2098
```

Figure 14f



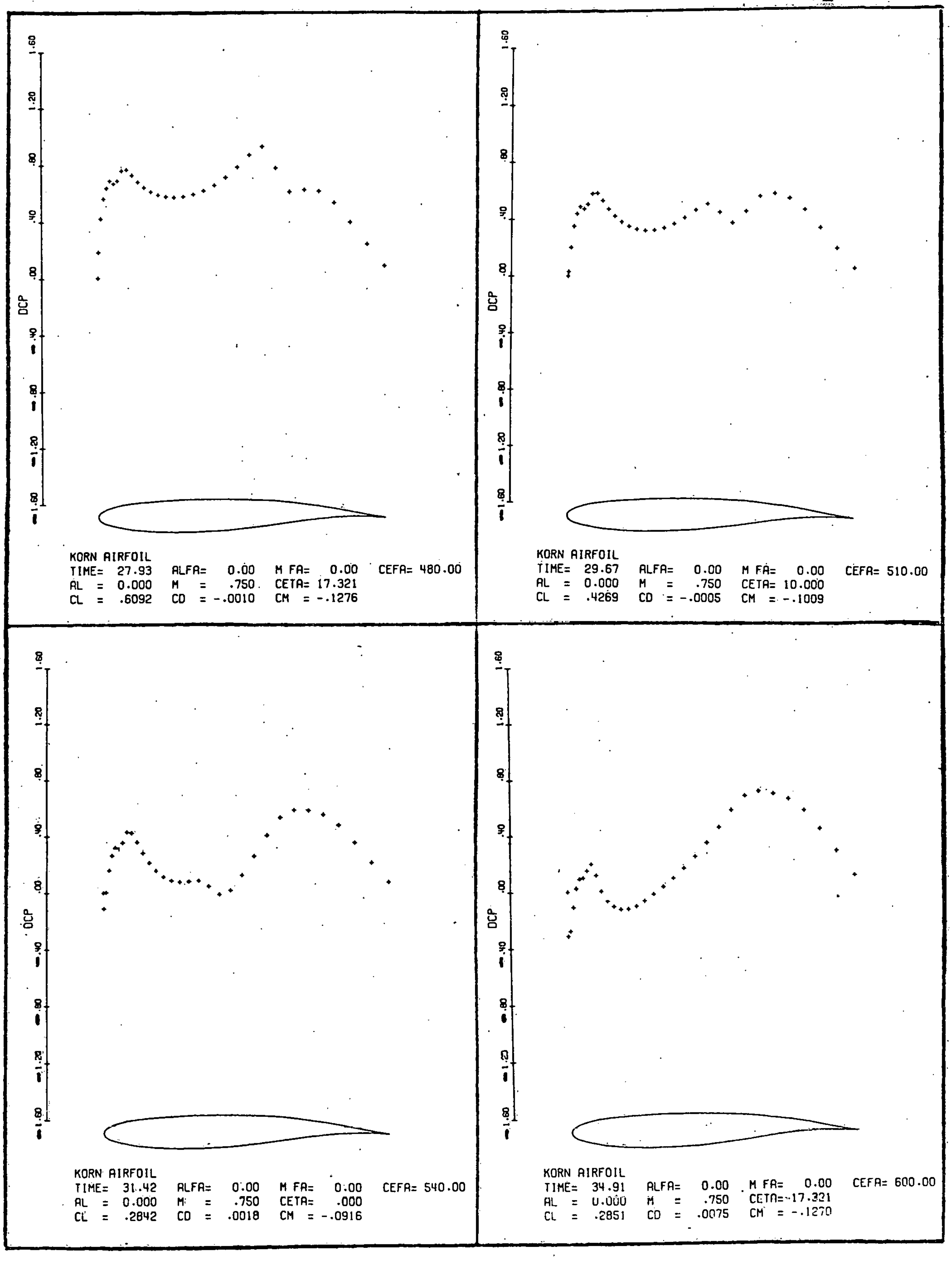
```
KORN AIRFOIL
TIME=  20.94   ALFA=   0.00   M FA=   0.00   CEFA= 360.00
AL   =  0.000  M    =   .750  CETA=  -.000
CL   =  .9136  CD   =  .0032  CM   = -.2084
```

```
KORN AIRFOIL
TIME=  22.69   ALFA=   0.00   M FA=   0.00   CEFA= 390.00
AL   =  0.000  M    =   .750  CETA= 10.000
CL   =  .9577  CD   =  .0016  CM   = -.2060
```

```
KORN AIRFOIL
TIME=  24.43   ALFA=   0.00   M FA=   0.00   CEFA= 420.00
AL   =  0.000  M    =   .750  CETA= 17.321
CL   =  .9123  CD   =  .0006  CM   = -.1899
```

```
KORN AIRFOIL
TIME=  26.18   ALFA=   0.00   M FA=   0.00   CEFA= 450.00
AL   =  0.000  M    =   .750  CETA= 20.000
CL   =  .7837  CD   = -.0004  CM   = -.1604
```

Figure 14g



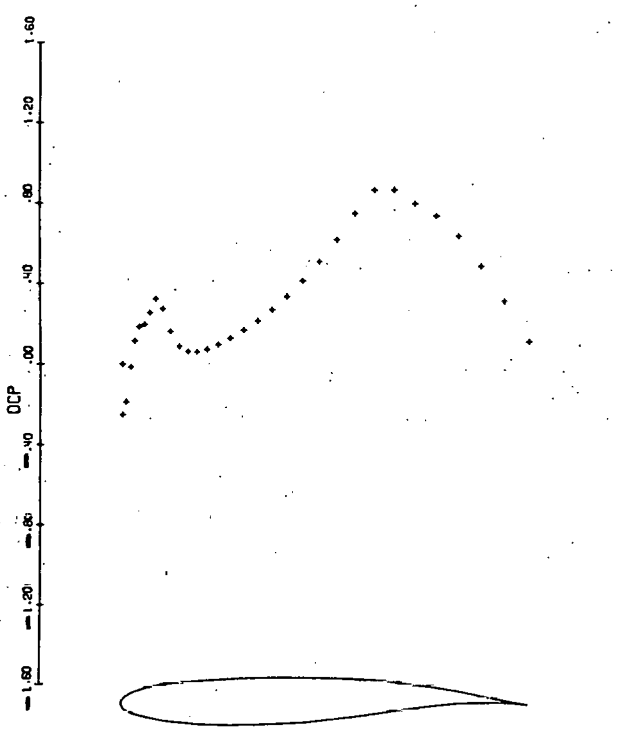
```
KORN AIRFOIL
TIME=  27.93    ALFA=   0.00    M FA=   0.00    CEFA= 480.00
AL  =  0.000    M    =   .750   CETA= 17.321
CL  =  .6092    CD   = -.0010   CM   = -.1276
```

```
KORN AIRFOIL
TIME=  29.67    ALFA=   0.00    M FA=   0.00    CEFA= 510.00
AL  =  0.000    M    =   .750   CETA= 10.000
CL  =  .4269    CD   = -.0005   CM   = -.1009
```

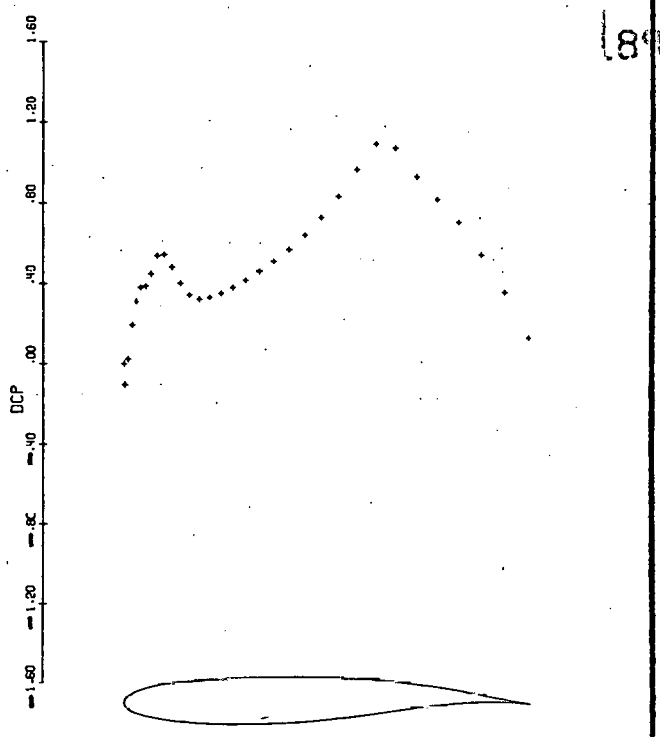
```
KORN AIRFOIL
TIME=  31.42    ALFA=   0.00    M FA=   0.00    CEFA= 540.00
AL  =  0.000    M    =   .750   CETA=   .000
CL  =  .2842    CD   =  .0018   CM   = -.0916
```

```
KORN AIRFOIL
TIME=  34.91    ALFA=   0.00    M FA=   0.00    CEFA= 600.00
AL  =  0.000    M    =   .750   CETA=-17.321
CL  =  .2851    CD   =  .0075   CM   = -.1270
```

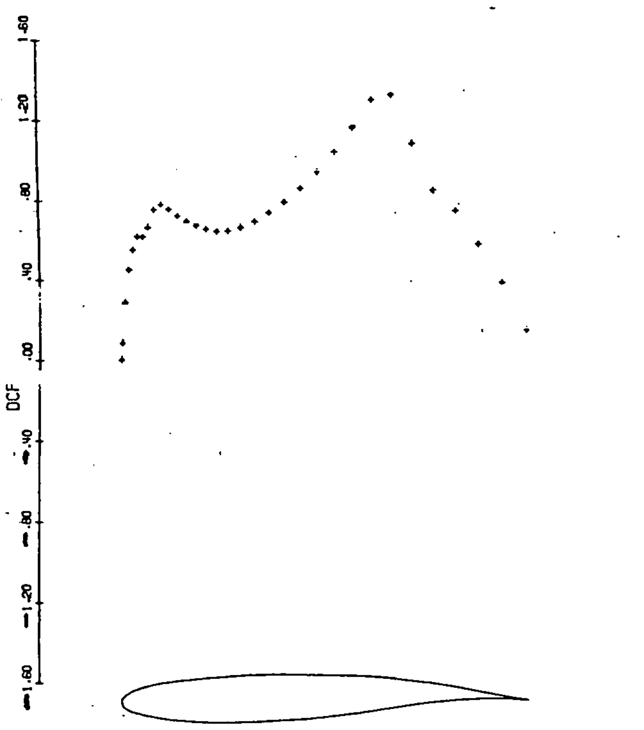
Figure 14h



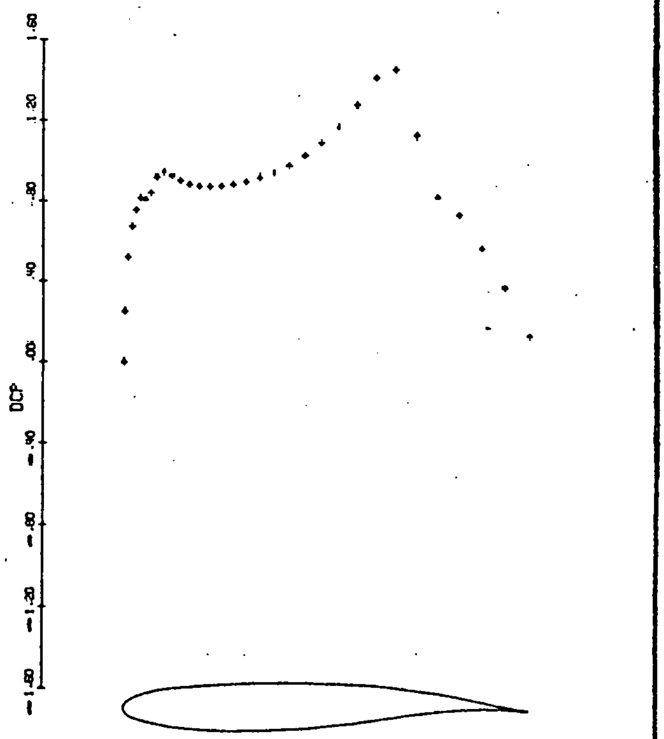
```
KORN AIRFOIL
TIME=  36.65   ALFA=   0.00   M FA=   0.00   CEFA= 630.00
AL  =  0.000   M   =   .750   CETA=-20.000
CL  =  .4223   CD  =  .0070   CM  = -.1524
```



```
KORN AIRFOIL
TIME=  38.40   ALFA=   0.00   M FA=   0.00   CEFA= 660.00
AL  =  0.000   M   =   .750   CETA=-17.321
CL  =  .6061   CD  =  .0068   CM  = -.1788
```

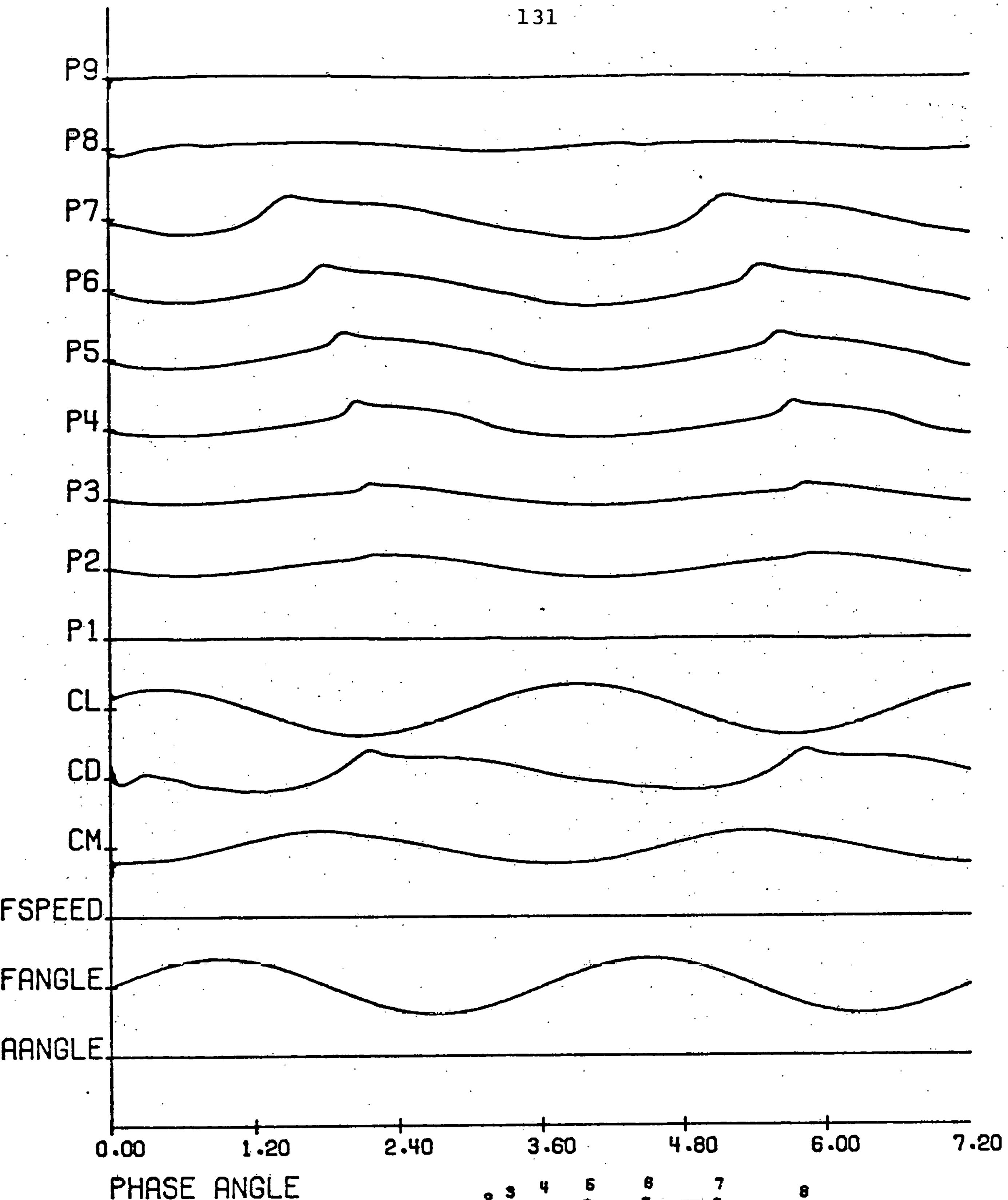


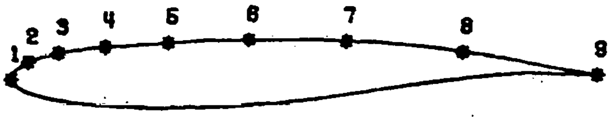
```
KORN AIRFOIL
TIME=  40.14   ALFA=   0.00   M FA=   0.00   CEFA= 690.00
AL  =  0.000   M   =   .750   CETA=-10.000
CL  =  .7869   CD  =  .0056   CM  = -.1997
```



```
KORN AIRFOIL
TIME=  41.89   ALFA=   0.00   M FA=   0.00   CEFA= 720.00
AL  =  0.000   M   =   .750   CETA=  -.000
CL  =  .9157   CD  =  .0035   CM  = -.2008
```

Figure 14i





KORN AIRFOIL
UNSTEADY TRACES OF AIRFOIL IN SINUSOIDAL RIGID BODY MOTION
MEAN ATTACK ANGLE= 0.00 AMP= 0.00 FREQ RATE= 0.00
MEAN FLIGHT SPEED= .75 AMP= 0.00 FREQ RATE= 0.00
MEAN FLIGHT ANGLE= 0.00 AMP=20.00 FREQ RATE= .30
Figure 14j

APPENDIX

A. A 5-Diagonal Matrix Solver

Here, we present a method of solving a 5-diagonal matrix problem. Suppose Ax = y is solved for x, where x and y are n×1 column vectors and A is of the form

$$\begin{pmatrix}
a_1 & c_1 & e_1 & & & & & \\
b_2 & a_2 & c_2 & e_2 & & & & \\
d_3 & b_3 & a_3 & c_3 & e_3 & & & \\
 & \cdot & \cdot & \cdot & \cdot & \cdot & & \\
 & & \cdot & \cdot & \cdot & \cdot & \cdot & \\
 & & & \cdot & \cdot & \cdot & \cdot & \cdot \\
 & & & & \cdot & \cdot & \cdot & \cdot & \cdot \\
 & & & & & d_{n-2} & b_{n-2} & a_{n-2} & c_{n-2} & e_{n-2} \\
 & & & & & & d_{n-1} & b_{n-1} & a_{n-1} & c_{n-1} \\
 & & & & & & & d_n & b_n & a_n
\end{pmatrix}$$

Assume the matrix can be factored in the tridiagonal form

$$A = LU = \begin{pmatrix}
\alpha_1 & & & & & \\
\beta_2 & \alpha_2 & & & & \\
\delta_3 & \beta_3 & \alpha_3 & & & \\
 & \cdot & \cdot & \cdot & & \\
 & & \cdot & \cdot & \cdot & \\
 & & & \cdot & \cdot & \cdot \\
 & & & \delta_n & \beta_n & \alpha_n
\end{pmatrix}
\begin{pmatrix}
1 & \gamma_1 & \varepsilon_1 & & & \\
 & 1 & \gamma_2 & \varepsilon_2 & & \\
 & & \cdot & \cdot & \cdot & \\
 & & & \cdot & \cdot & \cdot \\
 & & & 1 & \gamma_{n-2} & \varepsilon_{n-2} \\
 & & & & 1 & \gamma_{n-1} \\
 & & & & & 1
\end{pmatrix}$$

Then we find

$$(1)\quad \begin{aligned} \delta_i &= d_i \\ \delta_i\gamma_{i-2} + \beta_i &= b_i \\ \delta_i\varepsilon_{i-2} + \beta_i\gamma_{i-1} + \alpha_i &= a_i \\ \beta_i\varepsilon_{i-1} + \alpha_i\gamma_i &= c_i \\ \alpha_i &= e_i \end{aligned}$$

for $i = 1,\ldots,n$ if the default values are set equal to zero. Namely, $b_1 = d_1 = d_2 = c_n = e_{n-1} = e_n = 0$, $\beta_1 = \delta_1 = \delta_2 = \gamma_n = \varepsilon_{n-1} = \varepsilon_n = 0$ and which can be solved as

$$(2)\quad \begin{aligned} \delta_i &= d_i \\ \beta_i &= b_i - \gamma_{i-2}\delta_i \\ \alpha_i &= a_i - \beta_i\gamma_{i-1} - \delta_i\varepsilon_{i-2} \\ \gamma_i &= (c_i - \beta_i\varepsilon_{i-1})/\alpha_i \\ \varepsilon_i &= e_i/\alpha_i \end{aligned}$$

for $i = 1,\ldots,n$, in ascending order if none of the α_i vanish. The intermediate step $Lg = y$ becomes

$$\delta_i g_{i-2} + \beta_i g_{i-1} + \alpha_i g_i = y_i \quad \text{for} \quad i = 1,\ldots,n,$$

we can solve this system recursively in ascending order. Namely,

$$(3)\quad g_i = (y_i - \beta_i g_{i-1} - \delta_i g_{i-2})/\alpha_i$$

for $i = 1,\ldots,n$, if the default values $g_{-1} = g_{-2} = 0$.

The final step $Ux = g$ is expressed by

$$x_i + \gamma_i x_{i+1} + \varepsilon_i x_{i+2} = g_i \quad \text{for} \quad i = 1,\ldots,n,$$

where $x_{n+1} = x_{n+2} = 0$. The system can be solved in descending order recursively as

$$x_i = g_i - \gamma_i x_{i+1} - \varepsilon_i x_{i+2} \quad \text{for} \quad i = n,\ldots,1.$$

We remark that the LU factored form is not unique; for example, L and U can be the following

$$L = \begin{pmatrix} 1 & & & & & \\ \beta_2 & 1 & & & & \\ \delta_3 & \beta_2 & 1 & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & \delta_n & \beta_n & 1 \end{pmatrix} \qquad U = \begin{pmatrix} \alpha_1 & \gamma_1 & \varepsilon_1 & & & & \\ & \alpha_2 & \gamma_2 & \varepsilon_2 & & & \\ & & & & \ddots & & \\ & & & & \alpha_{n-2} & \gamma_{n-2} & \varepsilon_{n-2} \\ & & & & & \alpha_{n-1} & \gamma_{n-1} \\ & & & & & & \alpha_n \end{pmatrix}$$

B. Computer Program UFL05

1. Operation of the Program

The sheared parabolic coordinates described in Section III,3 are introduced. The input parameters XSING and YSING determine the location of the singular point about which the square root transformation is made. It is important to choose these two parameters so that the unfolded profile does not have any sharp bumps. The mapped coordinates are printed so that this can be checked.

The difference scheme for the steady routine used to initialize the calculation is in fact the standard line relaxation method. Faster convergence is usually obtained by using horizontal relaxation, y-sweep, marching toward the body. The difference scheme for the unsteady routine conforms closely to the description in Section IV, 1. It is implemented in the computational domain described in Section III, 3 as first performing a y-sweep, marching toward the body with horizontal lines, then followed by an x-sweep, with vertical lines marching from left boundary toward right boundary of the computational grid.

The initial values of the time dependent problem are provided by either using unsteady mode alone or using both steady and unsteady modes. The program contains a switch for the choice. For fine mesh, such as 128 × 32, the method employing both modes is recommended. A run using both modes can be described as follows. First, using steady mode, calculations are first performed on a coarse mesh and then on a fine mesh with twice as many intervals in each coordinate direction. The coarse mesh result is interpolated to provide the starting guess for the fine mesh. It usually consists of 200 cycles on coarse mesh, 32×8, followed by 100 cycles on a fine mesh, 64×16, 50 cyles on a finer mesh 128×32. The resulting reduced velocity potential is used as the starting guess for the steady iteration using the unsteady routine. After 75 cycles in this steady iteration step, we begin our time marching calculation. A better initial value can be obtained after one complete periodic cycle. Computational results show that the difference between the second and the third period cycles is small. We therefore consider the results from the second period as our desired output.

The input data deck for a run is arranged to include title cards listing the required data items. The complete set of title cards provides a list of all the data which must be supplied and can be used as a guide in setting up the data deck. Each title card is followed by a card supplying the numerical values for the parameters listed. The input parameters are given in the Glossary in the order of their appearance on the data cards. All data items are read in as floating point numbers in fields of 10 columns, and values representing integer parameters are converted inside the program. The data deck for NACA 0012 at $M = 0.79$, $\alpha = 0° + 1° \sin kt$ is shown in Table 1. The output consists of printout and Calcomp plots. The program prints the mapped coordinates of the airfoil generated at the mesh points of the computational grid. Parameters such as mesh size, flight speed, flight angle, angle of attack are also printed so that the case can be identified easily.

For each iteration using the steady routine the program prints the iteration number, the maximum correction to the reduced velocity potential, and the maximum residual for the steady flow equation together with the coordinates of the point where these occur in the computational grid, the circulation, the relaxation factor p1, p2, p3, and the number of supersonic points. After a maximum number of cycles has been completed or a convergence criterion has been satisfied, the angle of attack, flight speed, flight angle, lift, drag and moment coefficients are printed. If desired, the pressure distribution along the airfoil surface and a chart of the local Mach numbers can be printed. If the mesh is to be refined, the program then repeats the same sequences of calculations and output on the same mesh. A Calcomp plot is generated to show the pressure distribution over the airfoil on the finest mesh at the end of this subroutine.

For a steady iteration using nonsteady mode, the program first prints the flight conditions, the mesh size, and the dimensionless time step. After each iteration, the program prints the maximum change in the velocity potential with the coordinates of the point in the grid system. If desired, the pressure distribution along the airfoil and the local Mach number chart can be printed. Calcomp plots for the pressure distribution, the leading distribution, and the supersonic zone over the airfoil are generated separately at the end of this step.

Before the unsteady time marching process, the advanced time steps required to finish the assigned period is estimated and printed. After one complete period has been computed the flight conditions together with the aerodynamic forces, lift, drag and moment coefficients, are thereafter printed periodically. If desired, the pressure distribution over the airfoil is also printed. Calcomp plots for the pressure distribution, the loading distribution, and the supersonic zone over the airfoil are generated periodically. At the end of the calculation the unsteady traces of the airfoil motion and the grid system near the airfoil are plotted. The graphs can also be produced as individual frames in a film strip. Then a complete history of the time dependent motion will be visible.

2. Glossary of the Program

The input parameters are listed in the order of their occurrence on the data title cards.

Title
Card 1

ISYM	Indicates the type of profile. ISYM = 0 denotes a cambered profile. Coordinates are supplied for upper and lower surfaces, each ordered from nose to tail with the leading edge included in both surfaces. ISYM = 1 denotes a symmetric profile. A table of coordinates is read for the upper surface only.
NU	The number of upper surface coordinates.
NL	The number of lower surface coordinates. For ISYM = 1, NL = NU even though no lower surface coordinates are given.
NX	The number of mesh cells in the direction of the chord used at the start of the calculation. NX = 0 causes termination of the program.
Ny	The number of mesh cells in the direction normal to the chord.
MHALF	Determines whether the mesh will be refined. MHALF = 0. The computation terminates after completing the prescribed number of iteration cycles or after convergence for the input mesh size.

MHALF $\neq$ 0. The mesh spacing will be halved after NRELAX cycles have been run on the crude mesh size. The refinement will be performed MHALF times.

RSTAD	Determines whether the steady mode will be employed. RSTAD = 0. The steady mode will not be called, the steady flow calculation entirely depends upon the unsteady mode. RSTAD = 1. Both steady and unsteady modes are employed for the steady flow calculation.
STADI	Indicates the type of flow calculation. STADI = 1. The computation is running for the steady state solution. STADI = 0. The computation is a time dependent run.

Title
Card 2

NRELAX	The maximum number of iteration cycles which will be computed in the steady iteration process.
RELAXTO	The desired accuracy. If the maximum correction is less than RELAX TO the calculation terminates or proceeds to a finer mesh; otherwise the number of cycles set by NRELAX are completed.
CHECKPT	Determines whether the CHEKPTX is required. CHEKPTX = 1. The CHEKPTX is called.

Title
Card 3

COORS	The stretching factor in the x coordinate stretching transformation described in Section III, 3.

COORT	The stretching factor in the y coordinate stretching mapping described in Section III, 3.
RCBDY	To locate the computational boundaries.
Title Card 4	
P10	The subsonic relaxation factor for the reduced potential in the steady flow calculation routines. It is between 1. and 2. and should be increased towards 2. as the mesh is refined.
P20	The supersonic relaxation factor for the reduced potential in the steady routine. It is not greater than 1. and normally set to 1.
P30	The relaxation factor for the circulation. It is usually set to 1. but can be increased
P101 P102 P103	The increments of P10 as the mesh system is refined 1 time, 2 times and 3 times, respectively.
Title Card 5	
FREQRA MAPLA	The frequency rate (rad/time) and amplitude (degree) of the sinusoidal variation of angle of attack (in degrees).
FREQRM AMPLM	The frequency rate and amplitude (mach number) of the sinusoidal variation of flight speed in mach number.
FREQRC AMPLC	The frequency rate and amplitude of the sinusoidal variation of flight angle in degrees.

PERIOD The complete sinusoidal periods to be calculated.

DEGREE The degree interval to plot the graphs, pressure distribution, the loading and the supersonic zone over the airfoil.

Title
Card 6

ALPHA1 The angle of attack in degrees.

MACH1 The flight speed in mach number. The speed of sound at infinity is set to be unity.

THETA1 The flight angle in degrees

TSRATIO The ratio $\Delta t/\Delta x_{(min)}$ between time step and minimum spacial step.

Title
Card 7

TE ANGLE The included angle at the trailing edge in degrees. The profile may be open, in which case it is the difference in angle between the upper and lower surfaces.

TE SLOPE The slope of the mean camber line at the trailing edge. This is used to continue the coordinate surface, assumed to contain the vortex sheet, smoothly off the trailing edge.

XSING
YSING The coordinates of the singular point inside the nose about which the square root transformation is applied to generate parabolic coordinates. This point should be located as symmetrically as possible between the upper and lower surfaces at a distance from the nose

roughly proportional to the leading edge radius. It can be seen whether the location has been correctly chosen by inspecting the coordinates of the mapped profile printed in the output. If the mapped profile has a bump at the center, the singular point should be moved closer to the leading edge. If the mapped profile is not symmetric near the center, with a step increase in y, say, as x increases through 0, the singular point should be moved closer to the upper surface.

Title
Card 8

X
Y
THETA

The coordinates, upper surface coordinates, of the upper surface and its tangent angle in degrees. These are read on the data cards which follow, one pair of coordinates and its tangent angle per card in the first three fields of 10 from leading to trailing edge inclusive.

Title
Card 9

X
Y
THETA

The coordinates and its tangent angle at the lower surface, read from leading to trailing edge. The leading edge point is the same as the upper surface leading edge point. The trailing edge point may be different if the profile has an open tail.

Title
Card 10

End of the calculation.

Cols. / Title Card	1-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80
	NACA0012							
1	ISYM	NU	NL	NX	NY	MHALF	RSTAD	STADI
	1.	37.	37.	32.	8.	2.	1.	0.
2	NRELAX	RELAX TO	CHECKPT					
	200.	1.E-6	0.					
3	COORS	COORT	RCBDY					
	0.	1.	1.					
4	P10	P20	P30	P101	P102	P103		
	0.94	0.8	1.	.19	.58	.72		
5	FREQRA	AMPLA	FREQM	AMPLM	FREQC	AMPLC	PERIOD	DEGREE
	0.3	1.	0.	0.	0.	0.	2.	90.
6	ALPHA1	MACH1	THETA1	TGRATIO				
	0.	.79	0.	10.				
7	TEANGLE	TESLOPE	XSING	YSING				
	16.15	0.	.8	0.				
8	X	Y	THETA	(UPPER SURFACE)				
1 NN								
9	X	Y	THETA	(LOWER SURFACE)				
1 NL								

Table 1. Data Deck for the Program.

3. LISTING OF THE PROGRAM

```
      PROGRAM UFLO5(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE98=OUTPUT,
     1           TAPE7)
C      THE ANALYSIS OF TRANSONIC FLOW PAST AIRFOIL IN RIGID BODY MOTION
C      THE UNSTEADY TRANSONIC POTENTIAL FLOW EQUATION WITH RADIATION
C      BOUNDARY CONDITIONS IN MOVING SHEARED PARABOLIC COORDINATE SYSTEM
C      ARE SOLVED BY AN ALTERNATING DIRECTION IMPLICIT SCHEME WITH
C      FIVE DIAGONAL MATRIX SOLVER
C      PROGRAMMED BY I-CHUNG CHANG DURING SEPTEMBER 1980
C      G IS THE VELOCITY POTENTIAL IN THE ABSOLUTE FRAME
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),S0(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/B/ SV(132),SM(132),CP(132)
      COMMON/C/ XP(260),YP(260),D1(260),D2(260),D3(260)
      COMMON/D/ SLOPT,TRAIL,SCAL
      COMMON/E/ CHORD,XM,CL,CD,CM
      COMMON/F/ XR,YR,KS,XS(500),YS(500)
      COMMON/G/ TITLE(20),IPLOT
      COMMON/H/ DX,DY,DT,DXX,DYY,DTT,DXY,DXT,DYT,TSR
      COMMON/I/ X(260),Y(260)
      COMMON/J/ RAD,PI,ALS,ALT,ALTT,AMPLA,FREQRA,FASAGA,FMACHS,FMACHT
     1 ,AMPLM,FREQRM,FASAGM,CETAS,CETAT,CETATT,AMPLC,FREQRC,FASAGC,CETA
     2 ,FREQR,IPSURE
      COMMON/K/ I0,I1,I2,I3,J1,J2,J3
      COMMON/L/ TCL(801),TCD(801),TCM(801),TCP(9,801),TCPS(9),CLS,CDS
     1          ,CMS,NITS,IJUMP,NSTEP,JSTEP,PERIOD,MHALF
      COMMON/M/ P1,P2,P3,TAU
      COMMON/O/ COORS,COORT
      COMMON/STADI/ RR,IR,JR,IRSTAD
      COMMON/WAKE/ NIT,WG(132)
      DATA VAR/0/
      IREAD= 5
      IWRIT      = 6
      IPLOT      = -1
      PI         = 3.14159265358979
      RAD        = 57.2957795130823
    1 WRITE (IWRIT,600)
      WRITE (IWRIT,2)
    2 FORMAT(14HOPROGRAM UFLO5,70X,32H I-CHUNG CHANG,COURANT INSTITUTE/
     1       56HOSOLUTION OF UNSTEADY TRANSONIC POTENTIAL FLOW EQUATIONS)
      READ  (IREAD,530) (TITLE(I),I=1,20)
      WRITE (IWRIT,630) (TITLE(I),I=1,20)
      READ   (IREAD,500)
      READ   (IREAD,510) FSYM,FNU,FNL,FNX,FNY,FHALF,FRSTAD,FSTADI
      ISYM        = FSYM
      IRSTAD= FRSTAD
      ISTADI= FSTADI
      NU          = FNU
      NL          = FNL
      IF (NU.LT.1) GO TO 302
      NXO         = FNX
      NYO         = FNY
      NX          = NXO
      NY          = NYO
      IF(NX.NE.4*NY) GO TO 302
      MHALF       = FHALF
      NHALF       = 0
      READ   (IREAD,500)
      READ   (IREAD,510) FIT,COV,CHECPT
      ICHECK= CHECPT
      READ(IREAD,500)
      READ(IREAD,510)  COORS,COORT,RCBDY
      LHALF= RCBDY
      IF(LHALF.LE.0.OR.LHALF.GT.3) LHALF=1
      READ(IREAD,500)
      READ(IREAD,510) P10,P20,P30,P101,P102,P103
      READ(IREAD,500)
      READ(IREAD,510)FREQRA,AMPLA,FREQRM,AMPLM,FREQRC,AMPLC,PERIOD,DEGRE
      IF(PERIOD.GT.2.) DEGRE= 45.
      IF(PERIOD.LT.3) GO TO 3
      DEGRE= 45.
      IF(PERIOD.LT.4) GO TO 3
      DEGRE= 90.
    3 IGRAF= 360./DEGRE
      IF(IGRAF.GT.12) IGRAF= 12
      IPSURE= PERIOD*IGRAF
      FREQR= 100.
      IF(FREQRA.LE.0.) GO TO 4
      FREQR= AMIN1(FREQRA,FREQR)
    4 IF(FREQRM.LE.0.) GO TO 5
      FREQR= AMIN1(FREQRM,FREQR)
    5 IF(FREQRC.LE.0.) GO TO 6
      FREQR= AMIN1(FREQRC,FREQR)
    6 IF(FREQR.LE.0.) GO TO 302
      MITO        = FIT
      READ   (IREAD,500)
      READ    (IREAD,510)  AL1,FM1,CT1,TSR
      FMACH       = FM1
      FMACH2=FMACH*FMACH
      CETA= CT1
      CETAS= CETA/RAD
      READ   (IREAD,500)
      READ   (IREAD,510) TRAIL,SLOPT,XR,YR
      TRAIL       = TRAIL/RAD
      N           = NL   +NU   -1
      READ   (IREAD,500)
      DO 7 I=NL,N
    7 READ  (IREAD,510) X(I),Y(I)
      L           = NL   +1
      IF (ISYM.GT.0) GO TO 9
      READ  (IREAD,500)
      DO 8 I= 1,NL
      READ  (IREAD,510) VAL,DUM
      J           = L   -I
      X(J)        = VAL
    8 Y(J)        = DUM
      GO TO 11
    9 J           = L
      DO 10 I=NL,N
      J           = J  -1
      X(J)        = X(I)
   10 Y(J)        = -Y(I)
   11 CHORD       = X(1)   -X(NL)
      XM          = X(NL)   +.25*CHORD
      AL          = AL1
   12 ALPHA       = AL/RAD
      ALS= ALPHA
      KSYM        = ISYM
      IF (ALPHA.NE.0.) KSYM = 0
   13 CALL COORD(NL,N)
      IF(IX1+IX2.NE.NX+4) GO TO 302
      IF(IRSTAD.GT.0) GO TO 37
      CALL ESTIM
      GO TO 38
   37 CALL SESTIM
   38 UTIM= 0.
      MIT         = MITO
      ALT=0.
      ALTT= 0.
      FMACHS= FMACH
      FMACHT= 0.
      CETAT= 0.
      CETATT= 0.
   14 WRITE (IWRIT,600)
      WRITE (IWRIT,112)
      KX= NX+1
      DO 15 I= 3,KX
   15 WRITE (IWRIT,610) AO(I),SO(I),S1(I),S2(I),A1(I),A2(I),A3(I)
      WRITE (IWRIT,600)
      WRITE (IWRIT,116)
      MY= NY+2
      DO 16 J= 3,MY
   16 WRITE (IWRIT,610) BO(J),B1(J),B2(J),B3(J)
      IF(IRSTAD.GT.0) GO TO 50
      IF(NHALF.EQ.MHALF) MIT= MIT*1.5
      NIT=0
      J1= 3
      KHALF=LHALF*2**NHALF
      J3= 3 + NY-KHALF
      J2= J3 -1
      I0= 2 + KHALF
      I1 = I0 +1
      I3= 2 + NX -KHALF
      I2= I3 -1
      INX= I3 - I0
      INY= J3 - J1
      WRITE (IWRIT,600)
      WRITE (IWRIT,124)
      WRITE(IWRIT,640) INX,INY
      WRITE(IWRIT,126)
      WRITE(IWRIT,610) FMACH,AL
      WRITE(IWRIT,132) DT
      CALL SECOND(T)
      WRITE(IWRIT,660) T
      WRITE(IWRIT,128)
   20 NIT        = NIT  +1
      CALL USTADI
      WRITE(IWRIT,650) NIT,RG,IG,JG,NS
      IF(NIT.GE.MIT) GO TO 21
      IF(RG.GT.COV) GO TO 20
   21 CALL SECOND(T)
      WRITE (IWRIT,660) T
      IF( NHALF.GE.MHALF) GO TO 22
      NHALF=   NHALF + 1
      MIT=MIT/2
      NX=NX  +NX
      NY=   NY+  NY
      CALL COORD(NL,N)
      IF(IX1+IX2.NE.NX+4) GO TO 302
      CALL REFIN
      GO TO 14
C     USING THE STEADY MODE TO GENERATE THE INITIAL DATA
   50 WRITE (IWRIT,600)
      WRITE (IWRIT,124)
      WRITE (IWRIT,640) NX,NY
      WRITE (IWRIT,126)
      WRITE (IWRIT,610) FMACH,AL
      CALL SECOND(T)
      WRITE (IWRIT,660) T
      WRITE(IWRIT,129)
      NIT=0
      ALT=0.
      P1          = 2./(1.  +(2./P10  -1.)*.5**NHALF)
      IF(P101.EQ.0.) GO TO 51
      P1= P10
      IF( NHALF.EQ.1) P1= P10 + P101
      IF( NHALF.EQ.2.) P1= P10 + P102
      IF(NHALF.EQ.3) P1= P10 + P103
   51 P2= P20
      P3          = 1.
      J1= 3
   52 J3= NY+2
      I0= 3
      I3= NX+1
   53 J2= J3-1
      I1= I0 + 1
      I2= I3-1
C     STEADY ITERATION USING THE STEADY MODE
   54 NIT        = NIT  +1
      CALL STEADY
      WRITE(IWRIT,670) NIT,RG,IG,JG,RR,IR,JR,TAU,P1,P2,P3,NS
      IF(NIT.GE.MIT) GO TO 55
      IF(RR.GT.COV) GO TO 54
   55 CALL SECOND(T)
      WRITE (IWRIT,660) T
      IF( NHALF.LT.MHALF) GO TO 57
      CALL SVELO
      CALL FORCE
      WRITE (IWRIT,600)
      WRITE (IWRIT,182)
      WRITE(IWRIT,610) AL,FMACH,CETA,CL,CD,CM
      WRITE (IWRIT,184)
      DO 56 I=IX1,IX2
   56 WRITE (IWRIT,610) XP(I),YP(I),SV(I),SM(I),CP(I)
      WRITE (IWRIT,600)
      CALL CPLOT
      WRITE (IWRIT,600)
      CALL SCHART
      IROUTE= 1
      CALL PSURE
      IPLOT= 0
      IF(ISTADI.GT.0) GO TO 303
      GO TO 58
   57 NHALF=   NHALF + 1
      NX=NX  +NX
      NY=   NY+  NY
      CALL COORD(NL,N)
      IF(IX1+IX2.NE.NX+4) GO TO 302
      CALL SREFIN
      MIT=MIT/2
      GO TO 14
C     STEADY ITERATION USING UNSTEADY MODE
   58 NIT=0
      ALT=0.
      ALTT= 0.
      FMACHS= FMACH
      FMACHT= 0.
      CETAT= 0.
      CETATT= 0.
      CALL ESTIM
      MIT= MITO / 2
      WRITE(IWRIT,600)
      WRITE(IWRIT,134)
      WRITE(IWRIT,124)
      KHALF=LHALF*2**NHALF
      J3= 3 + NY-KHALF
      J2= J3 -1
      I0= 2 + KHALF
      I1 = I0 +1
      I3= 2 + NX -KHALF
      I2= I3 -1
      INX= I3 - I0
      INY= J3 - J1
      WRITE(IWRIT,640) INX,INY
      WRITE(IWRIT,126)
      WRITE(IWRIT,610) FMACH,AL
      WRITE(IWRIT,132) DT
      CALL SECOND(T)
      WRITE(IWRIT,660) T
      WRITE (IWRIT,128)
   59 NIT       = NIT  +1
      CALL USTADI
      WRITE(IWRIT,650) NIT,RG,IG,JG,NS
      IF(NIT.LT.3) GO TO 59
      IF( NIT.GE.MIT) GO TO 60
      IF(RR.GT.COV) GO TO 59
   60 CALL SECOND(T)
      WRITE (IWRIT,660) T
C     UNSTEADY CALCULATION --- TIME MARCHING
   22 NIT=0
C     INITIAL DATA
      NITS= 1
      JSTEP= 0
      CALL VELO
      CALL FORCE
      WRITE (IWRIT,600)
      WRITE (IWRIT,182)
      WRITE(IWRIT,610) AL,FMACH,CETA,CL,CD,CM
      WRITE (IWRIT,184)
      DO 23 I=IX1,IX2
   23 WRITE (IWRIT,610) XP(I),YP(I),SV(I),SM(I),CP(I)
      WRITE (IWRIT,600)
      CALL CPLOT
      WRITE (IWRIT,600)
      CALL CHART
      IROUTE= 1
      CALL PSURF
      IPLOT= 0
      IF(ISYM.GT.0.AND.ALS.EQ.0..AND.AMPLA.EQ.0..AND.AMPLC.EQ.0.) GOTO35
      IROUTE= IROUTE + 1
      CALL LURD
   35 IROUTE= IROUTE+ 1
      CALL SONIC
      CALL SECOND(T)
      WRITE (IWRIT,660) T
      IF(ISTADI.GT.0) GO TO 303
      WRITE(6,600)
      ISTEP= PERIOD*2.*PI/(FREQR*DT)
      WRITE(IWRIT,136) ISTEP
      MSTEP= ISTEP/IPSURE
      IJKLMN= MIN0(800,ISTEP)
      KFORC= IJKLMN/IPSURE
```

```
    JCHECK= ISTEP*.25
 24 IF(MOD(MSTEP,KFORC).EQ.0) GO TO 25
    KFORC= KFORC-1
    GO TO 24
 25 NSTEP= ISTEP/(KFORC*IPSURE)
    LSTEP= ISTEP + 5
    KSTEP= 1.
    KKSTEP= FLOAT(ISTEP)*.5
 26 CALL USTADI
    IF(PERIOD.GE.2.) GO TO 33
 34 IF(MOD(KSTEP,MSTEP).EQ.0) GO TO 27
    GO TO 30
 33 IF(KSTEP.LT.KKSTEP) GO TO 30
    GO TO 34
 27 IF(IROUTE.LT.25) GO TO 28
    CALL ROUTE(6LOUTPUT,2LLP)
    IROUTE= 0
    CALL PLOT(0.,0.,999)
    IPLOT= -1
 28 IROUTE=  IROUTE +1
    CALL VELO
    JSTEP= JSTEP + 1
    CALL FORCE
    WRITE (IWRIT,600)
    WRITE(IWRIT,138) UTIM,KSTEP
    WRITE (IWRIT,182)
    WRITE(IWRIT,610) AL,FMACH,CETA,CL,CD,CM
    WRITE (IWRIT,184)
    DO 29 I= IX1,IX2
 29 WRITE (IWRIT,610) XP(I),YP(I),SV(I),SM(I),CP(I)
    CALL PSURE
    IPLOT= 0
    IF(ISYM.GT.0.AND.ALS.EQ.0..AND.AMPLA.EQ.0..AND.AMPLC.EQ.0.) GOTO36
    IROUTE= IROUTE + 1
    CALL LORD
 36 IROUTE= IROUTE+ 1
    CALL SONIC
    GO TO 32
 30 IF(MOD(KSTEP,NSTEP).EQ.0) GO TO 31
    GO TO 32
 31 CALL VELO
    JSTEP=JSTEP+1
    CALL FORCE
 32 KSTEP=KSTEP +1
    IF(MOD(KSTEP,JCHECK).EQ.0) GO TO 41
 39 IF(KSTEP.GE.LSTEP) GO TO 301
    GO TO 26
 41 IF(ICHECK.GT.0) CALL CHEKPTX(VAR)
    GO TO 39
301 CALL TRACE
    J1= 3
    KHALF= 3* 2**MHALF
    J3= 3 + NY-KHALF
    IO= 2 + KHALF
```

```
      I3= 2 + NX -KHALF
      CALL GRID
  303 CALL PLOT(0.,0.,999)
  302 STOP
  112 FORMAT(41HOMAPPED COORDINATES AND X STRETCH FACTORS/
     1        15HO        X       ,15H        Y       ,15H        YP      ,
     2        15H       YPP       ,15H        A1      ,15H        A2      ,
     3        15H        A3       )
  116 FORMAT(18HOY STRETCH FACTORS/
     1        15HO        Y       ,15H        B1      ,15H        B2      ,
     2        15H        B3       )
  124 FORMAT(15HO HOR DIVISIONS,15H  VER DIVISIONS)
  126 FORMAT(15HO     MACH NO   ,15H  ANG OF ATTACK)
  128 FORMAT(10HOITERATION,15H    CORRECTION ,5H   I ,5H   J ,
     1      5H      ,10HSONIC PTS )
  129 FORMAT(10HOITERATION,15H    CORRECTION ,5H   I ,5H   J ,
     1                          15H     RESIDUAL  ,5H   I ,5H   J ,
     2        10H CIRCULATN,10H REL FCT 1,10H REL FCT 2,10H REL FCT 3,
     3        10H SONIC PTS)
  132 FORMAT(1HO,*TIME STEP =  *,F15.10)
  134 FORMAT(1HO,*UNSTEADY ITERATION*)
  136 FORMAT(1HO,*UNSTEADY STEPS = *, 5X,I10)
  138 FORMAT(1H ,*TIME=*,5X,F10.5,5X,*STEP=*,5X,I10,/,/)
  182 FORMAT(15HO ANG OF ATTACK,15H  FLIGHT SPEED ,15H  FLIGHT ANGLE ,
     1        15H        CL       ,15H        CD      ,15H        CM      )
  184 FORMAT(36HOCOORDINATES OF INTERPOLATED AIRFOIL,
     1        26H AND PRESSURE DISTRIBUTION/
     2        15H        X        ,15H        Y       ,15H       V/VO     ,
     3        15H     MACH NO     ,15H        CP      )
  500 FORMAT(1X)
  510 FORMAT(8F10.7)
  530 FORMAT(20A4)
  600 FORMAT(1H1)
  610 FORMAT(8F15.4)
  620 FORMAT(8E15.5)
  630 FORMAT(1HO,20A4)
  640 FORMAT(I8,7I15)
  650 FORMAT(I10,E15.5,2I5,I10)
  660 FORMAT(15HOCOMPUTING TIME,F10.3,10H   SECONDS)
  670 FORMAT(I10,E15.5,2I5,E15.5,2I5,4F10.5,I10)
      END


      SUBROUTINE COORD(L,N)
C     SETS UP MODIFIED PARABOLIC COORDINATE SYSTEM
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),SO(132),S1(132),S2(132)
     1         ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2         ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3         ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/C/ XP(260),YP(260),D1(260),D2(260),D3(260)
      COMMON/D/ SLOPT,TRAIL,SCAL
      COMMON/F/ XR,YR,KS,XS(500),YS(500)
      COMMON/H/ DX,DY,DT,DXX,DYY,DTT,DXY,DXT,DYT,TSR
      COMMON/I/ X(260),Y(260)
      COMMON/O/ COORS,COORT
      PI        = 3.14159265358979
      DX=4./NX
      DY        = 1./NY
      DXY=DX*DY
      DYY=DY*DY
      DXX=DX*DX
      KX= NX+1
      MX= NX+2
      KY        = NY   +1
      MY        = NY   +2
      S= COORS
      T= COORT
      XTE=1.
      SCAL      =-.50001*XTE**2/(X(N)   -XR)
      DO 12 I=1,N
      XO        = SCAL*(X(I)   -XR)
      YO        = SCAL*(Y(I)   -YR)
      R         = SQRT(XO*XO   +YO*YO)
      ANGL      = CMPLA(XO,YO)
      IF (I.LT.L.AND.ANGL.LT..5*PI) ANGL = ANGL   +PI   +PI
      IF (I.GT.L.AND.ANGL.GT.1.5*PI) ANGL = ANGL   -PI   -PI
      R         = SQRT(R   +R)
      ANGL      = .5*ANGL
      XP(I)     = R*COS(ANGL)
   12 YP(I)     = R*SIN(ANGL)
      DO 22 I= 3,KX
      XX= (I-2)*DX -2.
      B         = 1.
      IF (ABS(XX).GT.XTE) GO TO 23
      SX        = SIN(PI*XX/XTE)
      CX        = COS(PI*XX/XTE)
      XO        = XX   +S*XTE*SX/(PI*(1.   +S))
      X1        = 1./(1.   +S*(1.   +CX))
      X2        = S*PI*SX*X1/XTE
      X1        = (1.   +S)*X1
      GO TO 24
   23 IF (XX.LT.O.) B = -1.
      A         = 1.   -((XX   -B*XTE)/(2.   -XTE))**2
      XO        = B*XTE   +(XX   -B*XTE)/(A*(1.   +S))
      X1        = (1.   +S)*A*A/(2.   -A)
      X2        = -2.*(XX   -B*XTE)*(4.   -A)/(A*(2.   -A)*(2.   -XTE)**2)
   24 IF (XO.LT.XP(1)) IX1 = I
      IF (XO.LE.XP(N)) IX2 = I
      AO(I)     = XO
      A1(I)     = .5*X1/DX
      A2(I)     = X1*X1
   22 A3(I)=.5*X2/DX
      IX1       = IX1   +1
      DO 32 J= 3,MY
      YY= (J-3) *DY
      B         = 1.   -YY*YY
      Y1          = B*B/((2.   -B)*T)
      BO(J)       = T*YY/B
      B1(J)       = .5*Y1/DY
      B2(J)=Y1*Y1
   32 B3(J)=-YY*(4.-B)/(B*(2.-B)*DY)
      ANG         = ATAN(SLOPT)
      ANG1        = CMPLA((X(1)   -XR),(Y(1)   -YR))
      IF (ANG1.GT.PI) ANG1 = ANG1  -PI  -PI
      ANG2        = CMPLA((X(N)   -XR),(Y(N)   -YR))
      IF (ANG2.GT.PI) ANG2 = ANG2  -PI  -PI
      ANG1        = ANG  -.5*ANG1  +.5*TRAIL
      ANG2        = ANG  -.5*ANG2  -.5*TRAIL
      T1          = TAN(ANG1)
      T2          = TAN(ANG2)
      CALL SPLIF (1,N,XP,YP,D1,D2,D3,1,T1,1,T2,0,0.,IND)
      CALL INTPL (IX1,IX2,AO,SO,1,N,XP,YP,D1,D2,D3,0)
      X1          = X(1)  -.75*(X(1)   -X(L))
      SO(2)= 0.
      SO(MX)= 0.
      M           = IX1  -1
      A           = SLOPT*(X(1)   -X1)
      C           = 1./(X(1)   -X1)
      DO 42 I= 3,M
      XX          = .5*AO(I)**2/SCAL   +XR
      XO          = SCAL*(XX   -XR)
      YO          = SCAL*(Y(1)   +A*ALOG(C*(XX   -X1))   -YR)
      R           = SQRT(XO*XO   +YO*YO)
      ANGL        = CMPLA(XO,YO)
      IF (ANGL.LT..5*PI) ANGL = ANGL   +PI   +PI
      R           = SQRT(R   +R)
      ANGL        = .5*ANGL
   42 SO(I)       = R*SIN(ANGL)
      M           = IX2  +1
      A           = SLOPT*(X(N)   -X1)
      C           = 1./(X(N)   -X1)
      DO 52 I= M,KX
      XX          = .5*AO(I)**2/SCAL   +XR
      XO          = SCAL*(XX   -XR)
      YO          = SCAL*(Y(N)   +A*ALOG(C*(XX   -X1))   -YR)
      R           = SQRT(XO*XO   +YO*YO)
      ANGL        = CMPLA(XO,YO)
      IF (ANGL.GT.1.5*PI) ANGL = ANGL   -PI   -PI
      R           = SQRT(R   +R)
      ANGL        = .5*ANGL
   52 SO(I)       = R*SIN(ANGL)
      SCAL        = 1./SCAL
      DO 62 I= 3,KX
      DSI         = SO(I+1)   -SO(I-1)
      DSII=(SO(I+1)-2.*SO(I)+SO(I-1))/DXX +A3(I)*DSI
      S1(I)       = A1(I)*DSI
   62 S2(I)=A2(I)*DSII
      DO 72 I=IX1,IX2
      XP(I)       = .5*SCAL*(AO(I)**2   -SO(I)**2)   +XR
   72 YP(I)       = SCAL*AO(I)*SO(I)   +YR
```

```
      RETURN
      END



      SUBROUTINE  SPLIF(M,N,S,F,FP,FPP,FPPP,KM,VM,KN,VN,MODE,FQM,IND)
C     SPLINE FIT - JAMESON
C     INTEGRAL PLACED IN FPPP IF MODE GREATER THAN 0
C     IND SET TO ZERO IF DATA ILLEGAL
      DIMENSION   S(1),F(1),FP(1),FPP(1),FPPP(1)
      IND         = 0
      K           = IABS(N  -M)
      IF (K  -1) 81,81,1
    1 K           = (N  -M)/K
      I           = M
      J           = M  +K
      DS          = S(J)  -S(I)
      D           = DS
      IF (DS) 11,81,11
   11 DF          = (F(J)  -F(I))/DS
      IF (KM  -2) 12,13,14
   12 U           = .5
      V           = 3.*(DF  -VM)/DS
      GO TO 25
   13 U           = 0.
      V           = VM
      GO TO 25
   14 U           = -1.
      V           = -DS*VM
      GO TO 25
   21 I           = J
      J           = J  +K
      DS          = S(J)  -S(I)
      IF (D*DS) 81,81,23
   23 DF          = (F(J)  -F(I))/DS
      B           = 1./(DS  +DS  +U)
      U           = B*DS
      V           = B*(6.*DF  -V)
   25 FP(I)       = U
      FPP(I)      = V
      U           = (2.  -U)*DS
      V           = 6.*DF  +DS*V
      IF (J  -N) 21,31,21
   31 IF (KN  -2) 32,33,34
   32 V           = (6.*VN  -V)/U
      GO TO 35
   33 V           = VN
      GO TO 35
   34 V           = (DS*VN  +FPP(I))/(1.  +FP(I))
   35 B           = V
      D           = DS
   41 DS          = S(J)  -S(I)
      U           = FPP(I)  -FP(I)*V
      FPPP(I)    = (V   -U)/DS
      FPP(I)     = U
      FP(I)      = (F(J)   -F(I))/DS   -DS*(V   +U   +U)/6.
      V          = U
      J          = I
      I          = I   -K
      IF (J   -M) 41,51,41
   51 I          = N   -K
      FPPP(N)    = FPPP(I)
      FPP(N)     = B
      FP(N)      = DF   +D*(FPP(I)   +B   +B)/6.
      IND        = 1
      IF (MODE) 81,81,61
   61 FPPP(J)    = FQM
      V          = FPP(J)
   71 I          = J
      J          = J   +K
      DS         = S(J)   -S(I)
      U          = FPP(J)
      FPPP(J)    = FPPP(I)   +.5*DS*(F(I)   +F(J)   -DS*DS*(U   +V)/12.)
      V          = U
      IF (J   -N)  71,81,71
   81 RETURN
      END



      FUNCTION     CMPLA(X,Y)
C     ANGLE OF COMPLEX NUMBER X +I*Y IN RANGE 0. TO 2.*PI
      PI         = 3.14159265358979
      IF (ABS(Y)   -ABS(X)) 1,1,11
    1 SHIFT      = PI
      IF (X) 4,21,2
    2 SHIFT      = 0.
      IF (Y) 3,4,4
    3 SHIFT      = 2.*PI
    4 CMPLA      = SHIFT   +ATAN(Y/X)
      GO TO 31
   11 SHIFT      = .5*PI
      IF (Y) 12,12,13
   12 SHIFT      = 1.5*PI
   13 CMPLA      = SHIFT   -ATAN(X/Y)
      GO TO 31
   21 CMPLA      = 0.
   31 RETURN
      END



      SUBROUTINE   INTPL(MI,NI,SI,FI,M,N,S,F,FP,FPP,FPPP,MODE)
C     INTERPOLATION USING TAYLOR SERIES - JAMESON
C     ADDS CORRECTION FOR PIECEWISE CONSTANT FOURTH DERIVIATIVE
C     IF MODE GREATER THAN 0
      DIMENSION    SI(1),FI(1),S(1),F(1),FP(1),FPP(1),FPPP(1)
      K           = IABS(N   -M)
      K           = (N   -M)/K
      I           = M
      MIN         = MI
      NIN         = NI
      D           = S(N)   -S(M)
      IF (D*(SI(NI)   -SI(MI))) 11,13,13
   11 MIN         = NI
      NIN         = MI
   13 KI          = IABS(NIN   -MIN)
      IF (KI) 21,21,15
   15 KI          = (NIN   -MIN)/KI
   21 II          = MIN   -KI
      C           = 0.
      IF (MODE) 31,31,23
   23 C           = 1.
   31 II          = II   +KI
      SS          = SI(II)
   33 I           = I   +K
      IF (I   -N) 35,37,35
   35 IF (D*(S(I)   -SS)) 33,33,37
   37 J           = I
      I           = I   -K
      SS          = SS   -S(I)
      FPPPP       = C*(FPPP(J)   -FPPP(I))/(S(J)   -S(I))
      FI(II)      = QUARP(SS,F(I),FP(I),FPP(I),FPPP(I),FPPPP)
      IF (II   -NIN) 31,41,31
   41 RETURN
      END



      FUNCTION    QUARP(DS,F,FP,FPP,FPPP,FPPPP)
C     EVALUATES FIRST FOUR TERMS OF TAYLOR SERIES - JAMESON
      QUARP       = FPPP   +.25*DS*FPPPP
      QUARP       = FPP   +DS*QUARP/3.
      QUARP       = FP   +.5*DS*QUARP
      QUARP       = F   +DS*QUARP
      RETURN
      END



      SUBROUTINE  PARAF(S1,S2,S3,F1,F2,F3,F,FP,FPP)
C     PARABOLIC FIT - JAMESON
      S0          = .5*(S3   +S1)
      FP0         = (F3   -F1)/(S3   -S1)
      FPP         = (F3   -F2)/(S3   -S2)   -(F2   -F1)/(S2   -S1)
      FPP         = 2.*FPP/(S3   -S1)
      FP          = FP0   -FPP*S0
      F           = F2   -S2*(FP0   +FPP*(.5*S2   -S0))
      RETURN
      END
      SUBROUTINE SESTIM
C     STEADY ROUTINE
C     INITIAL ESTIMATE OF REDUCED POTENTIAL
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),S0(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/M/ P1,P2,P3,TAU
      COMMON/WAKE/ NIT,WG(132)
      IX= NX+4
      IY= NY+4
      KX= NX+1
      MY         = NY  +2
      CB= COS(ALPHA)
      SB= SIN(ALPHA)
      CA= FMACH*CB
      SA= FMACH*SB
      TAU= 0.
      DO 12 I= 1,IX
      DO 12 J= 1,IY
   12 G(I,J)     = 0.
      DO 22 I=IX1,IX2
      U0=                    CA*A0(I)  +SA*S0(I)
      BIS=   B1(3)*(1.    +S1(I)**2)
   22 G(I,2)=   G(I,4)-(CA*S0(I)   -SA*A0(I)   +U0*S1(I))/BIS
      DO 23 I= IX2,KX
      M= NX+4 -I
   23 WG(I)= G(I,3)-G(M,3)
      RETURN
      END




      SUBROUTINE SREFIN
C     STEADY ROUTINE
C     HALVES MESH SIZE
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),S0(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/WAKE/ NIT,WG(132)
      KX= NX+1
      MX= NX+2
      KY          = NY  +1
      MY=           NY+2
      IY= NY+3
      LX= NX/2 + 2
      LY= NY/2 +3
      DO 22 K= 2,LX
      I= LX+2-K
      II= (I-2)*2 +2
      DO 22 KK= 3,LY
      J= LY+3-KK
      JJ= (J-3)*2 +3
   22 G(II,JJ)=        G(I,J)
      DO 42 I= 2,MX,2
      DO 42 J= 4,MY,2
   42 G(I,J)     = .5*(G(I,J+1)  +G(I,J-1))
      DO 32 J= 3,IY
      DO 32 I= 3,KX,2
   32 G(I,J)     = .5*(G(I+1,J)  +G(I-1,J))
      DO 33 I= IX2,KX
      M= NX+4-I
   33 WG(I)= G(I,3)-G(M,3)
      DO 52 I=IX1,IX2
      GI= G(I+1,3)-G(I-1,3)
      U0=        A1(I)*GI +CA*A0(I)  +SA*S0(I)
      BIS=    B1(3)*(1.    +S1(I)**2)
   52 G(I,2)= G(I,4)     -(CA*S0(I) -SA*A0(I) +U0*S1(I))/BIS
      N            = IX1  -1
      DO 62 I= 3,N
      M= NX+4 -I
   62 G(M,2)= G(I,4) + WG(M)
      N            = IX2  +1
      DO 64 I= N,KX
      M= NX+4 -I
   64 G(M,2)= G(I,4)  -WG(I)
      RETURN
      END



      SUBROUTINE SVELO
C     STEADY ROUTINE
C     CALCULATES SURFACE VELOCITY
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),S0(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UIIM,CB,SB,NS,RG,IG,JG
      COMMON/B/ SV(132),SM(132),CP(132)
      COMMON/J/ RAD,PI,ALS,ALT,ALTT,AMPLA,FREQRA,FASAGA,FMACHS,FMACHT
     1 ,AMPLM,FREQRM,FASAGM,CETAS,CETAT,CETATT,AMPLC,FREQRC,FASAGC,CETA
     2 ,FREQR,IPSURF
      COMMON/K/ I0,I1,I2,I3,J1,J2,J3
      AA0= 1. + .2*FMACH2
      DO 12 I=IX1,IX2
      Y=B0(3)+S0(I)
      H=SQRT(A0(I)**2 +S0(I)**2)
      GI= G(I+1,3)-G(I-1,3)
      GJ= G(I,4)-G(I,2)
      U          =(A1(I)* GI  -S1(I)*B1(3)* GJ  +CA*A0(I)  +SA*Y)/H
      V          = (B1(3)* GJ  +SA*A0(I)  -CA*Y)/H
      QQ=U*U+V*V
      Q=SQRT(QQ)
      IF (U.LT.0.) Q = -Q
      SV(I)      = Q
      AA= AAO -.2*QQ
      AA= ABS(AA)
      A= SQRT(AA)
      SM(I)= Q/A
   12 CP(I)=   (AA**3.5  -1.)/(.7*FMACH2)
      RETURN
      END



      SUBROUTINE SCHART
C     STEADY ROUTINE
C     GENERATES MACH NO CHART
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),S0(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/J/ RAD,PI,ALS,ALT,ALTT,AMPLA,FREQRA,FASAGA,FMACHS,FMACHT
     1 ,AMPLM,FREQRM,FASAGM,CETAS,CETAT,CETATT,AMPLC,FREQRC,FASAGC,CETA
     2 ,FREQR,IPSURE
      COMMON/K/ I0,I1,I2,I3,J1,J2,J3
      DIMENSION IND(150)
      AAO = 1. + .2 * FMACH2
      IWRIT       = 6
      K           = NY/32
      IF (NY.GT.32*K) K = K  +1
      WRITE (IWRIT,2)
    2 FORMAT(14HOMACH NO CHART)
   11 DO 12 I= I0,I3
      J=J1
      N= 0
   14 N= N+1
      Y           = S0(I)  +B0(J)
      HH=A0(I)*A0(I)+Y*Y
      H=SQRT(HH)
      GI=G(I+1,J)-G(I-1,J)
      GJ=G(I,J+1)-G(I,J-1)
      U           =(A1(I)* GI  -S1(I)*B1(J)* GJ  +CA*A0(I)  +SA*Y)/H
      V           = (B1(J)* GJ  +SA*A0(I)  -CA*Y)/H
      QQ=U*U+V*V
      AA= AAO -.2*QQ
      AA= ABS(AA)
      QA=  QQ/AA
      IND(N)= 100.*SQRT(QA)
      IF (U.LT.0.) IND(N) = -IND(N)
      J= J+K
      IF(J.LE.J2) GO TO 14
   12 WRITE (IWRIT,610) (IND(J),J=1,N)
      RETURN
  610 FORMAT(1X,32I4)
      END
      SUBROUTINE STEADY
C     STEADY ROUTINE
C     STEADY TRANSONIC POTENTIAL FLOW EQUATION IN SHEARED
C     PARABOLIC COORDINATES SYSTEM SOLVED BY ROW RELAXATION
C     G IS THE VELOCITY POTENTIAL IN THE ABSOLUTE FRAME
C     AND IS THE REDUCED POTENTIAL IN THE UNIFORM MOVING FRAME
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),S0(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/H/ DX,DY,DT,DXX,DYY,DTT,DXY,DXT,DYT,PDT
      COMMON/K/ I0,I1,I2,I3,J1,J2,J3
      COMMON/M/ P1,P2,P3,TAU
      COMMON/STADI/ RR,IR,JR,IRSTAD
      DIMENSION C(132),D(132)
      J4= J3 +1
      I4= I3 +1
      II= I0-1
      II0= I0 +2
      II3= I3 -2
      DD= 1./DXX
      EE= 1./DYY
      NS         = 0
      AA0 = 1. + .2 * FMACH2
      RR=0.
      IR         = 0
      JR         = 0
      RG=0.
      IG         = 0
      JG         = 0
      RE=0.
      IE=0.
      JE=0.
      Q1         = 2./P1
      Q2         = 1./P2
      C(II)= 0.
      D(II)= 0.
      J= J3
   21 DO 32 I= I0,I3
      FX=1.+S1(I)**2
      Y=S0(I)+B0(J)
      HH=A0(I)*A0(I)+Y*Y
      H=SQRT(HH)
      DH= 1./H
      GI=G(I+1,J) -G(I-1,J)
      GJ=G(I,J+1)-G(I,J-1)
      U          =(A1(I)* GI   -S1(I)*B1(J)* GJ   +CA*A0(I)   +SA*Y)*DH
   36 V          = (B1(J)* GJ   +SA*A0(I)   -CA*Y)*DH
      AV         = V   -U*S1(I)
      AU= U + V*S1(I)
      S          = 1.
      IF (U.LT.0.) S = -1.
      T    =    1.
      IF( AV.LT.0.)    T=  -1.
      UU=U*U
      UV=U*V
      VV=V*V
      QQ=UU+VV
      AA= AA0 -.2*QQ
      AB=A1(I)*B1(J)
      GII=(  G(I+1,J)-G(I,J)-G(I,J)+G(I-1,J))*DD    +A3(I)*GI
      GIJ  =G(I+1,J+1) -G(I-1,J+1) -G(I+1,J-1) +G(I-1,J-1)
      GJJ=   (G(I,J+1) -G(I,J) -G(I,J) +G(I,J-1))*EE  + B3(J)*GJ
      R=               -(AA  -UU)*S2(I)*B1(J)* GJ
     1  +CA*(VV-UU)-2.*UV*SA
     2       + QQ*(U*A0(I)+V*Y)*DH
      IF(QQ.GE.AA) GO TO 33
      AXX          = (AA  -UU)*A2(I)
      AXY=-(AA*S1(I)+U*AV)*2.*AB
      AYY=(AA*FX-AV*AV)*B2(J)
      R=AXX*GII+AXY*GIJ+AYY*GJJ+R
      AI= -2.*AXX*DD  -Q1*AYY*EE
      BI= AXX*DD
      CI= AXX*DD
      YI=-R
      GO TO 35
   33 NS           = NS  +1
      K            = S
      IM           = I  -K
      IMM          = IM  -K
      L=   T
      JM= J-L
      JMM= JM-L
      AQ           = AA/QQ
      BXX=VV*A2(I)
      BXY= -2.*AB*V*AU
      BYY= AU*AU*B2(J)
      GNN=BXX*GII+BXY*GIJ+BYY*GJJ
      IF( IMM.LT.2.OR.IMM.GT.I4) GO TO 66
      GIIM=   (G(I,J) -G(IM,J) -G(IM,J) +G(IMM,J))*DD    +A3(I)*GI
      GO TO 67
   66 GIIM= GII
   67 GIJM=   G(I,J)  -G(IM,J)   -G(I,JM)   +G(IM,JM)
      IF( JMM.LT.2.OR.JMM.GT.J4) GO TO 64
      GJJM=   (G(I,J) -G(I,JM) -G(I,JM) +G(I,JMM)  )*EE  + B3(J)*GJ
      GO TO 65
   64 GJJM=   GJJ
   65 AXX          = UU*A2(I)
      AXY=8.*S*T*U*AV*AB
      AYY=AV*AV*B2(J)
      GSS=AXX*GIIM+AXY*GIJM+AYY*GJJM
      R            = (AQ  -1.)*GSS  +AQ*GNN  +R
      BB= (AQ-1.)*(AXX*DD + .5*AXY)
      AI= AQ*(-Q2*BYY*EE  -2.*BXX*DD)
     1      +(AQ-1.)* 2.* (AXX*DD  + AYY*EE  +AXY)
      BI= AQ*BXX*DD-(1.+S)*BB
      CI= AQ*BXX*DD  -(1.-S)*BB
      YI=-R
```

```
 35 IF(ABS(R).LE.RR) GO TO 37
    RR=ABS(R)
    IR=I
    JR=J
 37 A= 1./(AI-BI*C(I-1))
    C(I)= CI*A
    D(I)= (YI-BI*D(I-1))*A
 32 CONTINUE
    I= I3
    CG=0.
    DO 42 M= I0,I3
    CG          = D(I)   -C(I)*CG
    IF (ABS(CG).LE.ABS(RG)) GO TO 43
    RG=ABS(CG)
    IG=I
    JG=J
 43 G(I,J)=    G(I,J) +CG
 42 I           = I   -1
    J=J-1
    IF( J.GE.J1) GO TO 21
    TTAU= G(IX2,3)-G(IX1,3)
    IF( KSYM.LE.0) TAU= TAU+ P3*(TTAU-TAU)
    DO 52 I=IX1,IX2
    GI= G(I+1,3)-G(I-1,3)
    U0          = A1(I)* GI   +CA*A0(I)   +SA*S0(I)
    BIS=    B1( 3)*(1. +S1(I)**2)
    G(I,2)= G(I,4)      -(CA*S0(I)-SA*A0(I)+U0*S1(I))/BIS
 52 CONTINUE
    N           = IX1   -1
    DO 62 I= I0,N
    M= NX+ 4 -I
 62 G(M,2)= G(I,4)   +TAU
    N           = IX2   +1
    DO 164 I= N,I3
    M= NX+ 4 -I
164 G(M,2)= G(I,4)-TAU
    IF( FMACH.LT.1.) GO TO 91
    DO 82 J= J1,J4
    G(I1,J)= 3.*(G(I0,J)-G(I1,J)) + G(II0,J)
 82 G(I4,J)= 3.*(G(I3,J)-G(I2,J)) + G(II3,J)
    RETURN
 91 DO 92 J= J1,J3
    G(I1,J)= -.5*TAU
 92 G(I4,J)= .5*TAU
    G(I1,J4)= -.25*TAU
    G(I4,J4)= .25*TAU
    RETURN
    END
```

```
    SUBROUTINE  ESTIM
    INITIAL ESTIMATE OF POTENTIAL
```

```
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),S0(132),S1(132),S2(132.
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/H/ DX,DY,DT,DXX,DYY,DTT,DXY,DXT,DYT,TSR
      COMMON/J/ RAD,PI,ALS,ALT,ALTT,AMPLA,FREQRA,FASAGA,FMACHS,FMACHT
     1 ,AMPLM,FREQRM,FASAGM,CETAS,CETAT,CETATT,AMPLC,FREQRC,FASAGC,CETA
     2 ,FREQR,IPSURE
      COMMON/STADI/ RR,IR,JR,IRSTAD
      COMMON/WAKE/ NIT,WG(132)
      KX= NX+1
      MY        = NY  +2
      CB= COS(ALPHA)
      SB= SIN(ALPHA)
      CA= FMACH*CB
      SA= FMACH*SB
      IF(IRSTAD.GT.0) GO TO 11
      DO 12 I= 3,KX
      DO 12 J= 3,MY
      GM(I,J)= 0.
      GN(I,J)=0.
      G(I,J)= 0.
   12 CONTINUE
   11 DO 22 I= IX1,IX2
      Y=S0(I)+B0(3)
      X= A0(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      VBN=HH*(XT*S1(I) -YT)
      GI= G(I+1,3)-G(I-1,3)
      FX=1.+S1(I)**2
      BIS= FX*B1(3)
      GXSXVB= A1(I)*GI*S1(I)+VBN
      G(I,2) = G(I,4) -GXSXVB/BIS
      G(I,1)= G(I,5) -2.*GXSXVB/BIS
   22 CONTINUE
      DO 23 I= IX2,KX
      M= NX+4 -I
   23 WG(I)= G(I,3)-G(M,3)
      HMIN= 10.
      DO 13 I= 3,KX
      DO 13 J= 3,MY
      Y=S0(I)+B0(J)
      HH=Y*Y+A0(I)*A0(I)
      H=SQRT(HH)
      HX= .5*H/A1(I)
      HY= .5*H/B1(J)
      HMIN= AMIN1(HMIN,HX,HY)
   13 CONTINUE
      DT= HMIN*TSR
      IDT= 2.*PI/(FREQR*DT) + 1.
      IDT= IDT/IPSURE + 1.
      DT= 2.*PI/(IDT*IPSURE*FREQR)
      DTT=DT*DT
      DXT=DX*DT
      DYT=DY*DT
      RETURN
      END



      SUBROUTINE  REFIN
C     HALVES MESH SIZE
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),SO(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/H/ DX,DY,DT,DXX,DYY,DTT,DXY,DXT,DYT,TSR
      COMMON/J/ RAD,PI,ALS,ALT,ALTT,AMPLA,FREQRA,FASAGA,FMACHS,FMACHT
     1 ,AMPLM,FREQRM,FASAGM,CETAS,CETAT,CETATT,AMPLC,FREQRC,FASAGC,CETA
     2 ,FREQR,IPSURE
      COMMON/WAKE/ NIT,WG(132)
      DDT= DT
      HMIN= 10.
      KX= NX+1
      MX= NX+2
      KY         = NY  +1
      MY=          NY+2
      DO 13 I= 3,KX
      DO 13 J= 3,MY
      Y=SO(I)+B0(J)
      HH=Y*Y+A0(I)*A0(I)
      H=SQRT(HH)
      HX= .5*H/A1(I)
      HY= .5*H/B1(J)
      HMIN= AMIN1(HMIN,HX,HY)
   13 CONTINUE
      DT= HMIN*TSR
      IDT= 2.*PI/(FREQR*DT) + 1.
      IDT= IDT/IPSURE + 1.
      DT= 2.*PI/(IDT*IPSURE*FREQR)
      DT= AMIN1(DT,DDT)
   14 RATIO= DT/DDT
      DTT=DT*DT
      DXT=DX*DT
      DYT=DY*DT
      IY= NY+3
      LX= NX/2 + 2
      LY= NY/2 +3
      DO 22 K= 2,LX
      I= LX+2-K
      II= (I-2)*2 +2
      DO 22 KK= 3,LY
      J= LY+3-KK
      JJ= (J-3)*2 +3
      GM(II,JJ)= GM(I,J)*RATIO
   22 G(II,JJ)=       G(I,J)
      DO 42 I= 2,MX,2
      DO 42 J= 4,MY,2
      GM(I,J)= .5*(GM(I,J+1) + GM(I,J-1))
   42 G(I,J)     = .5*(G(I,J+1)  +G(I,J-1))
      DO 32 J= 3,IY
      DO 32 I= 3,KX,2
      GM(I,J)= .5*(GM(I+1,J) + GM(I-1,J))
   32 G(I,J)     = .5*(G(I+1,J)  +G(I-1,J))
      DO 33 I= IX2,KX
      M= NX+4-I
   33 WG(I)= G(I,3)-G(M,3)
      DO 52 I=IX1,IX2
      Y=SO(I)+BO(3)
      X= AO(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      VBN=HH*(XT*S1(I) -YT)
      GI= G(I+1,3)-G(I-1,3)
      FX=1.+S1(I)**2
      BIS= FX*B1(3)
      GXSXVB= A1(I)*GI*S1(I)+VBN
      G(I,2) = G(I,4) -GXSXVB/BIS
      G(I,1)= G(I,5) -2.*GXSXVB/BIS
      GM(I,2)= GM(I,4)
   52 CONTINUE
      N            = IX1  -1
      DO 62 I= 3,N
      M= NX+4 -I
      GM(M,2)= GM(I,4)
   62 G(M,2)= G(I,4) + WG(M)
      N            = IX2  +1
      DO 64 I= N,KX
      M= NX+4 -I
      GM(M,2)= GM(I,4)
   64 G(M,2)= G(I,4)  -WG(I)
      RETURN
      END




      SUBROUTINE VELO
C     CALCULATES SURFACE VELOCITY
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),SO(132),S1(132),S2(132)
     1          ,AO(132),A1(132),A2(132),A3(132),BO(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/B/ SV(132),SM(132),CP(132)
      COMMON/H/ DX,DY,DT,DXX,DYY,DTT,DXY,DXT,DYT,TSR
      COMMON/J/ RAD,PI,ALS,ALT,ALTT,AMPLA,FREQRA,FASAGA,FMACHS,FMACHT
     1 ,AMPLM,FREQRM,FASAGM,CETAS,CETAT,CETATT,AMPLC,FREQRC,FASAGC,CETA
     2 ,FREQR,IPSURE
      COMMON/K/ I0,I1,I2,I3,J1,J2,J3
      DDT= 1./DT
      AA0 = 1.
      DO 12 I=IX1,IX2
      Y=B0(3)+S0(I)
      X= A0(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      H= SQRT(HH)
      DH= 1./H
      GI= G(I+1,3)-G(I-1,3)
      GJ= G(I,4)-G(I,2)
      GX          = A1(I)* GI  -S1(I)*B1(3)* GJ
      GY= B1(3)*GJ
      U= GX*DH
      V= GY*DH
      QQ=U*U+V*V
      UR= XT*H+ U
      VR= YT*H + V
      QQR= UR*UR + VR*VR
      QR= SQRT(QQR)
      IF(UR.LT.0.) QR= -QR
      SV(I) = QR
      CHAIN= XT*GX + YT*GY
      FIT= GM(I,3) *DDT + CHAIN
      AA= AA0 -.2*QQ-.4*FIT
      AA= ABS(AA)
      A= SQRT(AA)
      SM(I)= QR/A
   12 CP(I)=    (AA**3.5  -1.)/(.7*FMACH2)
      RETURN
      END



      SUBROUTINE FORCE
C     CALCULATES FORCE COEFFICIENTS
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),S0(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,U1IM,CB,SB,NS,RG,IG,JG
      COMMON/B/ SV(132),SM(132),CP(132)
      COMMON/C/ XP(260),YP(260),D1(260),D2(260),D3(260)
      COMMON/E/ CHORD,XM,CL,CD,CM
      COMMON/L/ TCL(801),TCD(801),TCM(801),TCP(9,801),TCPS(9),CLS,CDS
     1          ,CMS,NITS,IJUMP,NSTEP,JSTEP,PERIOD,MHALF
      COMMON/WAKE/ NIT,WG(132)
      CL          = 0.
      CD          = 0.
      CM          = 0.
      N=IX2-1
      DO 12 I=IX1,N
      DX=(XP(I+1)-XP(I))/CHORD
      DY=(YP(I+1)-YP(I))/CHORD
      XA=(.5*(XP(I+1)+XP(I))-XM)/CHORD
      YA=.5*(YP(I+1)+YP(I))/CHORD
      CPA         = .5*(CP(I+1)  +CP(I))
      DCL         = -CPA*DX
      DCD         = CPA*DY
      CL          = CL  +DCL
      CD          = CD  +DCD
   12 CM          = CM  +DCD*YA  -DCL*XA
      DCL         = CL*COS(ALPHA)  -CD*SIN(ALPHA)
      CD          = CL*SIN(ALPHA)  +CD*COS(ALPHA)
      CL          = DCL
      IF(NIT.NE.0) RETURN
      IF(NITS.EQ.0) GO TO 11
      IJUMP= 2**MHALF
      CLS= CL
      CDS= CD
      CMS= CM
      ISP= (IX1+IX2)*.5
      I= 1
   10 TCPS(I)= CP(ISP)
      IF(ISP.GE.IX2) GO TO 11
      I= I+1
      ISP= ISP+ IJUMP
      GO TO 10
   11 NITS= 0
      JJSTEP= JSTEP + 1
      TCL(JJSTEP)= CL-CLS
      TCD(JJSTEP) = CD -CDS
      TCM(JJSTEP) = CM -CMS
      IJUMP= 2**MHALF
      ISP= (IX1 + IX2)* .5
      I= 1
   13 TCP(I,JJSTEP)= CP(ISP)  -TCPS(I)
      IF(ISP.GE.IX2) GO TO 14
      I= I+1
      ISP= ISP + IJUMP
      GO TO 13
   14 RETURN
      END


      SUBROUTINE CPLOT
C     PLOTS CP AT EQUAL INTERVALS IN THE MAPPED PLANE
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),S0(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
```

```
      COMMON/B/ SV(132),SM(132),CP(132)
      COMMON/C/ XP(260),YP(260),D1(260),D2(260),D3(260)
      COMMON/STADI/ RR,IR,JR,ISTADI
      DIMENSION  KODE(2),LINE(115)
      DATA          KODE/1H ,1H+/
C     CPO IS RESERVOR PRESSURE COEFFICIENT WHERE Q=0 AND FIT=0
      CPO= 0.
      IF(ISTADI.GT.0) CPO=((1.+.2*FMACH2)**3.5-1.)/(.7*FMACH2)
      I1=IX1
      I2=IX2
      IWRIT     = 6
      WRITE (IWRIT,2)
    2 FORMAT(50HOPLOT OF CP AT EQUAL INTERVALS IN THE MAPPED PLANE/
     1       9HO    X     ,9H      CP   )
      DO 12 I= 1,115
   12 LINE(I)    = KODE(1)
      DO 22 I=I1,I2
      K=    20.*(CPO-CP(I)) +55.0
      IF( K.LT.1)  K=1
      IF( K.GT.115) K=115
      LINE(K)    = KODE(2)
      WRITE (IWRIT,610)XP(I),CP(I),LINE
   22 LINE(K)    = KODE(1)
      RETURN
  610 FORMAT(1H ,2F9.4,115A1)
      END




      SUBROUTINE  CHART
C     GENERATES MACH NO CHART
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),S0(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/H/ DX,DY,DT,DXX,DYY,DTT,DXY,DXT,DYT,TSR
      COMMON/J/ RAD,PI,ALS,ALT,ALTT,AMPLA,FREQRA,FASAGA,FMACHS,FMACHT
     1 ,AMPLM,FREQRM,FASAGM,CETAS,CETAT,CETATT,AMPLC,FREQRC,FASAGC,CETA
     2 ,FREQR,IPSURE
      COMMON/K/ I0,I1,I2,I3,J1,J2,J3
      DIMENSION IND(150)
      DDT= 1./DT
      AAO = 1.
      IWRIT     = 6
      K         = NY/32
      IF (NY.GT.32*K) K = K   +1
      WRITE (IWRIT,2)
    2 FORMAT(14HOMACH NO CHART)
   11 DO 12 I= I0,I3
      J=J1
      N= 0
   14 N= N+1
      X= AC(I)
      Y=SO(I)+BO(J)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      H=SQRT(HH)
      DH= 1./H
      GI=G(I+1,J)-G(I-1,J)
      GJ=G(I,J+1)-G(I,J-1)
      GX          = A1(I)* GI   -S1(I)*B1(J)* GJ
      GY          =  B1(J)* GJ
      U= GX*DH
      V= GY*DH
      QQ=U*U+V*V
      CHAIN= XT*GX + YT*GY
      FIT= GM(I,J) *DDT + CHAIN
      AA= AAO -.2*QQ-.4*FIT
      AA= ABS(AA)
      UR= XT*H+ U
      VR= YT*H + V
      QQR= UR*UR + VR*VR
      QA=QQR/AA
      IND(N)= 100.*SQRT(QA)
      IF(UR.LT.O.) IND(N) = -IND(N)
      J= J+K
      IF(J.LE.J2) GO TO 14
   12 WRITE (IWRIT,610) (IND(J),J=1,N)
      RETURN
  610 FORMAT(1X,32I4)
      END




      SUBROUTINE PSURE
C     GENERATES PRESSURE DISTRIBUTION OVER AIRFOIL
C     AT EQUAL INTERVALS IN THE MAPPED PLANE
C     WITH THE ASSOCIATED SHOCKS
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),S0(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/B/ SV(132),SM(132),CP(132)
      COMMON/C/ XP(260),YP(260),D1(260),D2(260),D3(260)
      COMMON/E/ CHORD,XM,CL,CD,CM
      COMMON/G/ TITLE(20),IPLOT
      COMMON/J/ RAD,PI,ALS,ALT,ALTT,AMPLA,FREQRA,FASAGA,FMACHS,FMACHT
     1 ,AMPLM,FREQRM,FASAGM,CETAS,CETAT,CETATT,AMPLC,FREQRC,FASAGC,CETA
     2 ,FREQR,IPSURE
      DIMENSION X(260),Y(260),R(150)
      IF (IPLOT) 1,11,101
    1 CALL PLOTSBL(5000,23HI-CHUNG CHANG    109104W)
   11 CALL PLOT(2.5,2.00,-3)
      CALL SYMBOL(-2.0,-1.50,.07,3,0.,-1)
      CALL SYMBOL(6.5,-1.50,.07,3,0.,-1)
      ENCODE(48,12,R) TITLE
   12 FORMAT(12A4)
      CALL SYMBOL(-.5,-.75,.14,R,0.,48)
      FAA= FASAGA*180./PI
      FAM= FASAGM*180./PI
      FAC= FASAGC*180./PI
      ENCODE(57,14,R) UTIM,FAA,FAM,FAC
   14 FORMAT(5HTIME=,F7.2,3X,5HALFA=,F7.2,3X,5HM FA=,F7.2,3X,5HCEFA=,
     1      F7.2)
      CALL SYMBOL(-.5,-1.0,.14,R,0.,57)
      ENCODE(42,15,R) AL,FMACH,CETA
   15 FORMAT(5HAL   =,F7.3,3X,5HM    =,F7.3,3X,5HCETA=,F7.3)
      CALL SYMBOL(-.5,-1.25,.14,R,0.,42)
      ENCODE(42,16,R) CL,CD,CM
   16 FORMAT(5HCL   =F7.4,3X,5HCD   =,F7.4,3X,5HCM   =,F7.4)
      CALL SYMBOL(-.5,-1.50,.14,R,0.,42)
      XMAX=XP(IX1)
      XMIN=XP(IX1)
      DO 22 I=IX1,IX2
      XMAX=AMAX1(XP(I),XMAX)
   22 XMIN        =AMIN1(XP(I),XMIN)
      SCALE       = 5./(XMAX  -XMIN)
      DO 24 I= IX1,IX2
      X(I)=SCALE*(XP(I)-XMIN)
   24 Y(I)=SCALE*YP(I)
      N= IX2-IX1+1
      CALL LINE(X(IX1),Y(IX1),N,1,0,1,0.,1.,0.,1.)
      CPMAX= 0.
      IMAX= IX1
      DO 25 I= IX1,IX2
      ABSCP= ABS(CP(I))
      IF(ABSCP.LE.CPMAX) GO TO 25
      CPMAX=ABSCP
      IMAX= I
   25 CONTINUE
      CALL PLOT(0.,4.25,-3)
      CALL AXIS(-1.,-4.,2HCP,2,8.,90.,1.6,-.4,0)
C     CPC IS CRITICAL PRESSURE COEFFICIENT
      AA= (1.+.2*FMACH2)/1.2
      CPC=(AA**3.5-1.)/(.7*FMACH2)
      IF( CPC.GE.-1.6.AND.CPC.LE.1.6)
     1CALL SYMBOL(-1.,-2.50*CPC,.4,15,0.,-1)
      DO 32 I= IX1,IMAX
      IF(CP(I).GT.1.6) GO TO 32
      IF(CP(I).LT.-1.6) GO TO 32
      CALL SYMBOL(X(I),-2.50*CP(I),.07,3,45.,-1)
   32 CONTINUE
      DO 34 I= IMAX,IX2
      IF(CP(I).GT.1.6) GO TO 34
      IF(CP(I).LT.-1.6) GO TO 34
      CALL SYMBOL(X(I),-2.50*CP(I),.07,3,0.,-1)
   34 CONTINUE
      CALL SYMBOL(-2.0,-5.75,.07,3,0.,-1)
      CALL SYMBOL(6.5,-5.75,.07,3,0.,-1)
      CALL PLOT(-2.5,-6.25,-3)
      CALL FRAME(1)
      RETURN
  101 CALL PLOT(0.,0.,999)
      RETURN
      END



      SUBROUTINE LORD
C     GENERATES THE LOADING DISTRIBUTION OVER AIRFOIL
C     AT EQUAL INTERVALS IN THE MAPPED PLANE
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),S0(132),S1(132),S2(132)
     1         ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2         ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3         ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/B/ SV(132),SM(132),CP(132)
      COMMON/C/ XP(260),YP(260),D1(260),D2(260),D3(260)
      COMMON/E/ CHORD,XM,CL,CD,CM
      COMMON/F/ XR,YR,KS,XS(500),YS(500)
      COMMON/G/ TITLE(20),IPLOT
      COMMON/J/ RAD,PI,ALS,ALT,ALTT,AMPLA,FREQRA,FASAGA,FMACHS,FMACHT
     1 ,AMPLM,FREQRM,FASAGM,CETAS,CETAT,CETATT,AMPLC,FREQRC,FASAGC,CETA
     2 ,FREQR,IPSURE
      DIMENSION X(260),Y(260),R(150),DCP(132)
      I1=IX1
      I2=IX2
      IF (IPLOT) 1,11,101
    1 CALL PLOTSBL(5000,23HI-CHUNG CHANG   109104W)
   11 CALL PLOT(2.5,2.00,-3)
      CALL SYMBOL(-2.0,-1.50,.07,3,0.,-1)
      CALL SYMBOL(6.5,-1.50,.07,3,0.,-1)
      ENCODE(48,12,R) TITLE
   12 FORMAT(12A4)
      CALL SYMBOL(-.5,-.75,.14,R,0.,48)
      FAA= FASAGA*180./PI
      FAM= FASAGM*180./PI
      FAC= FASAGC*180./PI
      ENCODE(57,14,R) UTIM,FAA,FAM,FAC
   14 FORMAT(5HTIME=,F7.2,3X,5HALFA=,F7.2,3X,5HM FA=,F7.2,3X,5HCEFA=,
     1      F7.2)
      CALL SYMBOL(-.5,-1.0,.14,R,0.,57)
      ENCODE(42,15,R) AL,FMACH,CETA
   15 FORMAT(5HAL   =,F7.3,3X,5HM    =,F7.3,3X,5HCETA=,F7.3)
      CALL SYMBOL(-.5,-1.25,.14,R,0.,42)
      ENCODE(42,16,R) CL,CD,CM
   16 FORMAT(5HCL   =F7.4,3X,5HCD   =,F7.4,3X,5HCM   =,F7.4)
      CALL SYMBOL(-.5,-1.50,.14,R,0.,42)
      XMAX=XP(I1)
      XMIN=XP(I1)
      DO 22 I=I1,I2
      XMAX=AMAX1(XP(I),XMAX)
   22 XMIN       =AMIN1(XP(I),XMIN)
      SCALE      = 5./(XMAX  -XMIN)
      DO 24 I=I1,I2
      X(I)=SCALE*(XP(I)-XMIN)
   24 Y(I)=SCALE*YP(I)
      INOSE= .5*(IX1 + IX2)
      DO 26 I= INOSE,IX2
      M= NX + 4 - I
      DCP(I)  =  CP(M)-CP(I)
   26 CONTINUE
      N          = I2  -I1  +1
      CALL LINE(X(I1),Y(I1),N,1,0,1,0.,1.,0.,1.)
      CALL PLOT(0.,4.25,-3)
      CALL AXIS(-1.,-4.,3HDCP,3,8.,90.,-1.6,.4,0)
      DO 32 I=INOSE,I2
      IF(DCP(I).GT.1.6) GO TO 32
      IF(DCP(I).LT.-1.6) GO TO 32
      CALL SYMBOL(X(I), 2.5*DCP(I),.07,3,0.,-1)
   32 CONTINUE
      CALL SYMBOL(-2.0,-5.75,.07,3,0.,-1)
      CALL SYMBOL(6.5,-5.75,.07,3,0.,-1)
      CALL PLOT(-2.5,-6.25,-3)
      CALL FRAME(1)
      RETURN
  101 CALL PLOT(0.,0.,999)
      RETURN
      END



      SUBROUTINE SONIC
C     GENERATES SONIC LINE OVER AIRFOIL
C     RENAME COMMON/A/
C     THE POSITION OF GN IS OVERLAPPED BY SHOCK
      COMMON/A/ GM(132,36),G(132,36),SHOCK(132,36)
     *          ,S0(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/B/ SV(132),SM(132),CP(132)
      COMMON/C/ XP(260),YP(260),D1(260),D2(260),D3(260)
      COMMON/D/ SLOPT,TRAIL,SCAL
      COMMON/E/ CHORD,XM,CL,CD,CM
      COMMON/F/ XR,YR,KS,XS(500),YS(500)
      COMMON/G/ TITLE(20),IPLOT
      COMMON/H/ DX,DY,DT,DXX,DYY,DTT,DXY,DXT,DYT,TSR
      COMMON/J/ RAD,PI,ALS,ALT,ALTT,AMPLA,FREQRA,FASAGA,FMACHS,FMACHT
     1 ,AMPLM,FREQRM,FASAGM,CETAS,CETAT,CETATT,AMPLC,FREQRC,FASAGC,CETA
     2 ,FREQR,IPSURE
      COMMON/K/ I0,I1,I2,I3,J1,J2,J3
      DIMENSION XA(260),YA(260),R(150)
      DIMENSION Q(260)
      IF (IPLOT) 1,11,101
  1 CALL PLOTSBL(5000,23HI-CHUNG CHANG   109104W)
 11 CALL PLOT(2.5,2.00,-3)
    CALL SYMBOL(-2.0,-1.50,.07,3,0.,-1)
    CALL SYMBOL(6.5,-1.50,.07,3,0.,-1)
    ENCODE(48,12,R) TITLE
 12 FORMAT(12A4)
    CALL SYMBOL(-.5,-.75,.14,R,0.,48)
    FAA= FASAGA*180./PI
    FAM= FASAGM*180./PI
    FAC= FASAGC*180./PI
    ENCODE(57,14,R) UTIM,FAA,FAM,FAC
 14 FORMAT(5HTIME=,F7.2,3X,5HALFA=,F7.2,3X,5HM FA=,F7.2,3X,5HCEFA=,
   1       F7.2)
    CALL SYMBOL(-.5,-1.0,.14,R,0.,57)
    ENCODE(42,15,R) AL,FMACH,CETA
 15 FORMAT(5HAL   =,F7.3,3X,5HM    =,F7.3,3X,5HCETA=,F7.3)
    CALL SYMBOL(-.5,-1.25,.14,R,0.,42)
    ENCODE(42,16,R) CL,CD,CM
 16 FORMAT(5HCL   =F7.4,3X,5HCD   =,F7.4,3X,5HCM   =,F7.4)
    CALL SYMBOL(-.5,-1.50,.14,R,0.,42)
C     AIRFOIL
    XMAX= XP(IX1)
    XMIN= XP(IX1)
    DO 22 I= IX1,IX2
    XMAX=AMAX1(XP(I),XMAX)
 22 XMIN       =AMIN1(XP(I),XMIN)
    SCALE      = 5./(XMAX  -XMIN)
    DO 24 I= IX1,IX2
    XA(I) = SCALE*(XP(I)-XMIN)
 24 YA(I)= SCALE*YP(I)
    CALL PLOT(0.,4.,-3)
    N= IX2-IX1+1
    CALL LINE(XA(IX1),YA(IX1),N,1,0,1,0.,1.,0.,1.)
    AA0= 1.
    DDT= 1./DT
    DO 2 I= I0,I3
    DO 2 J= J1,J3
    X= A0(I)
    Y=S0(I)+B0(J)
    HH= X*X + Y*Y
    DHH= 1./HH
    XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
    YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
    H=SQRT(HH)
    DH= 1./H
    GI=G(I+1,J)-G(I-1,J)
    GJ=G(I,J+1)-G(I,J-1)
    GX         = A1(I)* GI  -S1(I)*B1(J)* GJ
    GY         =  B1(J)* GJ
    U= GX*DH
    V= GY*DH
    UU=U*U
    VV=V*V
    QQ=UU+VV
```

```
      UR= XT*H+ U
      VR= YT*H + V
      UUR= UR*UR
      VVR= VR*VR
      QQR= UUR + VVR
      CHAIN= XT*GX + YT*GY
      FIT= GM(I,J) *DDT + CHAIN
      AA=AA0-.2*QQ-.4*FIT
      AA=AMAX1(AA,.0001)
      SHOCK(I,J) = SQRT(QQR/AA) - 1.
    2 CONTINUE
C     LOCATES THE SONIC POINTS
      KS= 0
      DO 17 J= J1,J3
   17 Q(J) = SHOCK(I0,J)
      DO 18 I= I1,I2
      DO 18 K= J1,J2
      J= J1+J2-K
      QQ= SHOCK(I,J)
      IF(QQ.NE.0.) GO TO 19
      KS= KS +1
      XS(KS) = A0(I)
      YS(KS) = S0(I) + B0(J)
      GO TO 18
   19 IF(QQ*Q(J+1).GE.0.) GO TO 20
      RT=ABS(QQ/(QQ-Q(J+1)))
      KS= KS +1
      XS(KS) = A0(I)
      YS(KS)= S0(I)+BC(J) +RT*(B0(J+1)-B0(J))
   20 IF(QQ*Q(J).GE.0.) GO TO 18
      RT=ABS(QQ/(QQ-Q(J)))
      KS= KS + 1
      XS(KS)= A0(I) +RT*(A0(I-1)-A0(I))
      YS(KS)=S0(I)+B0(J)+RT*(S0(I-1)-S0(I))
   18 Q(J)= QQ
      DO 21 I= 1,KS
      XX= .5*SCAL*(XS(I)**2-YS(I)**2) + XR
      YS(I) = SCAL*XS(I)*YS(I) + YR
   21 XS(I) = XX
      DO 52 I= 1,KS
      XX= SCALE*(XS(I)-XMIN)
      YY= SCALE*YS(I)
      IF(XX.LT.-2.0) GO TO 52
      IF(XX.GT.6.5) GO TO 52
      IF(ABS(YY).GT.4.5) GO TO 52
      CALL SYMBOL(XX,YY,.07,3,0.,-1)
   52 CONTINUE
   70 CALL SYMBOL(-2.0,-5.5 ,.07,3,0.,-1)
      CALL SYMBOL(6.5,-5.5 ,.07,3,0.,-1)
      CALL PLOT(-2.5,-6.0,-3)
      CALL FRAME(1)
      RETURN
  101 CALL PLOT(0.,0.,999)
      RETURN
      END
      SUBROUTINE TRACE
C     GENERATES UNSTEADY TRACES OF AIRFOIL
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),S0(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
C     RENAME COMMON/C/
      COMMON/C/ XP(260),YP(260),D1(260),Y(260),R(260)
      COMMON/G/ TITLE(20),IPLOT
      COMMON/H/ DX,DY,DT,DXX,DYY,DTT,DXY,DXT,DYT,TSR
      COMMON/J/ RAD,PI,ALS,ALT,ALTT,AMPLA,FREQRA,FASAGA,FMACHS,FMACHT
     1 ,AMPLM,FREQRM,FASAGM,CETAS,CETAT,CETATT,AMPLC,FREQRC,FASAGC,CETA
     2 ,FREQR,IPSURE
      COMMON/L/ TCL(801),TCD(801),TCM(801),TCP(9,801),TCPS(9),CLS,CDS
     1          ,CMS,NITS,IJUMP,NSTEP,JSTEP,PERIOD,MHALF
      DIMENSION X(801),AOA(801),FS(801),FA(801)
      IF (IPLOT) 1,11,101
    1 CALL PLOTSBL(5000,23HI-CHUNG CHANG   109104W)
   11 CALL PLOT(2.5,2.00,-3)
      CALL SYMBOL(-2.0,-1.50,.07,3,0.,-1)
      CALL SYMBOL(6.5,-1.50,.07,3,0.,-1)
C     TITLE
      ENCODE(48,12,R) TITLE
   12 FORMAT(12A4)
      CALL SYMBOL(-.5,-.50,.14,R,0.,48)
      ENCODE(58,14,R)
   14 FORMAT(40HUNSTEADY TRACES OF AIRFOIL IN SINUSOIDAL,
     1       18H RIGID BODY MOTION)
      CALL SYMBOL(-.5,-.75,.14,R,0.,58)
      AOAS= ALS*RAD
      ENCODE(57,15,R)AOAS,AMPLA,FREQRA
   15 FORMAT(18HMEAN ATTACK ANGLE=,F5.2,5X,4HAMP=,F5.2,5X,
     1       10HFREQ RATE=,F5.2)
      CALL SYMBOL(-.5,-1.00,.14,R,0.,57)
      ENCODE(57,16,R) FMACHS,AMPLM,FPEQRM
   16 FORMAT(18HMEAN FLIGHT SPEED=,F5.2,5X,4HAMP=,F5.2,5X,
     1       10HFREQ RATE=,F5.2)
      CALL SYMBOL(-.5,-1.25,.14,R,0.,57)
      FANGLE= CETAS*RAD
      ENCODE(57,17,R) FANGLE,AMPLC,FREQRC
   17 FORMAT(18HMEAN FLIGHT ANGLE=,F5.2,5X,4HAMP=,F5.2,5X,
     1       10HFREQ RATE=,F5.2)
      CALL SYMBOL(-.5,-1.50,.14,R,0.,57)
C     AIRFOIL
      I1 = IX1
      I2= IX2
      XMAX=XP(I1)
      XMIN=XP(I1)
      DO 52 I= I1,I2
      XMAX=AMAX1(XP(I),XMAX)
   52 XMIN       =AMIN1(XP(I),XMIN)
      SCALE      = 3./(XMAX  -XMIN)
      DO 54 I=I1,I2
      X(I)=SCALE*(XP(I)-XMIN)
```

```
54 Y(I)=SCALE*YP(I)
   N          = 12   -I1   +1
   CALL PLOT(2.,-.25,-3)
   CALL LINE(X(I1),Y(I1),N,1,0,1,0.,1.,0.,1.)
C       DRAWS PRESSURE SENSORS ON THE AIRFOIL
   IJUMP= 2**MHALF
   ISP= (IX1+IX2)*.5
   IPOINT= 1
19 CALL SYMBOL(X(ISP),Y(ISP),.07,3,0.,-1)
   CALL SYMBOL(X(ISP),Y(ISP),.07,3,45.,-1)
   ENCODE(1,18,R) IPOINT
18 FORMAT(I1)
   CALL SYMBOL(X(ISP),.1+Y(ISP),.07,R,0.,1)
   IF(IPOINT.GE.9) GO TO 20
   IF(ISP.GE.IX2) GO TO 20
   IPOINT= IPOINT + 1
   ISP= ISP + IJUMP
   GO TO 19
C       UNSTEADY PRESSURE TRACES ON THE PRESSURE SENSORS
20 XSCAL= (6.*FREQR)/(2.*PI*PERIOD)
   IF(AMPLA.EQ.0.) GO TO 55
   ASCAL=.2/AMPLA
   GO TO 56
55 ASCAL= 0.
56 IF(AMPLM.EQ.0.) GO TO 57
   FSCAL= .2/AMPLM
   GO TO 58
57 FSCAL= 0.
58 IF(AMPLC.EQ.0.) GO TO 59
   CSCAL= .2/AMPLC
   GO TO 60
59 CSCAL= 0.
60 JS= JSTEP + 1
   DO 2 I=1,JS
   TIME= DT*NSTEP*(I-1)
   X(I)= XSCAL*TIMF
   AOA(I)= ASCAL*AMPLA*SIN(TIME*FREQRA)
   FS(I)= FSCAL*AMPLM*SIN(FREQRM*TIME)
   FA(I)=CSCAL*AMPLC*SIN(FREQRC*TIME)
 2 CONTINUE
   CALL PLOT(-2.5,.75,-3)
   CALL AXIS(0.,0.,11H              ,-11,6.,0.,0.,.60*PERIOD,0)
   ENCODE(11,50,R)
50 FORMAT(11HPHASE ANGLE)
   CALL SYMBOL(0.,-.5,.14,R,0.,11)
   CALL PLOT( 0.,0.,3)
   CALL PLOT(0.,8.,2)
   CALL PLOT(0.,.5,-3)
   CALL SYMBOL(0.,0.,.07,15,0.,-1)
   CALL SYMBOL(-.75,0.,.14,6HAANGLE,0.,6)
   CALL PLOT( 0.,0.,3)
   DO 3 I= 1,JS
   CALL PLOT(X(I),AOA(I),2)
 3 CONTINUE
```

```
      CALL PLOT(0.,.5,-3)
      CALL SYMBOL(0.,0.,.07,15,0.,-1)
      CALL SYMBOL(-.75,0.,.14,6HFANGLE,0.,6)
      CALL PLOT( 0.,0.,3)
      DO 4 I= 1,JS
      CALL PLOT(X(I),FA(I),2)
    4 CONTINUE
      CALL PLOT(0.,.5,-3)
      CALL SYMBOL(0.,0.,.07,15,0.,-1)
      CALL SYMBOL(-.75,0.,.14,6HFSPEED,0.,6)
      CALL PLOT( 0.,0.,3)
      DO 5 I= 1,JS
      CALL PLOT(X(I),FS(I),2)
    5 CONTINUE
      IF(KSYM.GT.0.AND.ALS.EQ.0..AND.AMPLA.EQ.0..AND.AMPLC.EQ.0.) GOTO76
      TCMMAX= 0.
      DO 71 I= 1,JS
      ABSTCM= ABS(TCM(I))
      IF(ABSTCM.LE.TCMMAX) GO TO 71
      TCMMAX= ABSTCM
   71 CONTINUE
      TCMCAL= .2/TCMMAX
      GO TO 74
   76 TCMCAL=0.
   74 CALL PLOT(0.,.5,-3)
      CALL SYMBOL(0.,0.,.07,15,0.,-1)
      CALL SYMBOL(-.28,0.,.14,2HCM,0.,2)
      CALL PLOT( 0.,0.,3)
      DO 6 I= 1,JS
      CALL PLOT(X(I),TCMCAL*TCM(I),2)
    6 CONTINUE
      TCDMAX= 0.
      DO 72 I= 1,JS
      ABSTCD= ABS(TCD(I))
      IF(ABSTCD.LE.TCDMAX) GO TO 72
      TCDMAX= ABSTCD
   72 CONTINUE
      TCDCAL= .2/TCDMAX
      CALL PLOT(0.,.5,-3)
      CALL SYMBOL(0.,0.,.07,15,0.,-1)
      CALL SYMBOL(-.28,0.,.14,2HCD,0.,2)
      CALL PLOT( 0.,0.,3)
      DO 7 I= 1,JS
      CALL PLOT(X(I),TCDCAL*TCD(I),2)
    7 CONTINUE
      IF(KSYM.GT.0.AND.ALS.EQ.0..AND.AMPLA.EQ.0..AND.AMPLC.EQ.0.) GOTO77
      TCLMAX= 0.
      DO 73 I= 1,JS
      ABSTCL= ABS(TCL(I))
      IF(ABSTCL.LE.TCLMAX) GO TO 73
      TCLMAX= ABSTCL
   73 CONTINUE
      TCLCAL= .2/TCLMAX
      GO TO 75
```

```
77 TCLCAL=0.
75 CALL PLOT(0.,.5,-3)
   CALL SYMBOL(0.,0.,.07,15,0.,-1)
   CALL SYMBOL(-.28,0.,.14,2HCL,0.,2)
   CALL PLOT( 0.,0.,3)
   DO 8 I= 1,JS
   CALL PLOT(X(I),TCLCAL*TCL(I),2)
 8 CONTINUE
   TCPMAX= 0.
   DO 30 K= 1,9
   DO 30 I= 1,JS
   ABSTCP= ABS(TCP(K,I))
   IF(ABSTCP.LE.TCPMAX) GO TO 30
   TCPMAX= ABSTCP
30 CONTINUE
   TCPCAL= .2/TCPMAX
   CALL PLOT(0.,.5,-3)
   CALL SYMBOL(0.,0.,.07,15,0.,-1)
   CALL SYMBOL(-.28,0.,.14,2HP1,0.,2)
   CALL PLOT( 0.,0.,3)
   DO 9 I= 1,JS
   CALL PLOT(X(I),TCPCAL*TCP(1,I),2)
 9 CONTINUE
   CALL PLOT(0.,.5,-3)
   CALL SYMBOL(0.,0.,.07,15,0.,-1)
   CALL SYMBOL(-.28,0.,.14,2HP2,0.,2)
   CALL PLOT( 0.,0.,3)
   DO 29 I= 1,JS
   CALL PLOT(X(I),TCPCAL*TCP(2,I),2)
29 CONTINUE
   CALL PLOT(0.,.5,-3)
   CALL SYMBOL(0.,0.,.07,15,0.,-1)
   CALL SYMBOL(-.28,0.,.14,2HP3,0.,2)
   CALL PLOT( 0.,0.,3)
   DO 41 I= 1,JS
   CALL PLOT(X(I),TCPCAL*TCP(3,I),2)
41 CONTINUE
   CALL PLOT(0.,.5,-3)
   CALL SYMBOL(0.,0.,.07,15,0.,-1)
   CALL SYMBOL(-.28,0.,.14,2HP4,0.,2)
   CALL PLOT( 0.,0.,3)
   DO 42 I= 1,JS
   CALL PLOT(X(I),TCPCAL*TCP(4,I),2)
42 CONTINUE
   CALL PLOT(0.,.5,-3)
   CALL SYMBOL(0.,0.,.07,15,0.,-1)
   CALL SYMBOL(-.28,0.,.14,2HP5,0.,2)
   CALL PLOT( 0.,0.,3)
   DO 43 I= 1,JS
   CALL PLOT(X(I),TCPCAL*TCP(5,I),2)
43 CONTINUE
   CALL PLOT(0.,.5,-3)
   CALL SYMBOL(0.,0.,.07,15,0.,-1)
   CALL SYMBOL(-.28,0.,.14,2HP6,0.,2)
```

```
      CALL PLOT( 0.,0.,3)
      DO 44 I= 1,JS
      CALL PLOT(X(I),TCPCAL*TCP(6,I),2)
   44 CONTINUE
      CALL PLOT(0.,.5,-3)
      CALL SYMBOL(0.,0.,.07,15,0.,-1)
      CALL SYMBOL(-.28,0.,.14,2HP7,0.,2)
      CALL PLOT( 0.,0.,3)
      DO 45 I= 1,JS
      CALL PLOT(X(I),TCPCAL*TCP(7,I),2)
   45 CONTINUE
      CALL PLOT(0.,.5,-3)
      CALL SYMBOL(0.,0.,.07,15,0.,-1)
      CALL SYMBOL(-.28,0.,.14,2HP8,0.,2)
      CALL PLOT( 0.,0.,3)
      DO 46 I= 1,JS
      CALL PLOT(X(I),TCPCAL*TCP(8,I),2)
   46 CONTINUE
      CALL PLOT(0.,.5,-3)
      CALL SYMBOL(0.,0.,.07,15,0.,-1)
      CALL SYMBOL(-.28,0.,.14,2HP9,0.,2)
      CALL PLOT( 0.,0.,3)
      DO 47 I= 1,JS
      CALL PLOT(X(I),TCPCAL*TCP(9,I),2)
   47 CONTINUE
      CALL PLOT(.5,-8.00,-3)
      CALL SYMBOL(-2.0,-1.50,.07,3,0.,-1)
      CALL SYMBOL(6.5,-1.50,.07,3,0.,-1)
      CALL PLOT(-2.5,-2.0,-3)
      CALL FRAME(1)
      RETURN
  101 CALL PLOT(0.,0.,999)
      RETURN
      END



      SUBROUTINE GRID
C     PLOTS THE MESH SYSTEM
C     RENAME COMMON/A/
C     THE POSITION OF GM AND GN ARE OVERLAPPED BY XMESH AND YMESH.
C     THE ROUTINE SHOULD BE CALLED AT THE RIGHT END OF THE PROGRAM
      COMMON/A/ XMESH(132,36),G(132,36),YMESH(132,36)
     *          ,S0(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/D/ SLOPT,TRAIL,SCAL
      COMMON/F/ XR,YR,KS,XS(500),YS(500)
      COMMON/G/ TITLE(20),IPLOT
      COMMON/K/ I0,I1,I2,I3,J1,J2,J3
      IF (IPLOT) 1,11,101
    1 CALL PLOTSBL(5000,23HI-CHUNG CHANG    109104W)
   11 CALL PLOT(2.5,2.0,-3)
      CALL SYMBOL(-2.0,-1.50,.07,3,0.,-1)
      CALL SYMBOL(6.5,-1.50,.07,3,0.,-1)
      ENCODE(80,12,R) TITLE
   12 FORMAT(20A4)
      CALL SYMBOL(0.,-.5,.14,R,0.,80)
      ENCODE(35,14,R) NX,NY
   14 FORMAT(24HNEAR FIELD GRID SYSTEM  ,I4,3H X ,I4)
      CALL SYMBOL(0.,-.75,.14,R,0.,35)
      CALL PLOT(1.75,4.5,-3)
C     MESH
      XO= XR/SCAL
      YO= YR/SCAL
      DO 13 I= IO,I3
      DO 13 J= J1,J3
      XMESH(I,J)= XO  +.5*(AO(I)**2  -(BO(J)  +SO(I))**2)
   13 YMESH(I,J)= YO  +AO(I)*(BO(J)  +SO(I))
C     DRAWS THE GRID CURVES AROUND AIRFOIL
      XMAX= XMESH(IX1,3)
      XMIN= XMAX
      DO 22 I= IX1,IX2
      XMAX= AMAX1(XMESH(I,3),XMAX)
   22 XMIN= AMIN1(XMESH(I,3),XMIN)
      SCALE= 1./(XMAX-XMIN)
      DO 32 J= J1,J3
      KP         = 3
      DO 32 I= IO,I3
      XP= SCALE*(XMESH(I,J)-XMIN)
      YP= SCALE*(YMESH(I,J)- YMESH(IO,3))
      IF(XP.LT.-3.75.OR.XP.GT.4.75.OR.YP.LT.-4.5.OR.YP.GT.4.5) GO TO 33
      CALL PLOT(XP,YP,KP)
      KP         = 2
      GO TO 32
   33 KP         = 3
   32 CONTINUE
C     DRAWS THE GRID CURVES RADIATING FROM AIRFOIL
      DO 42 I= IO,I3
      KP         = 3
      DO 42 J= J1,J3
      XP= SCALE*(XMESH(I,J)-XMIN)
      YP= SCALE*(YMESH(I,J)-YMESH(IO,3))
      IF(XP.LT.-3.75.OR.XP.GT.4.75.OR.YP.LT.-4.5.OR.YP.GT.4.5) GO TO 43
      CALL PLOT(XP,YP,KP)
      KP         = 2
      GO TO 42
   43 KP         = 3
   42 CONTINUE
      CALL PLOT(-1.75,-4.5,-3)
      CALL SYMBOL(-2.0,-1.50,.07,3,0.,-1)
      CALL SYMBOL(6.5,-1.50,.07,3,0.,-1)
      CALL PLOT(-2.5,-2.00,-3)
      CALL FRAME(1)
      RETURN
  101 CALL PLOT(0.,0.,999)
```

```
      RETURN
      END


      SUBROUTINE USTADI
C     UNSTEADY TRANSONIC POTENTIAL FLOW EQUATION IN QUASILINEAR FORM
C     WITH FIRST ORDER RADIATION BOUNDARY CONDITIONS IN MOVING
C     SHEARED PARABOLIC COORDINATES ARE SOVLED BY AN ALTERNATING
C     DIRECTION IMPLICIT SCHEME WITH Y-SWEEP FIRST
      COMMON/A/ GM(132,36),G(132,36),GN(132,36),SO(132),S1(132),S2(132)
     1          ,A0(132),A1(132),A2(132),A3(132),B0(36),B1(36),B2(36)
     2          ,B3(36),NX,NY,IX1,IX2,KSYM,FMACH,ALPHA,CA,SA,FMACH2
     3          ,AL,UTIM,CB,SB,NS,RG,IG,JG
      COMMON/H/ DX,DY,DT,DXX,DYY,DTT,DXY,DXT,DYT,TSR
      COMMON/J/ RAD,PI,ALS,ALT,ALTT,AMPLA,FREQRA,FASAGA,FMACHS,FMACHT
     1 ,AMPLM,FREQRM,FASAGM,CETAS,CETAT,CETATT,AMPLC,FREQRC,FASAGC,CETA
     2 ,FREQR,IPSURE
      COMMON/K/ IO,I1,I2,I3,J1,J2,J3
      COMMON/WAKE/ NIT,WG(132)
      DIMENSION C(132),E(132),F(132)
      COMPLEX  WA
      DDT= 1./DT
      DDXX= 1./DXX
      DDYY= 1./DYY
      NS          = 0
      AAO = 1.
      RG=0.
      IG          = 0
      JG          = 0
C     *****
C     Y-SWEEP
C     *****
      IM= IO -1
      IMM= IO - 2
      C(IM)= 0.
      C(IMM)= 0.
      E(IM)=0.
      E(IMM)=0.
      F(IM)=0.
      F(IMM)=0.
C     UPPER BOUNDARY
      J=J3
      I= IO
      ANG=PI*.75
      Y=SO(I)+BO(J)
      X= AO(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      H=SQRT(HH)
      DH= 1./H
      GI= (G(I+1,J)-G(I,J))*2.
      GJ= (G(I,J)-G(I,J-1))*2.
      GX         = A1(I)* GI  -S1(I)*B1(J)* GJ
      GY         =  B1(J)* GJ
      U= GX*DH
      V= GY*DH
      QQ=U*U+V*V
      CHAIN= XT*GX + YT*GY
      FIT= GM(I,J) *DDT + CHAIN
      AA=AA0-.2*QQ-.4*FIT
      AA=AMAX1(AA,.0001)
      A=SQRT(AA)
      WA=CMPLX(COS(ANG),SIN(ANG))*CMPLX(A,0.)
      U= U+REAL(WA)  +H*XT
      V= V + AIMAG(WA) + H*YT
      AV= V-U*S1(I)
      TGI= GI
      TGJ= GJ
      TGMI= 2.*(GM(I+1,J)-GM(I,J))
      TGMJ= 2.*(GM(I,J)-GM(I,J-1))
      YI= -GM(I,J) + DT*(U*TGMI*A1(I)+AV*TGMJ*B1(J))*DH
     1        -2.*DT*(U*TGI*A1(I)+AV*TGJ*B1(J))*DH
      BI= 0.
      DI= 0.
      EI= 0.
      CI= DT*DH*U*2.*A1(I)
      AI= 1.-CI
      GAMA= DI
      BEDA=BI-C(I-2)*GAMA
      ALFA= 1./(AI-BEDA*C(I-1)-GAMA*E(I-2))
      C(I)= (CI-BEDA*E(I-1))*ALFA
      E(I)= EI*ALFA
      F(I)=(YI-BEDA*F(I-1)-GAMA*F(I-2))*ALFA
      DO 1 I= I1,I2
      ANG= .5*PI
      Y=SO(I)+BO(J)
      X= AO(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      H=SQRT(HH)
      DH= 1./H
      GI= G(I+1,J)-G(I-1,J)
      GJ= 2.*(G(I,J)-G(I,J-1))
      TGJ= GJ
      TGMJ= 2.*(GM(I,J)-GM(I,J-1))
      GX         = A1(I)* GI  -S1(I)*B1(J)* GJ
      GY         =  B1(J)* GJ
      U= GX*DH
      V= GY*DH
      QQ=U*U+V*V
      CHAIN= XT*GX + YT*GY
      FIT= GM(I,J) *DDT + CHAIN
      AA=AAO-.2*QQ-.4*FIT
      AA=AMAX1(AA,.0001)
      A=SQRT(AA)
      WA=CMPLX(COS(ANG),SIN(ANG))*CMPLX(A,0.)
      U= U+REAL(WA)   +H*XT
      V= V + AIMAG(WA) + H*YT
      AV= V-U*S1(I)
      IF(U.LT.0.) GO TO 2
      TGI= 2.*(G(I,J)-G(I-1,J))
      TGMI= 2.*(GM(I,J)-GM(I-1,J))
      BI=     - 2.*A1(I)*DT*DH*U
      AI= 1.-BI
      CI= 0.
      GO TO 3
    2 TGI=2.*(G(I+1,J)-G(I,J))
      TGMI= 2.*(GM(I+1,J)-GM(I,J))
      CI=        2.*A1(I)*DT*DH*U
      AI= 1.-CI
      BI= 0.
    3 YI= -GM(I,J) + DT*(U*TGMI*A1(I)+AV*TGMJ*B1(J))*DH
     1          -2.*DT*(U*TGI*A1(I)+AV*TGJ*B1(J))*DH
      DI= 0.
      EI= 0.
      GAMA= DI
      BEDA=BI-C(I-2)*GAMA
      ALFA= 1./(AI-BEDA*C(I-1)-GAMA*E(I-2))
      C(I)= (CI-BEDA*E(I-1))*ALFA
      E(I)= EI*ALFA
      F(I)=(YI-BEDA*F(I-1)-GAMA*F(I-2))*ALFA
    1 CONTINUE
      I= I3
      ANG= .25*PI
      Y=SO(I)+BO(J)
      X= AO(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      H=SQRT(HH)
      DH= 1./H
      GI= (G(I,J)-G(I-1,J))*2.
      GJ= (G(I,J)-G(I,J-1))*2.
      GX          = A1(I)* GI  -S1(I)*B1(J)* GJ
      GY          =  B1(J)* GJ
      U= GX*DH
      V= GY*DH
      QQ= U*U+V*V
      CHAIN= XT*GX + YT*GY
      FIT= GM(I,J) *DDT + CHAIN
      AA=AAO-.2*QQ-.4*FIT
      AA= AMAX1(AA,.001)
      A= SQRT(AA)
      WA=CMPLX(COS(ANG),SIN(ANG))*CMPLX(A,0.)
      U= U+REAL(WA)  +H*XT
      V= V + AIMAG(WA) + H*YT
      AV= V-U*S1(I)
      TGI= GI
      TGJ= GJ
      TGMI= 2.*(GM(I,J)-GM(I-1,J))
      TGMJ= 2.*(GM(I,J)-GM(I,J-1))
      YI= -GM(I,J) + DT*(U*TGMI*A1(I)+AV*TGMJ*B1(J))*DH
     1        -2.*DT*(U*TGI*A1(I)+AV*TGJ*B1(J))*DH
      CI= 0.
      DI= 0.
      EI= 0.
      BI=-DT*DH*U*2.*A1(I)
      AI= 1.-BI
      GAMA= DI
      BEDA=BI-C(I-2)*GAMA
      ALFA= 1./(AI-BEDA*C(I-1)-GAMA*E(I-2))
      C(I)= (CI-BEDA*E(I-1))*ALFA
      E(I)= EI*ALFA
      F(I)=(YI-BEDA*F(I-1)-GAMA*F(I-2))*ALFA
      CG= 0.
      CCG= 0.
      DO 4 K= IO,I3
      I= I3+IO-K
      DG= CG
      CG=        F(I)-C(I)*CG-E(I)*CCG
      CCG= DG
    4 GN(I,J)= CG
    5 J= J-1
C     LEFT BOUNDARY
      I= IO
      ANG=PI
      Y=SO(I)+BO(J)
      X= AO(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      H=SQRT(HH)
      DH= 1./H
      GI= (G(I+1,J)-G(I,J))*2.
      GJ=G(I,J+1)-G(I,J-1)
      GX          = A1(I)* GI   -S1(I)*B1(J)* GJ
      GY          =  B1(J)* GJ
      U= GX*DH
      V= GY*DH
      QQ=U*U+V*V
      CHAIN= XT*GX + YT*GY
      FIT= GM(I,J) *DDT + CHAIN
      AA=AAO-.2*QQ-.4*FIT
      AA=AMAX1(AA,.0001)
      A=SQRT(AA)
      WA=CMPLX(COS(ANG),SIN(ANG))*CMPLX(A,0.)
      U= U+REAL(WA)   +H*XT
      V= V + AIMAG(WA) + H*YT
      AV          = V  -U*S1(I)
      TGI= GI
      TGMI= 2.*(GM(I+1,J)-GM(I,J))
      IF( AV.LT.O.) GO TO 6
      TGJ= 2.*(G(I,J)-G(I,J-1))
      TGMJ= 2.*(GM(I,J)-GM(I,J-1))
      GO TO 7
    6 TGJ=2.*(G(I,J+1)-G(I,J))
      TGMJ= 2.*(GM(I,J+1)-GM(I,J))
    7 YI= -GM(I,J) + DT*(U*TGMI*A1(I)+AV*TGMJ*B1(J))*DH
     1        -2.*DT*(U*TGI*A1(I)+AV*TGJ*B1(J))*DH
      BI= 0.
      DI= 0.
      EI= 0.
      CI= DT*DH*U*2.*A1(I)
      AI= 1.-CI
      GAMA= DI
      BEDA=BI-C(I-2)*GAMA
      ALFA= 1./(AI-BEDA*C(I-1)-GAMA*E(I-2))
      C(I)= (CI-BEDA*E(I-1))*ALFA
      E(I)= EI*ALFA
      F(I)=(YI-BEDA*F(I-1)-GAMA*F(I-2))*ALFA
C     INTERIOR
      DO 8 I= I1,I2
      FX=1.+S1(I)**2
      Y=SO(I)+BO(J)
      X= AO(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      CX= Y*Y-X*X
      CY= 2.*X*Y
      XTX= (CA*CX-CY*SA)*DHH*DHH
      XTY= -.5*(ALT+CETAT)-DHH*DHH*(SA*CX+CA*CY)
      YTX= -XTY
      YTY= XTX
      CX= X**3-3.*X*Y*Y
      CY= 3.*X*X*Y-Y**3
      BX= 2.*SA*CA
      BY= CA*CA-SA*SA
      XTT=-.5*Y*(ALTT+CETATT)-.25*X*(ALT+CETAT)**2
     1  +FMACHT*DHH*(X*CB+Y*SB)-ALT*DHH*(X*SA-Y*CA)-DHH**3*(CX*BY+BX*CY)
      YTT= .5*X*(ALTT+CETATT)-.25*Y*(ALT+CETAT)**2
     1  +FMACHT*DHH*(X*SB-Y*CB)+ALT*DHH*(X*CA+Y*SA)+DHH**3*(BY*CY-BX*CX)
      H=SQRT(HH)
      DH= 1./H
      GI=G(I+1,J)-G(I-1,J)
      GMI= GM(I+1,J)-GM(I-1,J)
      GJ=G(I,J+1)-G(I,J-1)
      GMJ= GM(I,J+1)-GM(I,J-1)
      GX          = A1(I)* GI  -S1(I)*B1(J)* GJ
      GY          =  B1(J)* GJ
      U= GX*DH
      V= GY*DH
      AU= U+V*S1(I)
      AV          = V  -U*S1(I)
      UR= XT*H+ U
      VR= YT*H + V
      AVR= VR-UR*S1(I)
      UUR= UR*UR
      VVR= VR*VR
      QQR= UUR + VVR
      S= 1.
      IF( UR.LT.O.) S= -1
      T=1.
      IF(AVR.LT.O.) T= -1.
      UU=U*U
      UV=U*V
      VV=V*V
      QQ=UU+VV
      CHAIN= XT*GX + YT*GY
      FIT= GM(I,J) *DDT + CHAIN
      AA=AAO-.2*QQ-.4*FIT
      AA=AMAX1(AA,.0001)
      AB=A1(I)*B1(J)
      GII=(G(I+1,J)-2.*G(I,J)+G(I-1,J))*DDXX + A3(I)*GI
      GIJ=G(I+1,J+1)-G(I+1,J-1)-G(I-1,J+1)+G(I-1,J-1)
      GJJ=(G(I,J+1)-2.*G(I,J)+G(I,J-1))*DDYY + B3(J)*GJ
      GMII=(GM(I+1,J)-2.*GM(I,J)+GM(I-1,J))*DDXX + A3(I)*GMI
      GMIJ=GM(I+1,J+1)-GM(I+1,J-1)-GM(I-1,J+1)+GM(I-1,J-1)
      GMJJ=(GM(I,J+1)-2.*GM(I,J)+GM(I,J-1))*DDYY + B3(J)*GMJ
C     ROTATED COORDINATES TERMS
      CX= XTT+ 2.*(U*XTX+ V*XTY)*DH
      CY= YTT + 2.*(U*YTX + V*YTY)*DH
      AX= A1(I)*CX
      AY= B1(J)*(CY-CX*S1(I))
      WR= AX*GI + AY*GJ
      R=  QQ *(U*X+V*Y)*DH -(AA-UUR)*S2(I)*GY -HH*WR
      IF(QQR.GE.AA) GO TO 9
C     CENTRAL DIFFERENCING
      AXX= (AA-UUR) *A2(I)
      AXY= -2.*AB*( AA*S1(I) + UR*AVR)
      AYY= B2(J) *(AA*FX-AVR*AVR)
      YR= R + AXX*GII + AXY*GIJ + AYY*GJJ
      AXY= -2.*UR*AVR*AB
      YMR=AXX*GMII + AXY*GMIJ + AYY*GMJJ
      BB= .5*DTT*DHH*A2(I)*(UUR-AA)
      DI= 0.
      BI= BB*(DDXX-A3(I))
      AI= 1.-2.*BB*DDXX
      CI= BB*(DDXX+A3(I))
      EI= 0.
      GO TO 10
C     TYPE DEPENDENT DIFFERENCNG
    9 NS          = NS  +1
      K           = S
      IM          = I  -K
      IMM        = IM   -K
      L=T
      JM=J-L
      JMM=JM-L
      AUR= UR+ VR*S1(I)
      AQ= AA/QQR
      BXX= VVR*A2(I)
      BXY= -2.*AB*VR*AUR
      BYY= AUR*AUR*B2(J)
      GNN=BXX*GII+BXY*GIJ+BYY*GJJ
      GMNN= BXX*GMII+BYY*GMJJ
      GIJM=G(I,J)-G(IM,J)-G(I,JM)+G(IM,JM)
      GMIJM= GM(I,J)-GM(IM,J)-GM(I,JM)+GM(IM,JM)
      IF( JMM.GT.J3) GO TO 11
      GJJM=(G(I,J)-2.*G(I,JM)+G(I,JMM))*DDYY + B3(J)*GJ
      GMJJM=( GM(I,J)-2.*GM(I,JM)+GM(I,JMM))*DDYY + B3(J)*GMJ
      GO TO 12
   11 GJJM=GJJ
      GMJJM= GMJJ
   12 IF( IMM.LT.IC.OR.IMM.GT.I3) GO TO 13
      GIIM=(G(I,J)-2.*G(IM,J)+G(IMM,J))*DDXX + A3(I)*GI
      GMIIM= (GM(I,J)-2.*GM(IM,J)+GM(IMM,J))*DDXX + A3(I)*GMI
      GO TO 14
   13 GIIM= GII
      GMIIM= GMII
   14 AXX= UUR*A2(I)
      AXY= 8.*S*T*UR*AVR*AB
      AYY= AVR*AVR*B2(J)
      GSS=AXX*GIIM+AXY*GIJM+AYY*GJJM
      GMSS= AXX*GMIIM+AYY*GMJJM
      YR         = (AQ  -1.)*GSS  +AQ*GNN  +R
      YMR=AQ*(GMSS+GMNN)
      GMSS       = AXX*GMIIM+AXY*GMIJM+AYY*GMJJM
      YMR= YMR -GMSS
      BB= .5*DTT*DHH*UUR*(1.-AQ)*A2(I)
      CC= -.5*DTT*DHH*AQ*VVR*A2(I)
      BBCC= BB+CC
      IF( UR.LT.O.) GO TO 15
      IF( I.EQ.I1) GO TO 16
      DI= BB*DDXX
      BI= DDXX*(CC-2.*BB) -A3(I)*BBCC
      AI= 1. +DDXX*(BB-2.*CC)
      CI= CC*DDXX + A3(I)*BBCC
      EI= O.
      GO TO 10
   15 IF( I.EQ.I2) GO TO 16
      DI= O.
      BI= CC*DDXX-A3(I)*BBCC
      AI= 1.+DDXX*(BB-2.*CC)
      CI= DDXX*(CC-2.*BB) + A3(I)*BBCC
      EI= BB*DDXX
      GO TO 10
   16 DI= O.
      BI= BBCC*(DDXX-A3(I))
      AI= 1. -2.*DDXX*BBCC
      CI= BBCC*(DDXX+A3(I))
      EI= 0.
C     ADVECTION TERMS
C     UPWIND DIFFERENCING
   10 YY=AVR*2.*B1(J)*(GM(I,J)-GM(I,J-1))
      IF(AVR.LT.0.) YY=AVR*B1(J)*(GM(I,J+1)-GM(I,J))*2.
      BB=UR*DT*DH*2.*A1(I)
      IF(UR.LT.0.) GO TO 18
      YY= YY+UR*2.*A1(I)*(GM(I,J)-GM(I-1,J))
      BI= BI-BB
      AI= AI+BB
      GO TO 19
   18 YY= YY+UR*2.*A1(I)*(GM(I+1,J)-GM(I,J))
      CI= CI+BB
      AI= AI-BB
   19 YI= GM(I,J)+DTT*(YR-.5*YMR)*DHH -DT*YY*DH
      GAMA= DI
      BEDA=BI-C(I-2)*GAMA
      ALFA= 1./(AI-BEDA*C(I-1)-GAMA*E(I-2))
      C(I)= (CI-BEDA*E(I-1))*ALFA
      E(I)= EI*ALFA
      F(I)=(YI-BEDA*F(I-1)-GAMA*F(I-2))*ALFA
    8 CONTINUE
C     RIGHT BOUNDARY
      I= I3
      ANG= 0.
      Y=SO(I)+BO(J)
      X= AO(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      H=SQRT(HH)
      DH= 1./H
      GI= (G(I,J)-G(I-1,J))*2.
      GJ=G(I,J+1)-G(I,J-1)
      GX          = A1(I)* GI   -S1(I)*B1(J)* GJ
      GY          =  B1(J)* GJ
      U= GX*DH
      V= GY*DH
      QQ= U*U+V*V
      CHAIN= XT*GX + YT*GY
      FIT= GM(I,J) *DDT + CHAIN
      AA=AA0-.2*QQ-.4*FIT
      AA=AMAX1(AA,.0001)
      A=SQRT(AA)
      WA=CMPLX(COS(ANG),SIN(ANG))*CMPLX(A,0.)
      U= U+REAL(WA)  +H*XT
      V= V + AIMAG(WA) + H*YT
      AV          = V  -U*S1(I)
      TGI= GI
      TGMI= 2.*(GM(I,J)-GM(I-1,J))
      IF( AV.LT.0.) GO TO 20
      TGJ= 2.*(G(I,J)-G(I,J-1))
      TGMJ= 2.*(GM(I,J)-GM(I,J-1))
      GO TO 21
   20 TGJ=2.*(G(I,J+1)-G(I,J))
      TGMJ= 2.*(GM(I,J+1)-GM(I,J))
   21 YI= -GM(I,J) + DT*(U*TGMI*A1(I)+AV*TGMJ*B1(J))*DH
     1        -2.*DT*(U*TGI*A1(I)+AV*TGJ*B1(J))*DH
      DI= 0.
      EI= 0.
      CI= 0.
      BI= -DT*DH*U*2.*A1(I)
      AI= 1.-BI
      GAMA= DI
      BEDA=BI-C(I-2)*GAMA
      ALFA= 1./(AI-BEDA*C(I-1)-GAMA*E(I-2))
      C(I)= (CI-BEDA*E(I-1))*ALFA
      E(I)= EI*ALFA
      F(I)=(YI-BEDA*F(I-1)-GAMA*F(I-2))*ALFA
      CG= 0.
      CCG= 0.
      DO 22 K= I0,I3
      I= I3+I0-K
      DG= CG
      CG=       F(I)-C(I)*CG-E(I)*CCG
      CCG= DG
   22 GN(I,J)= CG
      IF( J.GT.3) GO TO 5
C     *****
C     X-SWEEP
C     *****
      C(1)= 0.
      C(2)= 0.
      E(1)= 0.
      E(2)= 0.
      F(1)= 0.
      F(2)= 0.
C     LEFT BOUNDARY
      I= I0
      DO 23 J= J1,J3
      IF( J.EQ.J1) GO TO 24
      IF( J.EQ.J3) GO TO 25
      ANG= PI
      GJ= G(I,J+1)-G(I,J-1)
      GO TO 26
   24 ANG= 1.25*PI
      GJ= G(I,J+1)-G(I,J-1)
      GO TO 26
   25 ANG= .75*PI
      GJ= 2.*(G(I,J)-G(I,J-1))
   26 Y=S0(I)+B0(J)
      X= A0(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      HH=AO(I)*AO(I)+Y*Y
      H=SQRT(HH)
      DH= 1./H
      GI= (G(I+1,J)-G(I,J))*2.
      GX          = A1(I)* GI   -S1(I)*B1(J)* GJ
      GY          =  B1(J)* GJ
      U= GX*DH
      V= GY*DH
      QQ=U*U+V*V
      CHAIN= XT*GX + YT*GY
      FIT= GM(I,J) *DDT + CHAIN
      AA=AAO-.2*QQ-.4*FIT
      AA=AMAX1(AA,.0001)
      A=SQRT(AA)
      WA=CMPLX(COS(ANG),SIN(ANG))*CMPLX(A,0.)
      U= U+REAL(WA)  +H*XT
      V= V + AIMAG(WA) + H*YT
      AV= V-U*S1(I)
      IF( J.EQ.J1) GO TO 27
      IF( J.EQ.J3) GO TO 28
      IF( AV.LT.0.) GO TO 27
   28 BI=  -DT*DH*AV*2.*B1(J)
      AI= 1.-BI
      CI= 0.
      GO TO 29
   27 CI= DT*DH*AV*2.*B1(J)
      AI= 1.-CI
      BI= 0.
   29 YI= GN(I,J)
      DI= 0.
      EI= 0.
      GAMA= DI
      BEDA=BI-C(J-2)*GAMA
      ALFA= 1./(AI-BEDA*C(J-1)-GAMA*E(J-2))
      C(J)= (CI-BEDA*E(J-1))*ALFA
      E(J)= EI*ALFA
      F(J)=(YI-BEDA*F(J-1)-GAMA*F(J-2))*ALFA
   23 CONTINUE
      CCG= 0.
      CG= 0.
      DO 30 K= J1,J3
      J= J3 + J1 -K
      DG= CG
      CG= F(J)-C(J)*CG-E(J)*CCG
      CCG= DG
   30 GN(I,J)= CG
C     INTERIOR
      DO 31 I= I1,I2
      FX= 1. + S1(I)**2
      DO 32 J= J1,J3
      IF( J.EQ.J3) GO TO 33
      Y=SO(I)+BO(J)
      X= AO(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      H=SQRT(HH)
      DH= 1./H
      GI=G(I+1,J)-G(I-1,J)
      GJ=G(I,J+1)-G(I,J-1)
      GX          = A1(I)* GI   -S1(I)*B1(J)* GJ
      GY          =   B1(J)* GJ
      U= GX*DH
      V= GY*DH
      AU= U+V*S1(I)
      AV          = V   -U*S1(I)
      UR= XT*H+ U
      VR= YT*H + V
      AVR= VR-UR*S1(I)
      UUR= UR*UR
      VVR= VR*VR
      QQR= UUR + VVR
      QQ= U*U + V*V
      CHAIN= XT*GX + YT*GY
      FIT= GM(I,J) *DDT + CHAIN
      AA=AA0-.2*QQ-.4*FIT
      AA=AMAX1(AA,.0001)
      IF(QQR.GE.AA) GO TO 34
      BB= .5*DTT*DHH*(AVR*AVR-AA*FX)*B2(J)
      DI= 0.
      BI= BB*(DDYY-B3(J))
      AI= 1.-2.*BB*DDYY
      CI= BB*(DDYY + B3(J))
      EI= 0.
      GO TO 36
   34 AQ= AA/QQR
      AUR= UR+VR*S1(I)
      BB= .5*DTT*DHH*AVR*AVR*(1.-AQ)*B2(J)
      CC= -.5*DTT*DHH*AQ*AUR*AUR*B2(J)
      BBCC= BB+ CC
      IF( J.EQ.4) GO TO 38
      IF( J.EQ.J2) GO TO 39
      IF(AVR.LT.0.) GO TO 40
      DI= BB*DDYY
      BI= DDYY*(CC-2.*BB) -B3(J)*BBCC
      AI= 1. + DDYY*(BB-2.*CC)
      CI= CC*DDYY + B3(J)*BBCC
      EI= 0.
      GO TO 36
   40 DI= 0.
      BI= DDYY*CC -B3(J)*BBCC
      AI= 1. + DDYY*(BB-2.*CC)
      CI= DDYY*(CC-2.*BB)+B3(J)*BBCC
      EI= BB*DDYY
      GO TO 36
   38 DI= 0.
```

```
      IF(AVR.LT.0.) GO TO 41
      EI= 0.
      BI= BBCC*(DDYY-B3(J))
      AI= 1. -2.*BBCC*DDYY
      CI= BBCC*(DDYY + B3(J))
      GO TO 36
   41 BI= CC*DDYY -B3(J)*BBCC
      AI= 1. + DDYY*(BB-2.*CC)
      CI= DDYY*(CC-2.*BB) + B3(J)*BBCC
      EI= BB*DDYY
      GO TO 36
   39 EI= 0.
      IF(AVR.LT.0.) GO TO 42
      DI= BB*DDYY
      BI= DDYY*(CC-2.*BB) -B3(J)*BBCC
      AI= 1.+DDYY*(BB-2.*CC)
      CI= DDYY*CC + B3(J)*BBCC
      GO TO 36
   42 DI= 0.
      BI= BBCC*(DDYY-B3(J))
      AI= 1.-2.*BBCC*DDYY
      CI= BBCC*(DDYY+B3(J))
C     ADVECTION TERMS
   36 BB= DT*DH*AVR*2.*B1(J)
      IF(AVR.LT.0.) GO TO 43
      BI= BI-BB
      AI= AI+ BB
      GO TO 46
   43 CI= CI +BB
      AI= AI-BB
      GO TO 46
   33 ANG= .5*PI
      Y=S0(I)+B0(J)
      X= A0(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      H=SQRT(HH)
      DH= 1./H
      GI= G(I+1,J)-G(I-1,J)
      GJ= 2.*(G(I,J)-G(I,J-1))
      GX          = A1(I)* GI  -S1(I)*B1(J)* GJ
      GY          =  B1(J)* GJ
      U= GX*DH
      V= GY*DH
      QQ=U*U+V*V
      CHAIN= XT*GX + YT*GY
      FIT= GM(I,J) *DDT + CHAIN
      AA=AA0-.2*QQ-.4*FIT
      AA=AMAX1(AA,.0001)
      A=SQRT(AA)
      WA=CMPLX(COS(ANG),SIN(ANG))*CMPLX(A,0.)
      U= U+REAL(WA)   +H*XT
      V= V + AIMAG(WA) + H*YT
      AV= V-U*S1(I)
   48 BI=  -DT*DH*AV*B1(J)*2.
      AI= 1.-BI
      CI= 0.
      DI= 0.
      EI= 0.
   46 YI= GN(I,J)
      IF( J.EQ.3) YI= YI-BI*GM(I,2) -DI*GM(I,1)
      IF( J.EQ.4) YI= YI-DI*GM(I,2)
      GAMA= DI
      BEDA=BI-C(J-2)*GAMA
      ALFA= 1./(AI-BEDA*C(J-1)-GAMA*E(J-2))
      C(J)= (CI-BEDA*E(J-1))*ALFA
      E(J)= EI*ALFA
      F(J)= (YI-BEDA*F(J-1)-GAMA*F(J-2))*ALFA
   32 CONTINUE
      CCG= 0.
      CG= 0.
      DO 49 K= J1,J3
      J= J3+J1-K
      DG= CG
      CG= F(J)-C(J)*CG-E(J)*CCG
      CCG= DG
   49 GN(I,J)= CG
   31 CONTINUE
C     RIGHT BOUNDARY
      I= I3
      DO 50 J= J1,J3
      IF( J.EQ.J1) GO TO 51
      IF( J.EQ.J3) GO TO 52
      ANG= 0.
      GJ= G(I,J+1)-G(I,J-1)
      GO TO 53
   51 ANG= -.25*PI
      GJ= G(I,J+1)-G(I,J-1)
      GO TO 53
   52 ANG= .25*PI
      GJ= 2.*(G(I,J)-G(I,J-1))
   53 Y=SO(I)+BO(J)
      Y=SO(I)+BO(J)
      X= AO(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      H=SQRT(HH)
      DH= 1./H
      GI= 2.*(G(I,J)-G(I-1,J))
      GX        = A1(I)* GI  -S1(I)*B1(J)* GJ
      GY        =  B1(J)* GJ
      U= GX*DH
      V= GY*DH
      QQ=U*U+V*V
      CHAIN= XT*GX + YT*GY
      FIT= GM(I,J) *DDT + CHAIN
      AA=AA0-.2*QQ-.4*FIT
      AA=AMAX1(AA,.0001)
      A=SQRT(AA)
      WA=CMPLX(COS(ANG),SIN(ANG))*CMPLX(A,0.)
      U= U+REAL(WA)   +H*XT
      V= V + AIMAG(WA) + H*YT
      AV= V-U*S1(I)
      IF( J.EQ.J1) GO TO 54
      IF( J.EQ.J3) GO TO 55
      IF( AV.LT.0.) GO TO 54
   55 BI=  -DT*DH*AV*2.*B1(J)
      AI= 1.-BI
      CI= 0.
      GO TO 56
   54 CI= DT*DH*AV*2.*B1(J)
      AI= 1.-CI
      BI= 0.
   56 YI= GN(I,J)
      DI= 0.
      EI= 0.
      GAMA= DI
      BEDA=BI-C(J-2)*GAMA
      ALFA= 1./(AI-BEDA*C(J-1)-GAMA*E(J-2))
      C(J)= (CI-BEDA*E(J-1))*ALFA
      E(J)= EI*ALFA
      F(J)=(YI-BEDA*F(J-1)-GAMA*F(J-2))*ALFA
   50 CONTINUE
      CCG= 0.
      CG= 0.
      DO 57 K= J1,J3
      J= J3 + J1 -K
      DG= CG
      CG= F(J)-C(J)*CG-E(J)*CCG
      CCG= DG
   57 GN(I,J)= CG
C     UPDATE NEXT RUN DATA
      DO 58 I= I0,I3
      DO 58 J= J1,J3
      CG= GN(I,J)
      G(I,J)= G(I,J)+CG
      GM(I,J)= CG
      IF( ABS(CG).LE.RG) GO TO 58
      RG= ABS(CG)
      IG= I
      JG= J
   58 CONTINUE
      IF(NIT.NE.0) GO TO 59
      UTIM=UTIM + DT
      FASAGA= FREQRA*UTIM
      ALPHA= ALS + AMPLA*SIN(FASAGA)/RAD
      AL= ALPHA*RAD
      ALT= AMPLA*FREQRA*COS(FASAGA)/RAD
      ALTT= -AMPLA*FREQRA**2*SIN(FASAGA)/RAD
      FASAGM= FREQRM*UTIM
      FMACH= FMACHS + AMPLM*SIN(FASAGM)
      FMACHT= AMPLM*FREQRM*COS(FASAGM)
      FASAGC= FREQRC*UTIM
      CETARD=CETAS + AMPLC*SIN(FASAGC)/RAD
      CETA= CETARD *RAD
      CETAT= AMPLC*FREQRC*COS(FASAGC)/RAD
      CETATT= -AMPLC*FREQRC**2*SIN(FASAGC)/RAD
      CB= COS(ALPHA)
      SB= SIN(ALPHA)
      CA= FMACH*CB
      SA= FMACH*SB
      FMACH2= FMACH**2
C     WAKE CONDITION
   59 IF( NIT.EQ.1) GO TO 60
      GO TO 61
   60 DO 62 I= IX2,I3
   62 WG(I)= G(IX2,3)-G(IX1,3)
   61 I= IX2
      WG(I)= WG(I) + GN(IX2,3)-GN(IX1,3)
   63 I= I+1
      Y=SO(I)+BO(3)
      X= AO(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      H=SQRT(HH)
      YP= Y
      HP= H
      GI= G(I+1,3)-G(I-1,3)
      IF( I.EQ.I3) GI= 2.*(G(I,3)-G(I-1,3))
      GJ= 2.*(G(I,4)-G(I,3))
      UP          =(A1(I)* GI  -S1(I)*B1(3)* GJ )/H
      M= NX+ 4 -I
      Y=SO(M)+BO(3)
      HH=AO(M)*AO(M)+Y*Y
      H=SQRT(HH)
      HM= H
      YM= Y
      GI= G(M+1,3)-G(M-1,3)
      IF( I.EQ.I3) GI= 2.*(G(M+1,3)-G(M,3))
      GJ=2.*(G(M,4)-G(M,3))
      UM          =(A1(M)* GI  -S1(M)*B1(3)* GJ)/H
      Y= .5*(YP-YM)
      U= .5*(UP-UM)
      H= .5*(HP+HM)
      BF= 2.*DT*A1(I)*(U/H +XT)
      WG(I)= (WG(I)+BF*WG(I-1))/(1.+BF)
      IF( I.LT.I3) GO TO 63
      DO 67 I= I0,I3
      CCG= G(I,1)
      CG= G(I,2)
      M= NX+ 4 -I
      IF (I.GT.IX2) GO TO 65
      IF (I.LT.IX1) GO TO 66
      TANGENTIAL BOUNDARY CONDITION
      Y=SO(I)+BO(3)
      X= AO(I)
      HH= X*X + Y*Y
      DHH= 1./HH
      XT= -.5*Y*(ALT+CETAT) + DHH*(CA*X + SA*Y)
      YT= .5*X*(ALT+CETAT) -DHH*(CA*Y-SA*X)
      VBN=HH*(XT*S1(I) -YT)
      GI= G(I+1,3)-G(I-1,3)
      FX=1.+S1(I)**2
      BIS= FX*B1(3)
      GXSXVB= A1(I)*GI*S1(I)+VBN
      G(I,2) = G(I,4) -GXSXVB/BIS
      G(I,1)= G(I,5) -2.*GXSXVB/BIS
      GO TO 64
   65 G(I,2)= G(M,4)+WG(I)
      G(I,1)= G(M,5)+WG(I)
      GO TO 64
   66 G(I,2)= G(M,4)-WG(M)
      G(I,1)= G(M,5)-WG(M)
   64 GM(I,2)= G(I,2)-CG
      GM(I,1)= G(I,1)-CCG
   67 CONTINUE
      RETURN
      END
```