Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

1990-09

Aerodynamic coefficients of a symmetrical airfoil in an oscillating flow

Chamberlain, George A.

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/10801

NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD-A241 073

OCT 0 3 1991
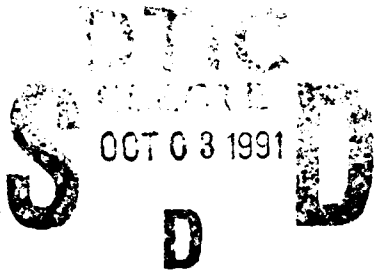
THESIS

AERODYNAMIC COEFFICIENTS OF A
SYMMETRICAL AIRFOIL IN AN
OSCILLATING FLOW

by

LT George A. Chamberlain, Jr.

September, 1990

Thesis Advisor: James A. Miller

Approved for public release; distribution is unlimited.

91-12191

REPORT DOCUMENTATION PAGE		Form Approved OMB No 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION AVAILABILITY OF REPORT
2b DECLASSIFICATION/DOWNGRADING SCHEDULE	Approved for public release; distribution is unlimited.
4 PERFORMING ORGANIZATION REPORT NUMBER(S) Naval Postgraduate School	5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION Naval Postgraduate School	6b OFFICE SYMBOL (If applicable) Code 31	7a NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		7b ADDRESS (City, State and ZIP Code) Monterey, CA 93943-5000
8a NAME OF FUNDING SPONSORING ORGANIZATION	8b OFFICE SYMBOL (If applicable)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code)		10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO

11 TITLE (Include Security Classification)

AERODYNAMIC COEFFICIENTS OF A SYMMETRICAL AIRFOIL IN AN OSCILLATING FLOW (UNCLASSI

12 PERSONAL AUTHOR(S)
Chamberlain, George A., Jr.

13a TYPE OF REPORT Master;s Thesis	13b TIME COVERED FROM ___ TO ___	14 DATE OF REPORT (Year, Month Day) September, 1990	15 PAGE COUNT 102

16 SUPPLEMENTARY NOTATION
The views in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U. S. Government

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Oscillating Flow, Aerodynamic coefficients, Symmetrical airfoil

19 ABSTRACT (Continue on reverse if necessary and identify by block number)
A symmetrical airfoil at fixed angles of attack from 0 to 35 degrees was placed in an air flow with controlled velocity oscillations at 10 Hz. The instantaneous pressure distribution and flow velocity were measured simultaneously. Pressure information was integrated numerically to determine instantaneous values for lift, drag and moment about the quarter chord. The results were plotted as three-dimensional surface plots in angle of attack, cycle time and performance coefficient axes, or with chordwise position, cycle time and pressure coefficient axes, as appropriate. The pressure distribution over to airfoil was shown to be primarily a function of freestream static pressure. Aerodynamic coefficients as a function of angle of attack and time exhibited seemingly random, erratic behavior, whereas the mean values of the same coefficients were similar in nature to those obtained in steady flow. The data indicate that the instantaneous performance of an airfoil in an oscillating flow is not merely a continuum of many quasi-steady states.

20 DISTRIBUTION AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL James A. Miller	22b TELEPHONE (Include Area Code) (408) 646-2897 22c OFFICE SYMBOL AA/MO

i

Aerodynamic Coefficients of a
Symmetrical Airfoil in an
Oscillating Flow

by

George A. Chamberlain, Jr.
Lieutenant, United States Navy
B.S., United States Naval Academy, 1982

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the
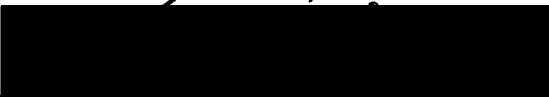
NAVAL POSTGRADUATE SCHOOL
September, 1990

Author: _____
George A. Chamberlain, Jr.

Approved by: _____
James A. Miller, Thesis Advisor

_____
Louis V. Schmidt, Second Reader

_____
E. Roberts Wood, Chairman
Department of Aeronautical Engineering

ABSTRACT

A symmetrical airfoil at fixed angle of attack from 0 to 35 degrees was placed in an air flow with controlled velocity oscillations at 10 Hz. The instantaneous pressure distribution and flow velocity were measured simultaneously. Pressure information was integrated numerically to determine instantaneous values for lift, drag and moment about the quarter chord. The results were plotted as three-dimensional surface plots in angle of attack, cycle time and performance coefficient axes, or with chordwise position, cycle time and pressure coefficient axes, as appropriate. The pressure distribution over the airfoil was shown to be primarily a function of freestream static pressure. Aerodynamic coefficients as a function of angle of attack and time exhibited seemingly random, erratic behavior, whereas the mean values of the same coefficients were similar in nature to those obtained in steady flow. The data indicate that the instantaneous performance of an airfoil in an oscillating flow is not merely a continuum of many quasi-steady states.

iii

TABLE OF CONTENTS

I. INTRODUCTION

The instantaneous pressure distribution on an airfoil in an oscillating flow is not well understood at present, chiefly because previous attempts to map it were hampered by the inability to measure pressure and velocity simultaneously at data rates compatible with the oscillating flow frequency, by the lack of computer resources for data reduction and by the lack of graphics software necessary to effectively display the results. The widespread use of microcomputers in aerodynamic research and the availability of three-dimensional plotting software now makes possible the acquisition, calculation and display of the pressure distribution and performance coefficients, C_L, C_D and $C_{Mc/4}$, in real-time format. This type of information is of particular interest to designers of helicopter rotor blades, as it reflects the varying performance of such a wing during its normal cyclic operation.

The basic approach taken in this investigation was to observe the pressure distribution over a stationary symmetrical airfoil at angles of attack ranging from 0 to 20 degrees, in increments of 5 degrees, in a steady flow, and at angles of attack ranging from 0 to 35 degrees, in increments of 5 degrees, in a controlled oscillating flow at 10 Hz. The pressure distribution was converted to coefficient of pressure, C_p, and integrated to obtain the normal force coefficient, C_N. This was further resolved into lift and drag components, C_L and C_D. Additionally, the C_P distributions were integrated to obtain the

1

moment coefficient about the quarter chord, $C_{M^{c/4}}$. This information was then plotted as a function of cycle time, t/T, and chordwise position, x/c, or as a function of angle of attack and cycle time, as appropriate, and a comparison was made of the resulting similarities and differences between steady flow and oscillating flow aerodynamic coefficients.

II. EXPERIMENTAL EQUIPMENT

A. WIND TUNNEL

1. General Description

The experiment was conducted in the Naval Postgraduate School Aeronautical Engineering low-speed oscillating flow wind tunnel. Figure 1, a plan view of the wind tunnel, shows that it is a continuous flow, in-draft facility.

The inlet is an eight foot square cross section, which contracts to a two foot square cross section in the test section. The inlet is equipped with three high solidity screens. These, and the 16:1 contraction ratio result in a very smooth, steady flow through the test section. Measurement of steady flow test section velocity profiles have shown that velocity variation is less than 1.0 percent of the mean to within 3 inches of any wall. Typical freestream turbulence intensity in steady flow is 0.3 percent.

The flow is produced by two Joy Axivane Fans, mounted in series aft of the test section. Each has an internal 1750 rpm, 100 horsepower motor. The fan blades are adjustable from 25 to 55 degrees. A set of multileaf-type inlet vanes immediately in front of each fan assembly allows the operator to control the mass flow rate and pre-swirls the flow in the direction of fan rotation, thereby reducing the power required to operate the wind tunnel. Flows of 10 to 250 feet per second can be produced. Figure 2 is an overall view of the wind tunnel.

3
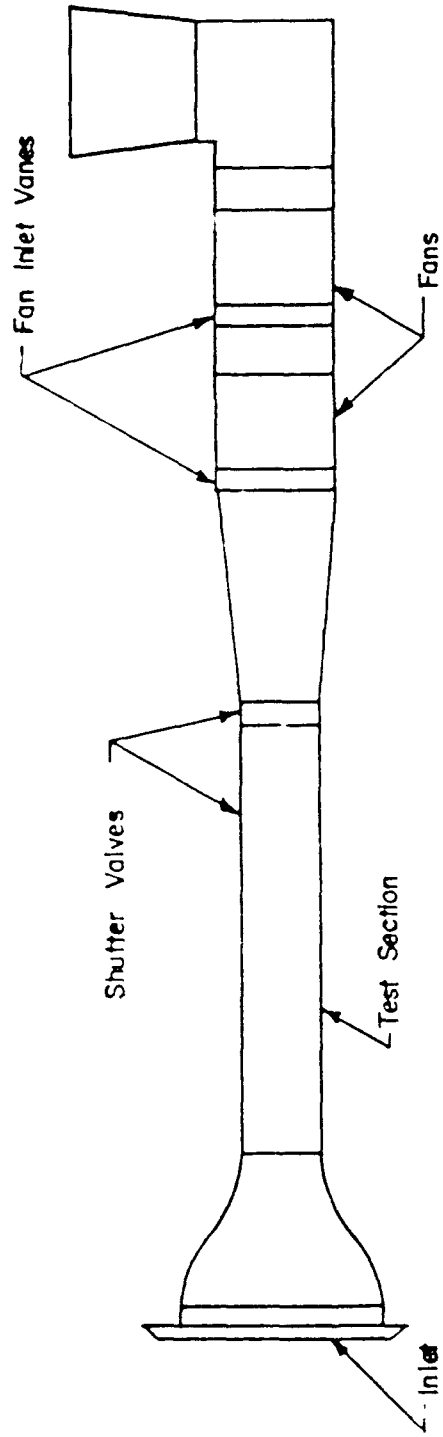
Figure 1. Plan View of Wind Tunnel.

4

Figure 2. Overall View of Wind Tunnel.

2. Test Section

The test section is 223 inches long and has a twenty-four inch square cross section. The top and bottom panels are constructed of continuous two-inch thick aluminum slabs. Access ports through the bottom slab permit the mounting of hotwires and pitot-static probes; they may be sealed off if not in use. The front sidewall is constructed of three two-inch thick stress-relieved Lucite panels, each secured by twelve bolts to the top and bottom slabs and to the vertical frames that divide the test section into three sections. The three front wall panels can be raised by hydraulic actuators to allow access to the test section. The rear side panels are two-inch thick plywood, and can be used for mounting internal or external instrumentation. The extraordinarily heavy construction of the test section is intended to minimize deflection due to rapidly changing static pressure in an oscillating flow.

3. Rotating Shutters

The method used to generate the oscillating flow was first used by Karlsson [Ref. 1], and later by Miller [Ref. 2]. The scheme consists of four equally spaced shutter vanes, mounted on horizontal shafts which span the wind tunnel horizontally just aft of the test section and protrude through the rear vertical side. The bottom shaft is driven by an electric motor, while the other shafts are each driven by the one immediately adjacent by means of a timing belt and pulley arrangement. This serves to keep all four vanes in phase as they rotate. Alternatively, the vanes can be locked in the horizontal plane to allow the tunnel to produce a steady flow. The electric motor's speed can be

varied by the operator to produce an oscillation at any frequency from 2 to 240 Hz. The amplitude of flow oscillation about the mean can be altered by varying the width of the shutter vanes, and thus the degree of test section closure. In this work, shutter vanes of six-inch width were used, resulting in nearly 100 percent closure. With a mean flow velocity of 50 feet per second, oscillation amplitudes of approximately 35 feet per second RMS about the mean velocity were produced. Figure 3 is a photograph of the shutter vanes.

B.   MODEL

The airfoil model was identical to that employed by Banning [Ref. 3]. It is a NACA 63-010 section modified by a straight line fairing from 60 percent chord to th trailing edge. The model has a 24-inch span and a 6-inch chord. Twenty-three pressure taps are distributed chordwise at mid-span on the top surface of the airfoil. Two additional taps are located on the underside, directly opposite ports number 11 and 16, to facilitate the calibration of a 0° angle of attack. Figure 4 shows the general shape of the airfoil and the location of the pressure ports.

The airfoil model was pivoted at mid-chord for support and angle of attack control. Angle of attack was adjusted by means of a vernier mechanism, permitting the operator to set any desired angle from -40 to +40 degrees relative to the tunnel centerline. The pressure port leads penetrated the rear side wall through the angle of attack mechanism. They were then connected to the Scanivalve pressure measurement equipment, which is described below. Figure 5 is a photograph of the model mounted in the test section.
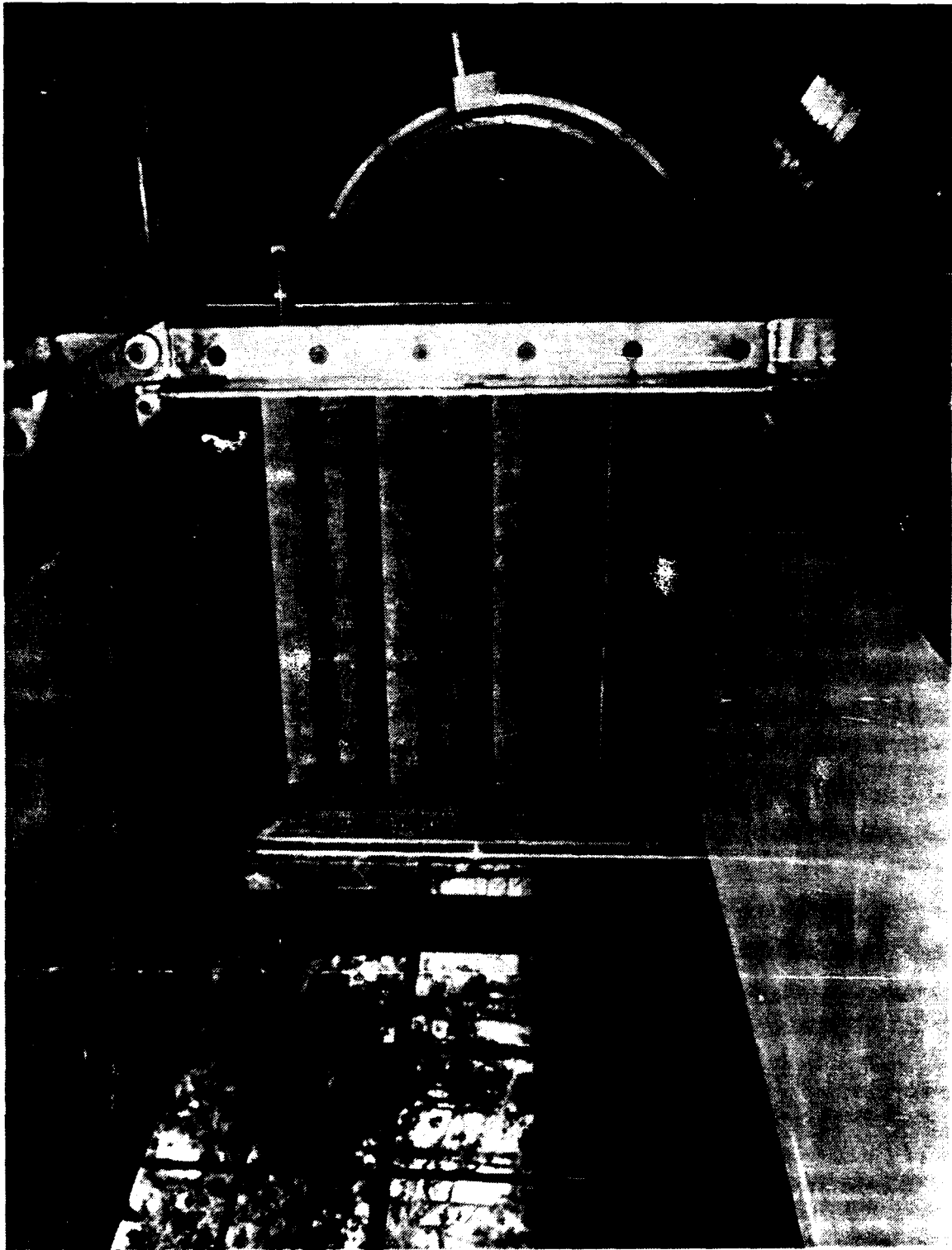
Figure 3. Shutter Vanes Aft of Test Section.

8

Figure 4. Pressure Tap Locations on Airfoil Model.

Mahogany Overlay

Steel Spar

STATION	DISTANCE (x/c)
1	.00
2	.025
3	.050
4	.075
5	.100
6	.125
7	.150
8	.175
9	.200
10	.225
11	.250
12	.300
13	.350

STATION	DISTANCE (x/c)
14	.400
15	.450
16	.500
17	.600
18	.700
19	.750
20	.800
21	.850
22	.900
23	.950
*24	.250
*25	.500
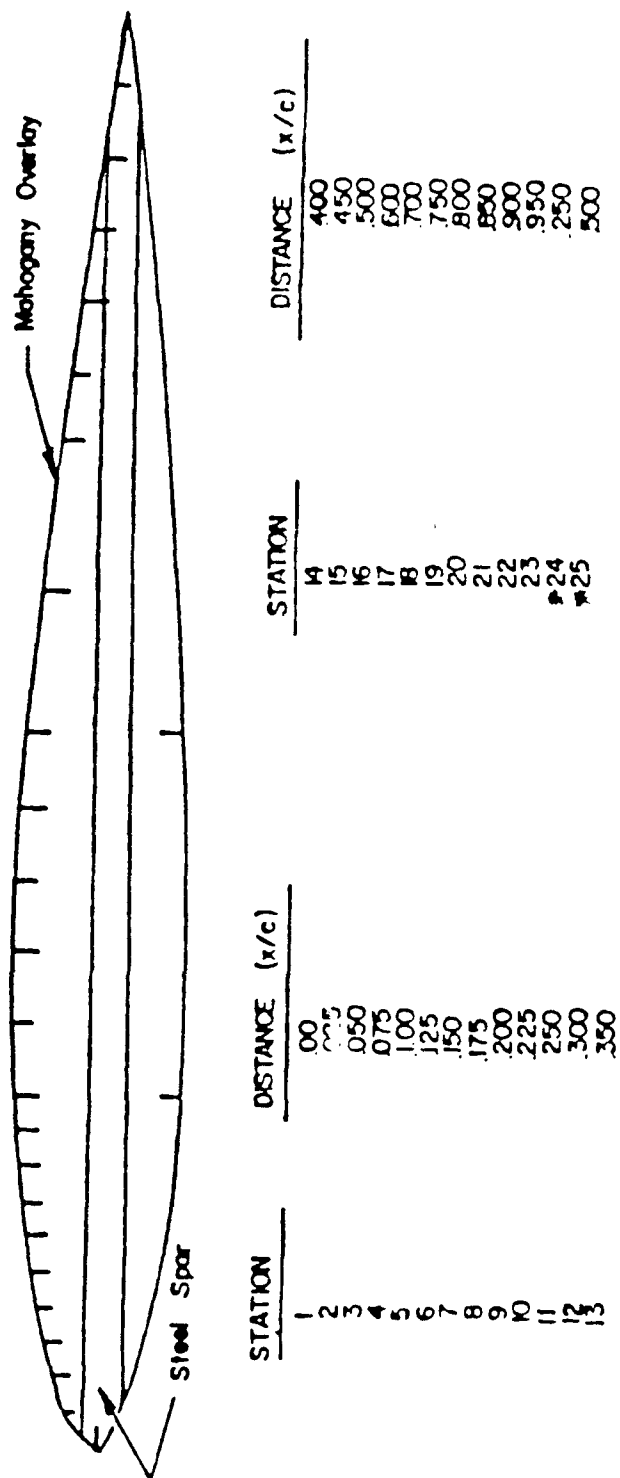
*Taps 24 & 25 Are On Lower Surface

C. INSTRUMENTATION

1. Velocity Measurement

Two methods were used to measure flow velocity during experimental runs. First, a pitot-static tube was installed approximately 15 inches ahead of and 6 inches below the leading edge of the model, 4 inches from the side wall. The pitot-static tube was connected to a water micro-manometer and was used to establish a known steady
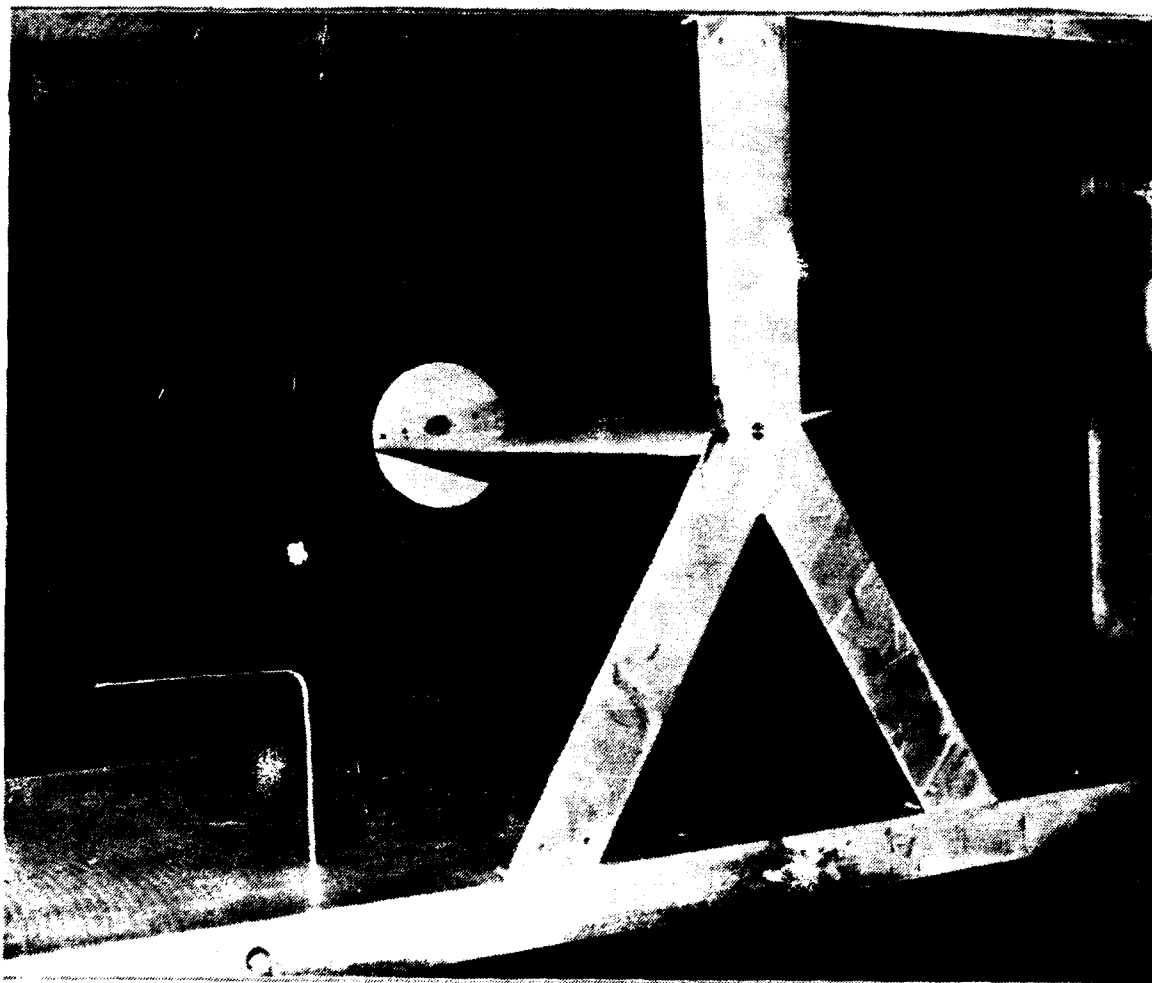
Figure 5. Airfoil Model Mounted in Test Section.

10

flow speed in order to calibrate the hotwire anemometer, and for measuring the mean flow velocity in oscillating flows. It should be stated that even with oscillations at 10 Hz, which produced velocities ranging from 0 to 80 feet per second, the mean velocity of 50 feet per second was displayed steadily by the water manometer. All pressure fluctuations due to the velocity oscillation were damped out by the pitot-static tube and manometer.

A hotwire anemometer was employed to measure the instantaneous velocity during all experimental runs. The hotwire was installed approximately 3 inches ahead of and 8 inches below the model leading edge, 6 inches from the side wall. The hotwire and pitot-static tube are visible in Figure 5.

The linearized, constant temperature hotwire was described by Miller [Ref. 4].

2. Pressure Measurement

Instantaneous pressure was measured by means of a Scanivalve pressure transducer and signal conditioner. The Scanivalve unit was mounted on a free-standing mobile dolly and positioned adjacent to the model's pressure port tubes where they exited the test section. This arrangement served to isolate the Scanivalve from the wind tunnel, thereby minimizing mechanical coupling and signal noise due to wind tunnel vibration.

The model's twenty-five pressure ports were connected to the Scanivalve by a set of uniform length thin-walled stainless steel tubes, each 24 inches long and with 0.047 inches inner diameter. An additional lead was connected to a spare Scanivalve port for use in calibrating the pressure transducer.

11

The uniform length of the model's pressure port tubing allows a single transfer function to be applied to the instantaneous pressure measurements from any port to account for phase lag and amplitude damping in the pressure signal due to the pressure port plumbing. A calibration was performed on the tubing to determine the transfer function necessary to convert the pressure signals received at the Scanivalve back to a signal representative of the pressure sensed at the model. An acoustic driver was used to generate a pressure signal of varying frequency at one end of the tubing. The pressure was measured at the source and at the remote end of the tube, and the resulting amplitude ratio and phase shift were plotted as a function of frequency. This technique has been described in detail by Banning [Ref. 3]. The results of this calibration are depicted in Figures 6 and 7.

3. Data Acquisition System

Instantaneous pressure and velocity measurements were recorded and stored on separate 64-K capacity recording boards installed in a 12 MHz 80286-based microcomputer. The data acquisition program was cued by an electrical trigger, which was actuated mechanically by the rotating shutter vanes. The data acquisition software armed the two data recording boards and, when triggered, the boards simultaneously recorded 1024 samples of pressure and velocity over a period of 1.024 seconds. This was programmed to match the oscillation frequency of 10 Hz, which was selected by the operator at the beginning of the experimental session.

The pressure and velocity signals were output continuously from the Scanivalve pressure transducer and the hotwire anemometer, respectively, as DC voltages

12
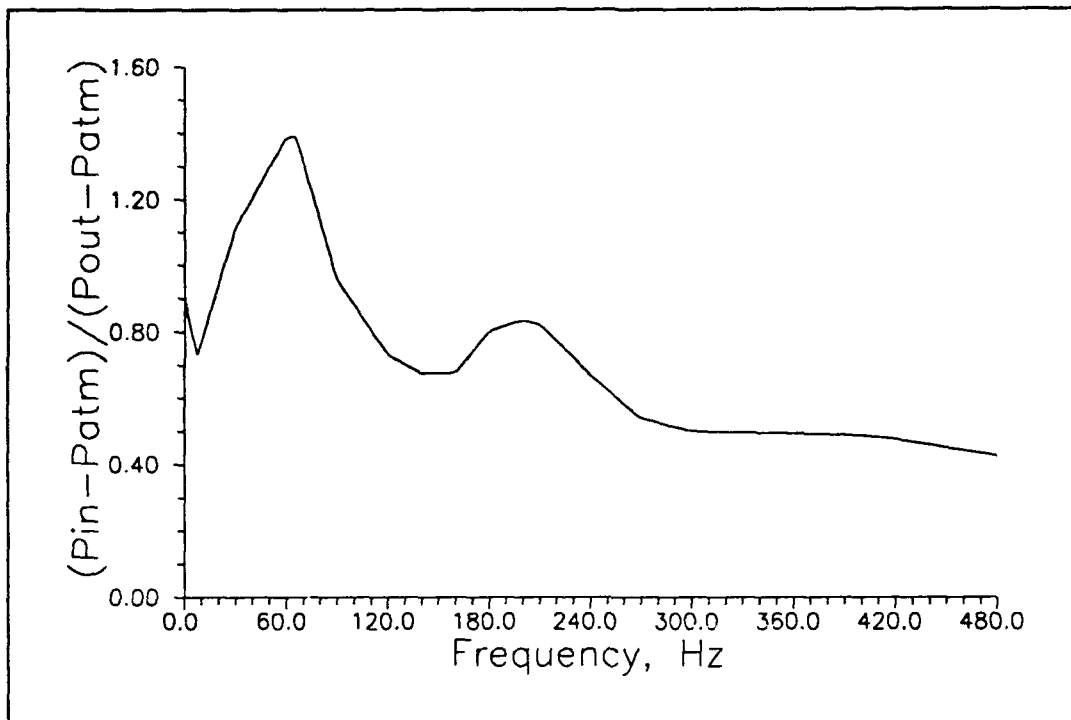
Figure 6. Transducer Gain as a Function of Frequency.
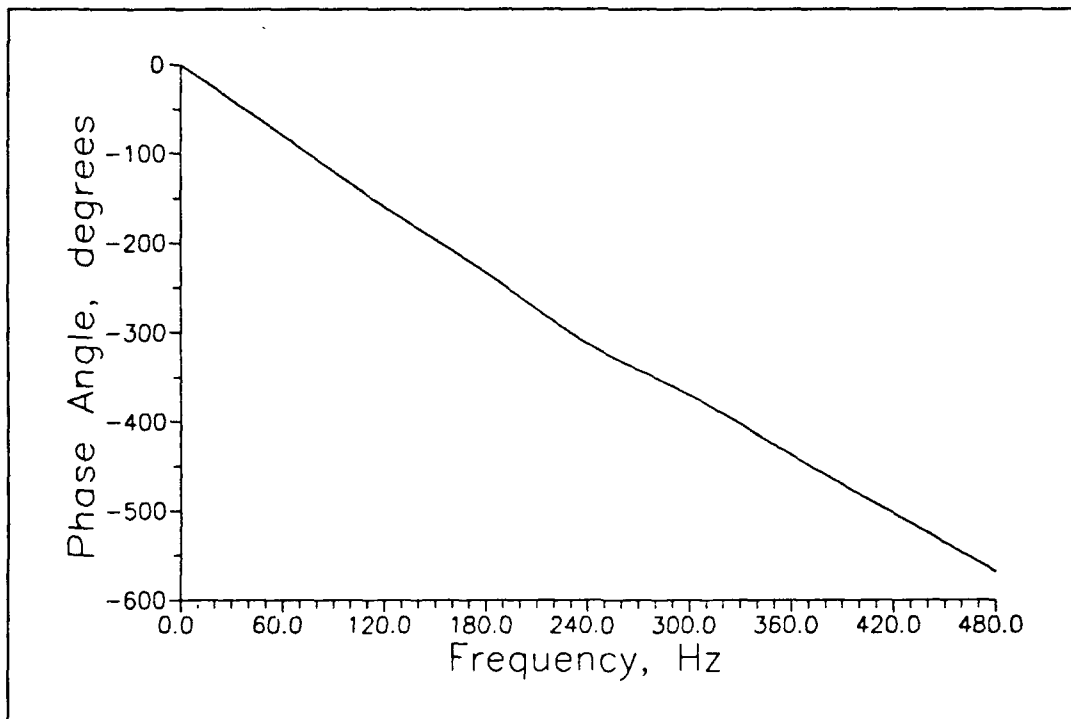
Figure 7. Transducer Phase Shift as a Function of Frequency.

13

corresponding to a percent deflection of full scale, where full scale refers to a full scale range on the data acquisition boards. In both cases, full scale deflection was limited to plus or minus five volts. As mentioned, the pressure and velocity analog signals were sampled at slightly greater than 100 times the 10 Hz frequency of the oscillating flow to produce a sample size of 2^N, which was necessary for compatability of the data acquisition system. The recording boards are capable of data acquisition up to a data rate of 1 MHz.

The recording boards are of 12-bit resolution, capable of differentiating 1 part in 4096. By calibrating the pressure signal recording board response for a maximum signal of plus or minus 1 psi, an ability to resolve 0.0005 psi was achieved. The velocity recording board was calibrated to a maximum velocity of plus or minus 125 feet per second. therefore its resolution was 0.061 feet per second.

4. Controls

The operator's console consisted of several panels of instrumentation which enabled the experimenter to control and monitor nearly every aspect of the experiment in progress. Figure 8 is a photograph of the console system. On the far left is a Tektronix Type 555 Dual Beam Oscilloscope, used to monitor the pressure and velocity signals in real-time. On the left end of the left console is the control panel for a the ten-channel linearized hotwire system. Only one channel was used in the present work. To the right of the hotwire system is a Singer Model 303-01 RMS Voltmeter, used to monitor the RMS velocity fluctuation during oscillating flow experiments. The DC voltmeter to its right may be used to monitor the mean velocity. On the upper right side

14

Figure 8. Operator's Console System.

15

of the left console is a Dynasciences Signal Counter, which continually measured and displayed the rotating shutter frequency. Below that is an Analog Devices AD2501 digital thermometer, which displayed freestream temperature in degrees Fahrenheit, measured by a thermocouple just downstream of the model. On the lower panel are located the hydraulic switches used to raise and lower the test section access doors, the mass flow rate adjustment switches which control the degree of closure of the multileaf vanes, the shutter activation and speed controls, and the emergency shutoff switches for the two electric fan motors. The water manometer used to establish tunnel velocity for calibration and to measure freestream mean velocity in oscillating flows sits atop the desk in front of the left console.

The right-hand console incorporates the power supply for the data acquisition trigger, the Scanivalve Position Display and solenoid controller, and the data acquisition/reduction microcomputer. The computer is an Everex System 1700 Model-C AT clone with an Intel 80286-12 CPU chip and an 80287 math coprocessor chip. Also seen are an oscilloscope and an Ampex PR2200 magnetic tape recorder, which may be used to monitor and record analog signals. On the table next to the console are an HP 7470A plotter, IBM Proprinter II and an IBM PC/XT microcomputer, used for additional data reduction capability.

III. EXPERIMENTAL PROCEDURES

A. CALIBRATION

1. Flow Velocity

Prior to each run, the hotwire anemometer was calibrated ·sing a water manometer to establish a known velocity for comparison. Calibration points were taken at 0 and 100 feet per second for every run. First, a zero velocity was assured by placing a cover over the hotwire probe. In this condition, the hotwire current was adjusted at maximum gain to obtain an output of 0 volts, to an accuracy of 0.0001 volts. Then the wind tunnel was started and the flow adjusted to generate a water column of the proper height for 100 feet per second, based on local barometric pressure and temperature. The hotwire gain control was adjusted to obtain an output of 4.0000 volts to an accuracy of 0.0005 volts. To avoid over-voltage of the data acquisition system, a limit of 5 volts, corresponding to 125 feet per second, was established for velocity measurements. For calibration, these voltages were determined by the digital voltmeter in the linearized hotwire circuit.

2. Pressure Measurement

Pressure measurement calibration was performed by first setting the Scanivalve pressure transducer output equal to zero with the calibration pressure port open to the atmosphere. Then, the pressure transducer voltage gain was adjusted to read -2.50 volts

with an external pressure input of 0.50 psi above atmospheric pressure applied, as read on a water column manometer. Thus, the voltage output of the Scanivalve would represent:

$$P_{Scanivalve} = P_{Model} - P_{Atmosphere} \qquad (1)$$

The maximum voltage span for the Scanivalve was set at plus or minus 5 volts, allowing a pressure equal to atmospheric plus or minus 1.0 psi to be measured without causing over-voltage of the data acquisition system. The pressure calibration voltages were determined by using a Fluke Digital Voltmeter, with an accuracy of 0.001 volts.

3. Data Acquisition System

The data acquisition system boards were calibrated for a true zero reading at a 0.0 volt input, and set to accept a 10-volt span, corresponding to a reading of 2048 digital counts at +5.00 volts and -2047 digital counts at -5.00 volts, for both pressure and velocity measurements. The calibration results are presented in Figure 9, which shows a linear, symmetrical response centered within 0.1 percent of true zero.

B. TUNNEL OPERATION

1. General

Once calibration procedures were completed, the tunnel velocity, oscillation frequency and model angle of attack were set at the desired values. The operator supplied information regarding tunnel operating conditions and ambient conditions (pressure, temperature, etc.) to the data acquisition program for inclusion in the data file header.

18
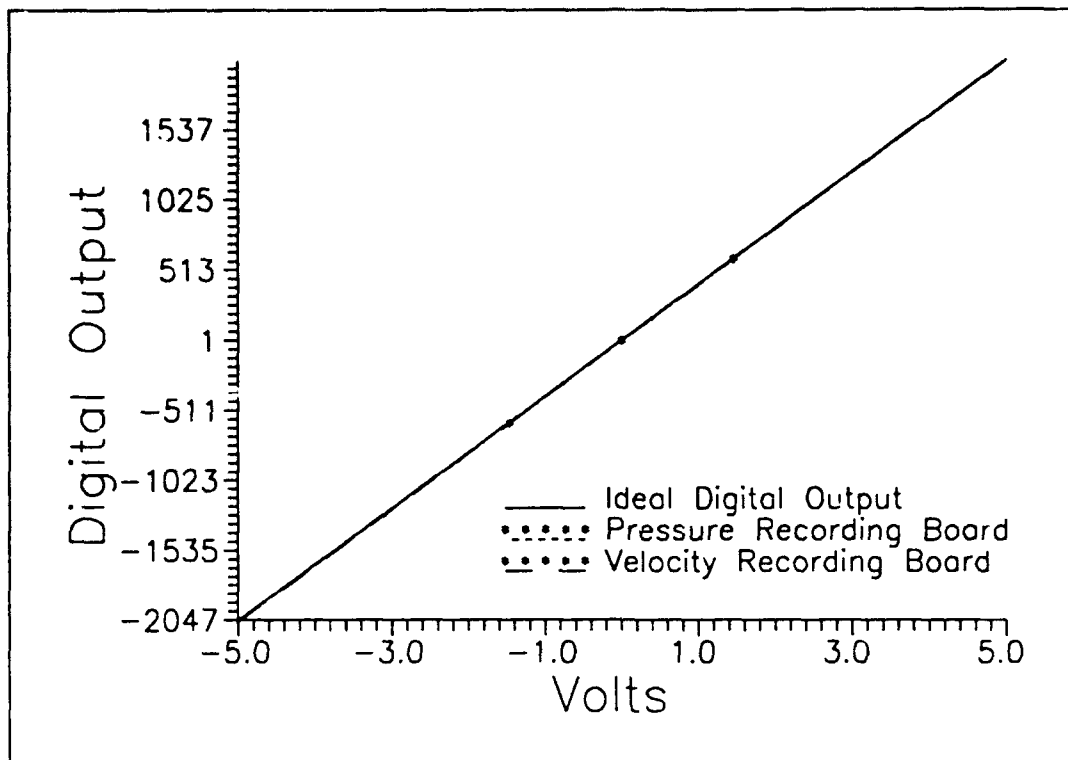
Figure 9. Data Acquisition Boards Calibration Results.

The computer controlled the actual data acquisition process, pausing after reading the first 23 pressure ports to allow the operator to rotate the model to a negative angle of attack. Thus, the same pressure taps represented both the upper an lower surface of the symmetrical airfoil. The tunnel velocity and Scanivalve pressure and port selection were monitored continually by the operator to ensure proper operation throughout the run.

2. Steady Flow

For steady flow operation, the rotating vanes were secured fully open in the horizontal position. A flow velocity of 100 feet per second was established as measured by the pitot-static tube and water manometer, with the model at the desired angle of attack. Model angles of attack from 0 to 20 degrees, in increments of 5 degrees, were

investigated to establish baseline values for aerodynamic coefficients and steady-state pressure distribution as a function of angle of attack. The chord Reynolds number for steady flow work was 306.000, where the Reynolds number was given by:

$$R = \frac{\rho \bar{U} c}{\mu} \qquad (2)$$

3. Oscillating Flow

All oscillating flow runs were made at an oscillation frequency of 10 Hz. The mean flow velocity was established at 50 feet per second as measured by the pitot-static tube and water manometer, with the vanes rotating and the model fixed at the desired angle of attack. The velocity oscillations were of the order of plus or minus 70 percent of the mean value. Model angles of attack from 0 to 35 degrees, in increments of 5 degrees, were investigated. A Reynolds number of 152,000 and a Strouhal number of 0.314, referenced to the mean flow velocity and the model chord length, were used for all oscillating flows. Strouhal number was determined by the relation:

$$S = \frac{2\pi f \frac{c}{2}}{U} \qquad (3)$$

A typical Strouhal number for rotary wing aircraft is on the order or 0.025.

IV. DATA ANALYSIS

A. DATA ACQUISITION

An automated data acquisition program controlled the progress of the experimental runs under the supervision of the operator. Because the experimental data, consisting of the time-dependent pressures and velocities, were stored in order of occurrence as voltages on the data acquisition boards, a data acquisition program was written in Quick BASIC, which permitted the interrogation of specific memory locations on the recording boards and retrieval of discrete pieces of information related to the control of the experiment.

During the data acquisition cycle, data was taken for 1.024 seconds at a time, which spans 10 cycles of flow oscillation. The signals were ensemble averaged to attenuate the effect of transients in the flow, and the time-dependent portion of each signal was separated from its respective mean value for each pressure port location. The time-dependent portion of the pressure signal then underwent a Fourier transform and was divided by the previously determined transfer function, correcting for the amplitude damping and phase shift associated with the pressure tap connecting tubes. Next, an inverse Fourier transform was applied. This resulted in a discrete representation of the pressure distribution at each port and the corresponding velocity profile over a single cycle of flow oscillation. By separating the time-dependent portion of the two signals,

21

greater accuracy could be achieved in the Fourier transforms and transfer function processes.

Output files were formatted such that the header information was listed at the top of the file, and could be accessed for data reduction. This was followed by the 46 average pressures, 46 average velocities, 4600 values of the time-dependent pressure signal and 4600 values of the time-dependent velocity signal. A naming convention was adopted for easy file identification, wherein the output files were named RUNxyy.DAT, where 'x' refers to the run series (0 for steady flow, 1 for oscillating flow) and 'yy' refers to the angle of attack, in degrees. Figure 10 is a block diagram of the data acquisition program, and a complete program listing appears in Appendix A.

B.   DATA REDUCTION

The data reduction program was written in FORTRAN for simplicity and for speed in repetitive calculations. The program used the RUNxyy.DAT file in conjunction with another file, XOVRC.DAT, containing the pressure port chordwise locations, and wrote information to nine output files including the upper and lower surface pressure, velocity and pressure coefficient as a function of chordwise position and cycle time, and regarding the C_L, C_D and $C_{Mc/4}$ values as a function of cycle time and angle of attack. The pressure and velocity files were examined to ensure correlation with what was observed on the oscilloscope during the experiments. The C_P, C_L, C_D and $C_{Mc/4}$ files are included in this report as two-dimensional curves for steady flow results and as three-dimensional surface plots for oscillating flow results, in Chapter V.

22

READ TRANSFER FUNCTION
This correlates the pressure
read by the Scanivalve to the
actual pressure, P(t), which is
experienced at the model

READ HEADER INFORMATION
Program queries operator for
run number 'xyy', AOA, P_{Atm}, T
and flow oscillation frequency

SET SCANIVALVE TO HOME

ADVANCE SCANIVALVE TO PORT 1

INITIALIZE RECORDING BOARDS
AND ANALOG-TO-DIGITAL SYSTEM
Performed once, prior to each
data run, to establish various
recording board control parameters

FOR PORT = 1 TO 23

READ PORT NUMBER

RECORD PRESSURE & VELOCITY
1024 values of P and V are
recorded simultaneously by
the recording boards.

LOAD P, V INTO ARRAYS AND PROCESS
Ensemble average 10 cycles' worth
of P, V values, separate P_{MEAN} and
V_{MEAN} from P(t) and V(t). Apply FFT,
transfer function & IFFT to pressure
signal only. Scale P to lbf/ft^2, V to
ft/sec.

NEXT PORT

PAUSE TO ALLOW OPERATOR TO SET
NEGATIVE AOA, REPEAT LOOP TO
RECORD LOWER SIDE DATA

WRITE OUTPUT FILE RUNxyy.DAT

Figure 10. Data Acquisition Program Flow Chart.

As previously described, the Scanivalve output was set to

$$P_{Scanivalve} = P_{Model} - P_{Atm} \tag{1}$$

Total pressure was found to be equal to atmospheric pressure, due to the low inlet velocity through the high solidity inlet screens. Thus,

$$P_0 = P_{Atm} = P_\infty + q_\infty \tag{4}$$

The coefficient of pressure is given by

$$C_P = \frac{P_{Model} - P_\infty}{q_\infty} \tag{5}$$

Substituting (1) and (4) into (5) yields

$$C_P = \frac{[P_{Scan} + P_{Atm}] - [P_{Atm} - q_\infty]}{q_\infty} \tag{6}$$

which further simplifies to

$$C_P = \frac{P_{Scan} + q_\infty}{q_\infty} \approx \frac{P_{Scan}}{q_\infty} + 1 \tag{7}$$

This result was applied to each pressure value to compute the corresponding C_P value. The resulting distribution with respect to cycle time and chordwise parameter was numerically integrated in the following manner

24

$$C_N = \int_{\frac{x}{c}=0}^{1} [C_{P_{lower}} - C_{P_{upper}}] \, d\frac{x}{c} \approx \sum_{i=1}^{23} [C_{P_{l_i}} - C_{P_{u_i}}] * \frac{1}{2} * [\frac{x}{c_{i+1}} - \frac{x}{c_{i-1}}] \tag{8}$$

to obtain a normal force coefficient, C_N. This was resolved into lift and drag components by applying the cosine or sine of the angle of attack, as appropriate. Likewise, the following computation

$$C_{M_{\frac{c}{4}}} = \int_{\frac{x}{c}=0}^{1} [C_{P_l} - C_{P_u}] * [0.25 - \frac{x}{c}] \, d\frac{x}{c} \tag{9}$$

$$\approx \sum_{i=1}^{23} [C_{P_{l_i}} - C_{P_{u_i}}] * [0.25 - \frac{x}{c_i}] * \frac{1}{2} * [\frac{x}{c_{i+1}} - \frac{x}{c_{i-1}}] \tag{10}$$

was applied to obtain the moment coefficient about the quarter chord, $C_{M^{c/4}}$. A block diagram of the data reduction program appears in Figure 11, and a complete program listing appears in Appendix B.

C. UNCERTAINTY ANALYSIS

As previously stated, the data acquisition system was calibrated to determine the linearity of response to a known input and the accuracy of the digital count with reference to true zero. The calibration data is reported in Figure (9). True zero appeared at -5 of

25

2047 counts on the velocity board, and at -3 of 2047 counts on the pressure board. The slope of both boards was linear and consistent with the ideal slope, corresponding to a count of -2047 at an input of -5.0 volts and a count of +2048 at an input of +5.0 volts, to within 0.1 percent. With the range of the pressure board set at plus or minus 1.0 psi, or 144.0 psf, this corresponds to a maximum error of 0.144 psf, or 0.001 psi in reading a full-scale deflection pressure of 1.0 psi. Similarly, for a full scale velocity reading of 125 feet per second, the maximum error would be 0.125 feet per second. Since the maximum values encountered were approximately 0.3 psi and 85 feet per second, the uncertainty in these measurements is of the order of 0.1 percent.

The sensitivity of the Scanivalve pressure transducer is guaranteed to be 17.5 millivolts, plus or minus 3 millivolts, as published in the Scanivalve calibration certificate. Given the voltage and pressure scale in use, this results in a contribution to the uncertainty of 0.41 percent due to the Scanivalve.

The water manometer is marked in graduations that permit the operator to distinguish 0.005 centimeters of water.

OPEN RUNxyy.DAT, XOVRC.DAT

READ NECESSARY HEADER INFO

RECONSTRUCT PRESSURE, VELOCITY
DISTRIBUTION FROM P_{MEAN}, V_{MEAN},
$P(t)$ & $V(t)$

COMPUTE C_pUPPER AND C_pLOWER
AND STORE IN ARRAYS

WRITE P, V, C_p OUTPUT FILES
FOR THIS RUN NUMBER

INTEGRATE C_p DISTRIBUTION
TO OBTAIN C_N, $C_M c/4$

COMPUTE C_L, C_D AND WRITE
THESE PLUS $C_M c/4$ TO THEIR
RESPECTIVE OUTPUT FILES
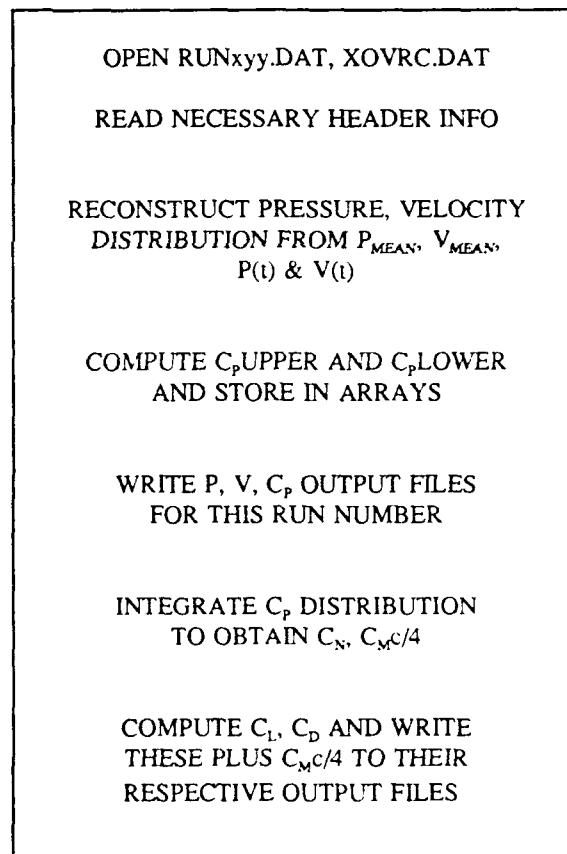
Figure 10. Data Reduction Program Flow Chart.

26

For an oscillating flow of 50 feet per second mean velocity, this corresponds to an uncertainty of 0.41 percent.

The cumulative result of these uncertainties in calculation of C_P, assuming the worst case of an overestimation of the Scanivalve pressure reading combined with an underestimation of the dynamic pressure, may be calculated starting with the Scanivalve uncertainty:

$$P_{Scanivalve} = 1.0041 * P_{True} \tag{11}$$

Then the effect of recording board nonlinearity may be written:

$$P_{Rec.Board} = 1.001 * P_{Scanivalve} = 1.0051 * P_{True} \tag{12}$$

The effect of uncertainties in the value of the dynamic pressure, q, may be written:

$$q_{Input} = 0.9966 * q_{True} \tag{13}$$

These results may then be used to calculate the uncertainty in C_p:

$$Uncertainty = \frac{C_{P_{w/errors}} - C_{P_{True}}}{C_{P_{True}}} \tag{14}$$

or:

$$= \frac{[\frac{P_{Rec.Board}}{q_{Input}} + 1] - [\frac{P_{True}}{q_{True}} + 1]}{[\frac{P_{True}}{q_{True}} + 1]} \tag{15}$$

27

Finally, using equations (12), (13) and (15), one may write:

$$Uncertainty = \frac{[\frac{1.0051 * P_{True}}{0.9966 * q_{True}} + 1] - [\frac{P_{True}}{q_{True}} + 1]}{[\frac{P_{True}}{q_{True}} + 1]} \qquad (16)$$

or simplifying:

$$Uncertainty = \frac{[0.0085 * \frac{P_{True}}{q_{True}}]}{[\frac{P_{True}}{q_{True}} + 1]} \qquad (17)$$

Thus the uncertainty in the calculation of C_P approaches a maximum value of 0.85 percent

as the ratio $\frac{P}{q} > 1$.

The uncertainty in the coefficients, C_L, C_D and $C_{Mc/4}$, which are derived by integrating C_P, may be expected to be of a similar value. All computation in the data reduction program was performed using FORTRAN double precision, to avoid propagation of roundoff errors.

28

V. RESULTS AND DISCUSSION

A. STEADY FLOW RESULTS

The C_p distributions for the five steady flow runs are presented below in Figures 12-16. The plots in these figures were generated by averaging the corresponding C_p output files with respect to cycle time, t/T, since there were no oscillations in the flow. The resulting data points are all displayed, connected by a cubic spline curve of medium tension.
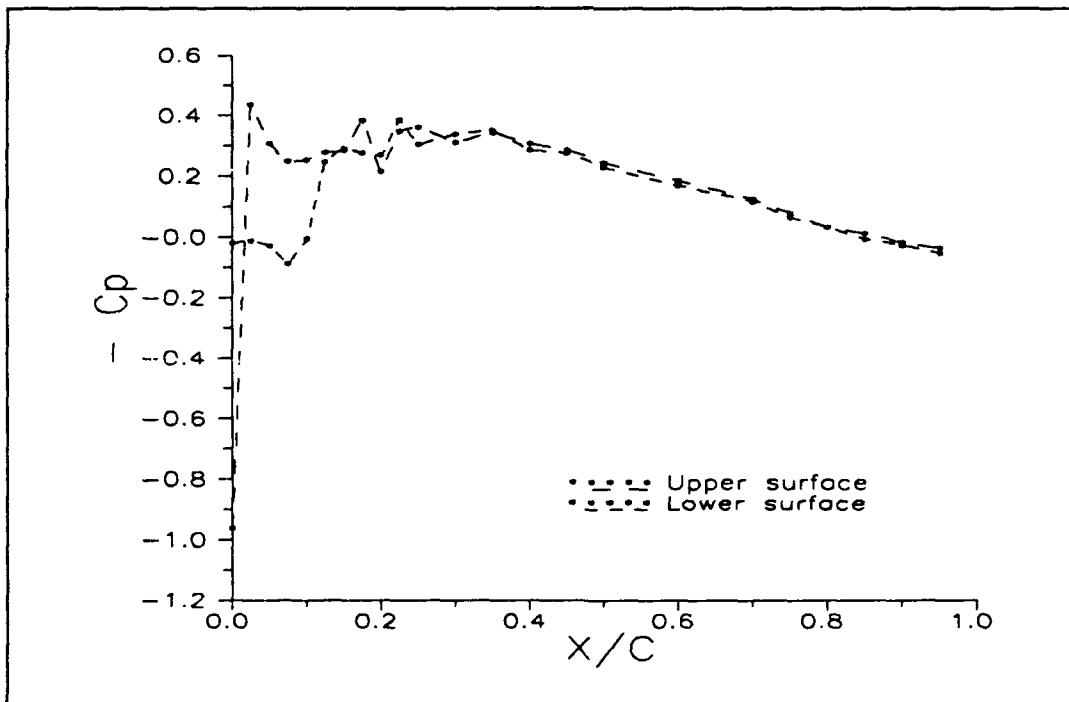
Figure 12. Pressure Coefficient Distribution at 0° Angle of Attack in Steady Flow.
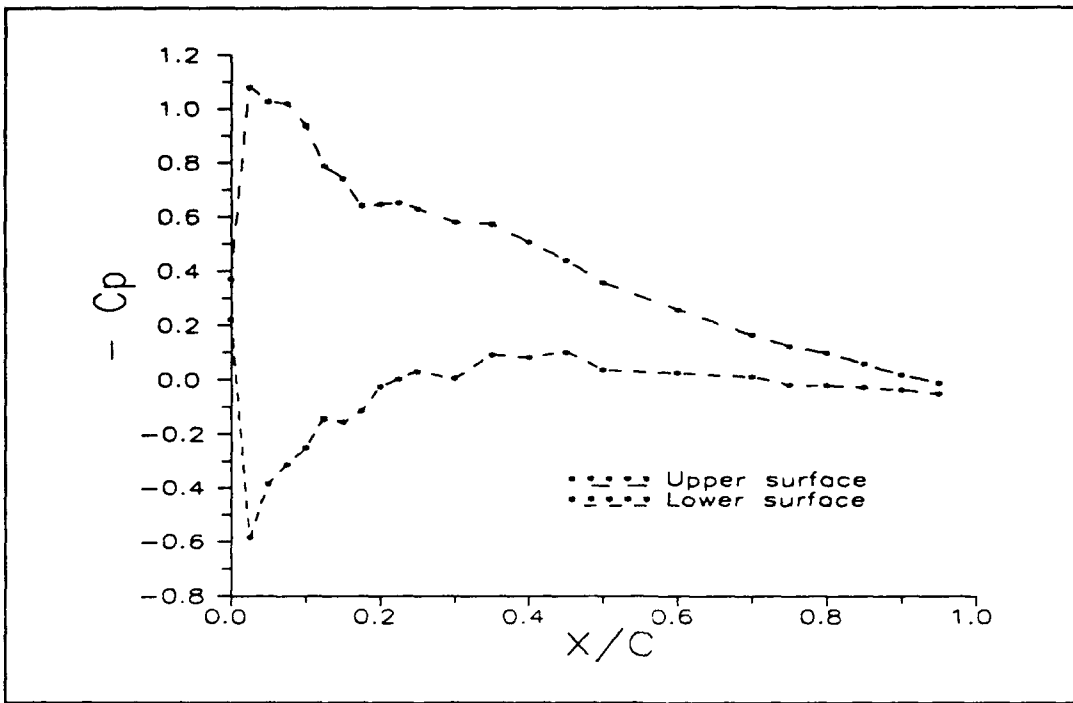
Figure 13. Pressure Coefficient at 5° Angle of Attack in Steady Flow.
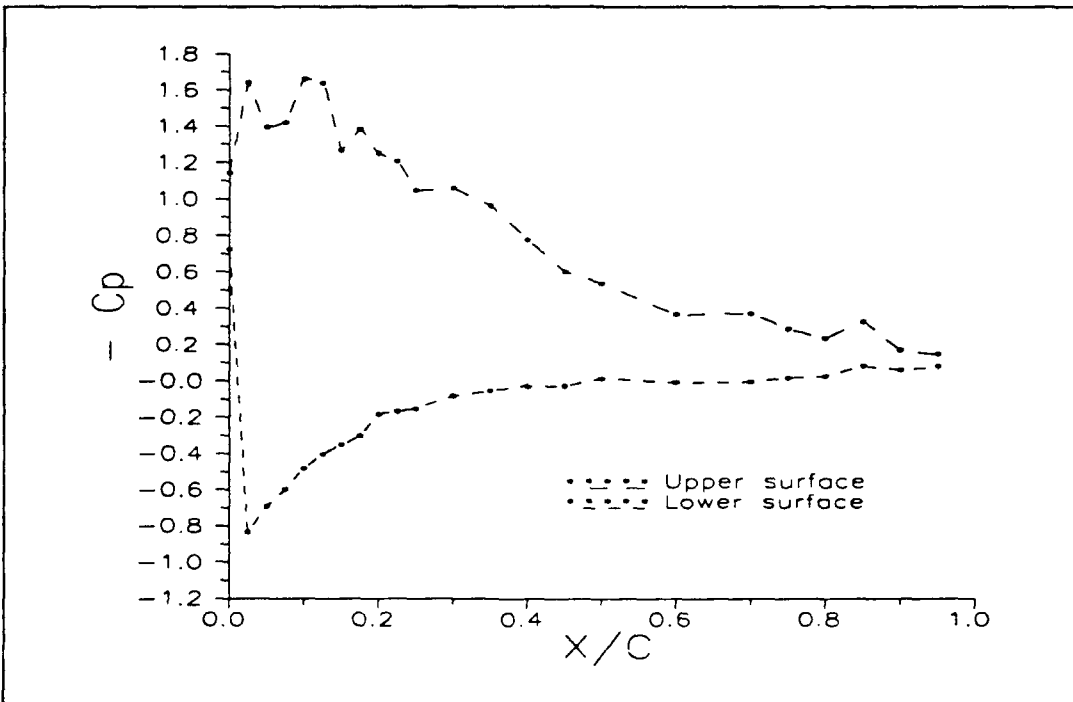
Figure 14. Pressure Coefficient at 10° Angle of Attack in Steady Flow.

30
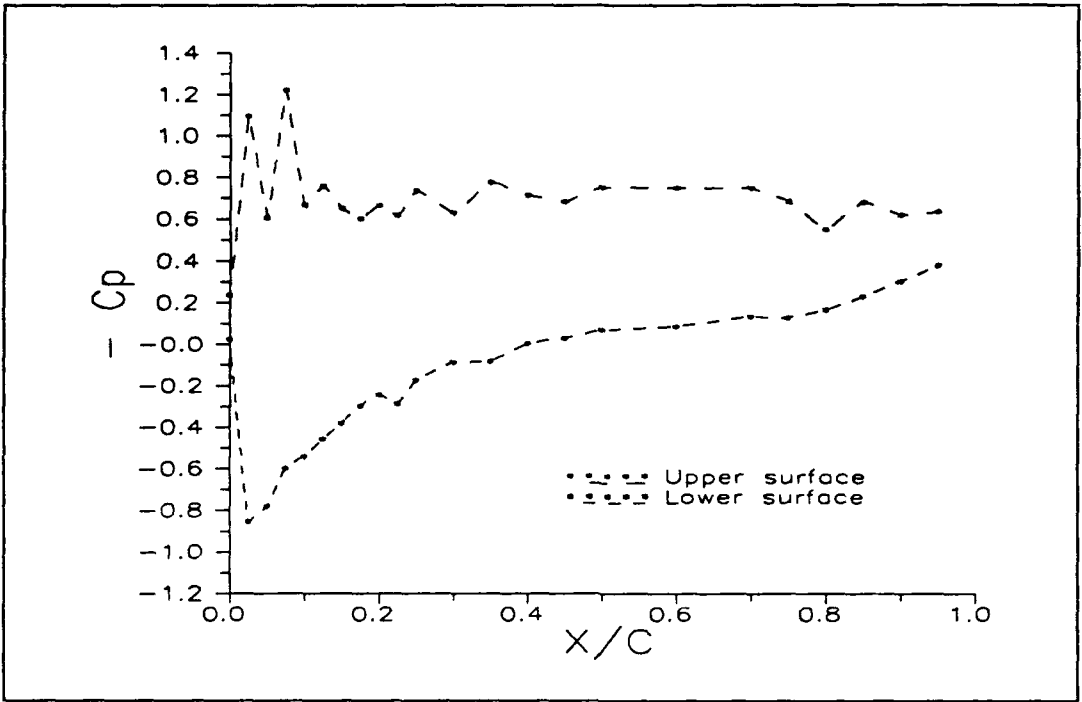
Figure 15. Pressure Coefficient at 15° Angle of Attack in Steady Flow.
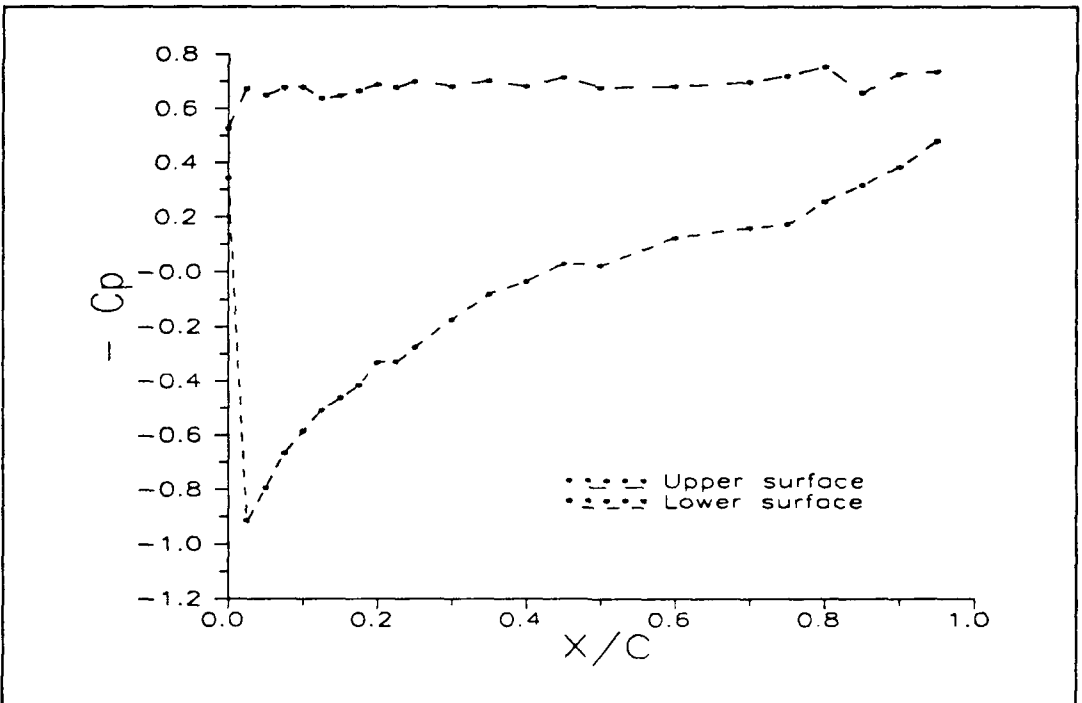
Figure 16. Pressure Coefficient at 20° Angle of Attack in Steady Flow.

31

The pressure distributions in steady flow are as one might expect for a symmetrical airfoil of this type. It can be seen that a slight leading edge separation bubble exists between ports 2 and 4, indicated by irregularities in several of the C_P curves at that location. The mean pressure distributions at 5, 10 and 15 degrees angle of attack are quite similar to those previously reported by Banning [Ref. 3]. C_P distributions at 0 and 20 degrees angle of attack were not reported by Banning.

The steady flow performance coefficients, C_L, C_D and $C_{Mc/4}$, are shown below as a function of angle of attack in Figures 17-19. The lift coefficient curve agrees well with that reported by Banning, who did not report drag or moment coefficients. The lift coefficient curve exhibits classical behavior for steady flow, with stall occurring at
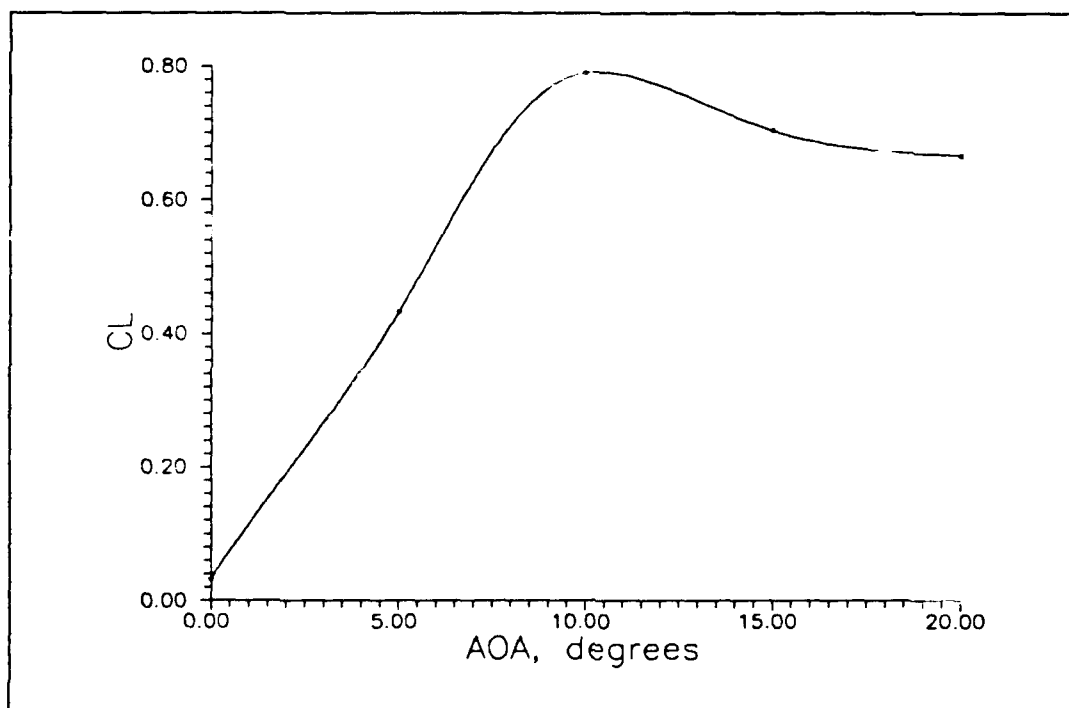
Figure 17. Lift Coefficient in Steady Flow.

32
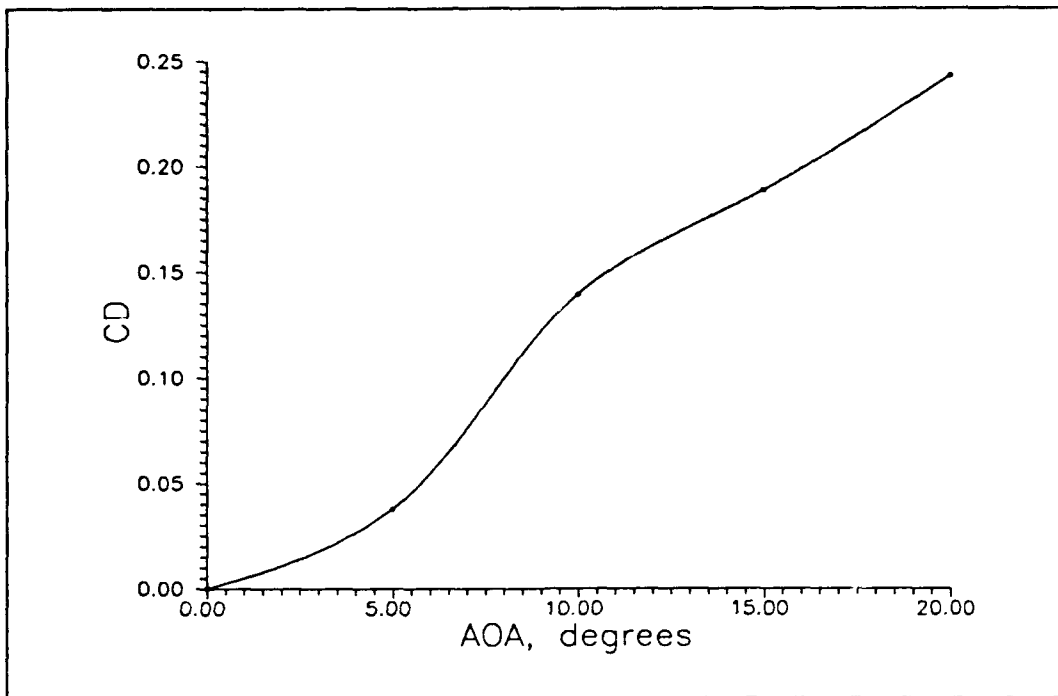
Figure 18. Drag Coefficient in Steady Flow.
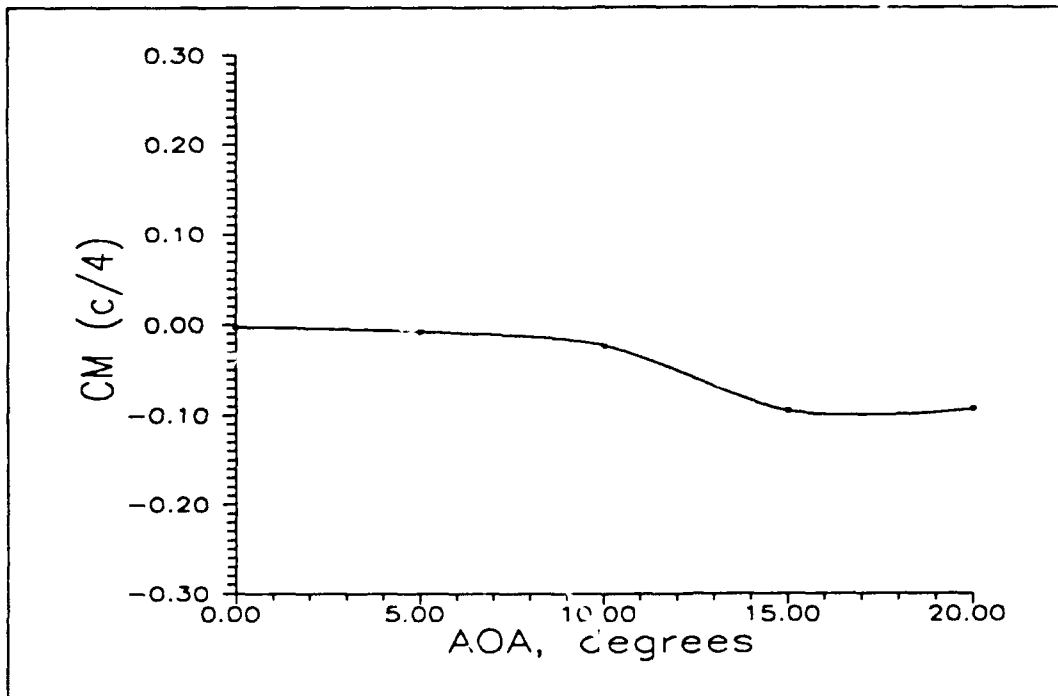
Figure 19. Quarter Chord Moment Coefficient in Steady Flow.

33

approximately 12 degrees. The offset from the origin at 0° angle of attack is attributed to possible pressure tap errors near the leading edge.

B.   OSCILLATING FLOW RESULTS

The C_P distributions a function of dimensionless time, t/T, and chordwise position, x/c, for angles of attack from 0 to 35 degrees, and their respective velocity profiles, are presented in Figures 20-43. The C_P distribution surface plots were generated by first creating a grid consisting of chordwise data at intervals of x/c = 0.025 and timewise data at intervals of t/T = 0.01. Thus some interpolation had to be made to create a 3900-point grid out of the 2300 existing data points. This was done using Surfer, a three-dimensional surface plot software package, which uses an inverse distance squared routine wherein

$$Z = \frac{\displaystyle\sum_{i=1}^{n} \frac{z_i}{d_i^2}}{\displaystyle\sum_{i=1}^{n} \frac{1}{d_i^2}} \tag{16}$$

In this equation, Z is the C_P value of the grid point to be calculated, d is the distance to the a neighboring grid point, and n is the number of nearest neighbors to be used in the calculations. A grid size of 0.025 by 0.02 was used, to ensure that only the existing data points were used, to the maximum extent possible, with a minimum number of grid points

34

being interpolated to fill in between any two actual data points. The surface plots are presented with only chordwise lines, that is, lines of constant time t/T. Additionally, the plots have been rotated and elevated so as to give the best view of the major features. The reader should therefore take care to note the orientation of the axis scales.

As is readily apparent, the pressure distribution over the airfoil was primarily a result of the oscillating flow and its concurrent oscillation in static pressure. As a consequence, chordwise pressure variations are very small compared to these induced static pressures. However, in spite of the lack of an orderly chordwise variation in C_P, one can see by comparing the maximum and minimum values of $-C_P$ found on the upper and lower surfaces that the upper surface appears to have higher peaks $-C_P$s at $t/T = 0.13$ and 0.55, whereas deeper valleys are found at $t/T = 0.40$ and 0.80 in the lower surface C_P surface plots. The net result, when the curves are integrated, is a non-zero normal force coefficient.

Some of the distributions exhibit random anomalies at a single chordwise location, usually just aft of mid-chord. These anomalies are most noticeable when they exceed the peak value of negative C_P and when they bridge the early, more shallow valley. In terms of the velocity cycle, this corresponds to the region in the velocity profile at which the flow deceleration suddenly becomes quite steep and quickly reaches a minimum. This feature of the C_P surface plots is attributed to the onset of separation. The condition remains generally only until the flow is re-established in a steep acceleration. This phenomenon is most noticeable on the upper surface of the airfoil, as would be expected.

35

Figure 20. Upper Surface Pressure Coefficient at 0° Angle of Attack in Oscillating Flow.

Figure 21. Lower Surface Pressure Coefficient at 0° Angle of Attack in Oscillating Flow.

Figure 22. Velocity Profile, RUN100.

37

Figure 23. Upper Surface Pressure coefficient at 5° Angle Of Attack in Oscillating Flow.

Figure 24. Lower Surface Pressure Coefficient at 5° Angle of Attack in Oscillating Flow.

Figure 25. Velocity Profile, RUN105.
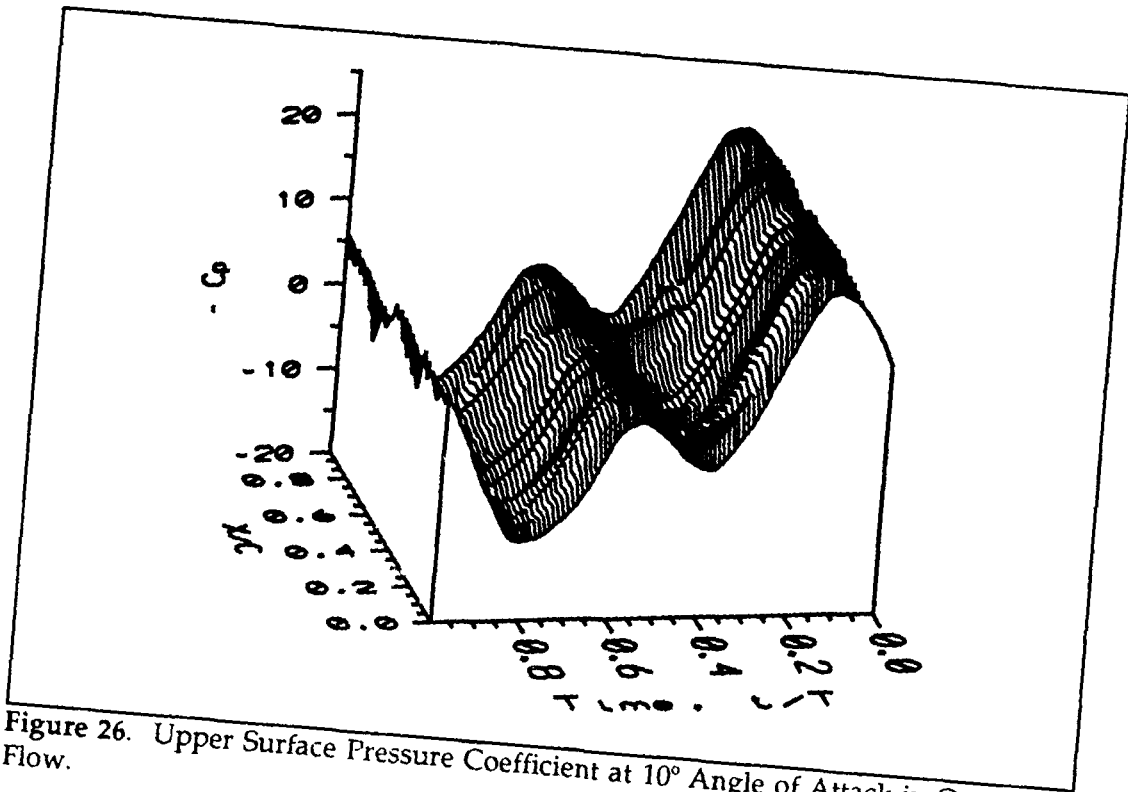
Figure 26. Upper Surface Pressure Coefficient at 10° Angle of Attack in Oscillating Flow.
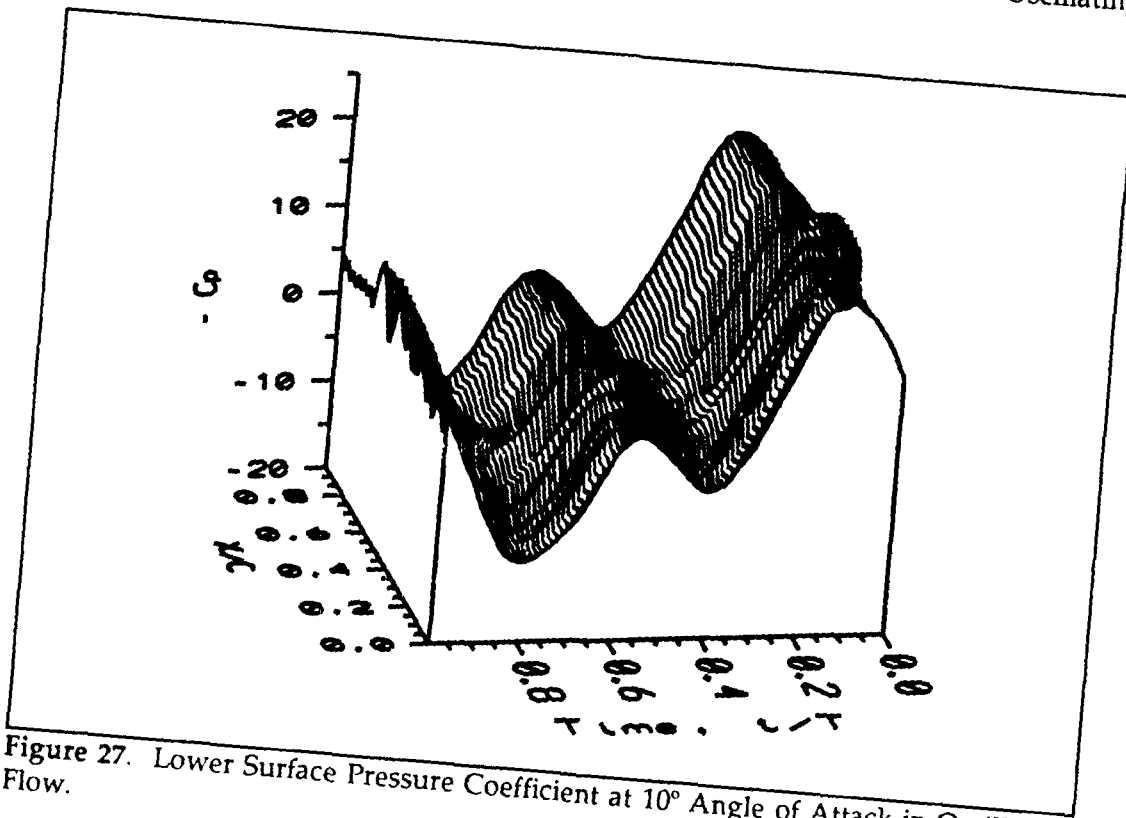
Figure 27. Lower Surface Pressure Coefficient at 10° Angle of Attack in Oscillating Flow.

40
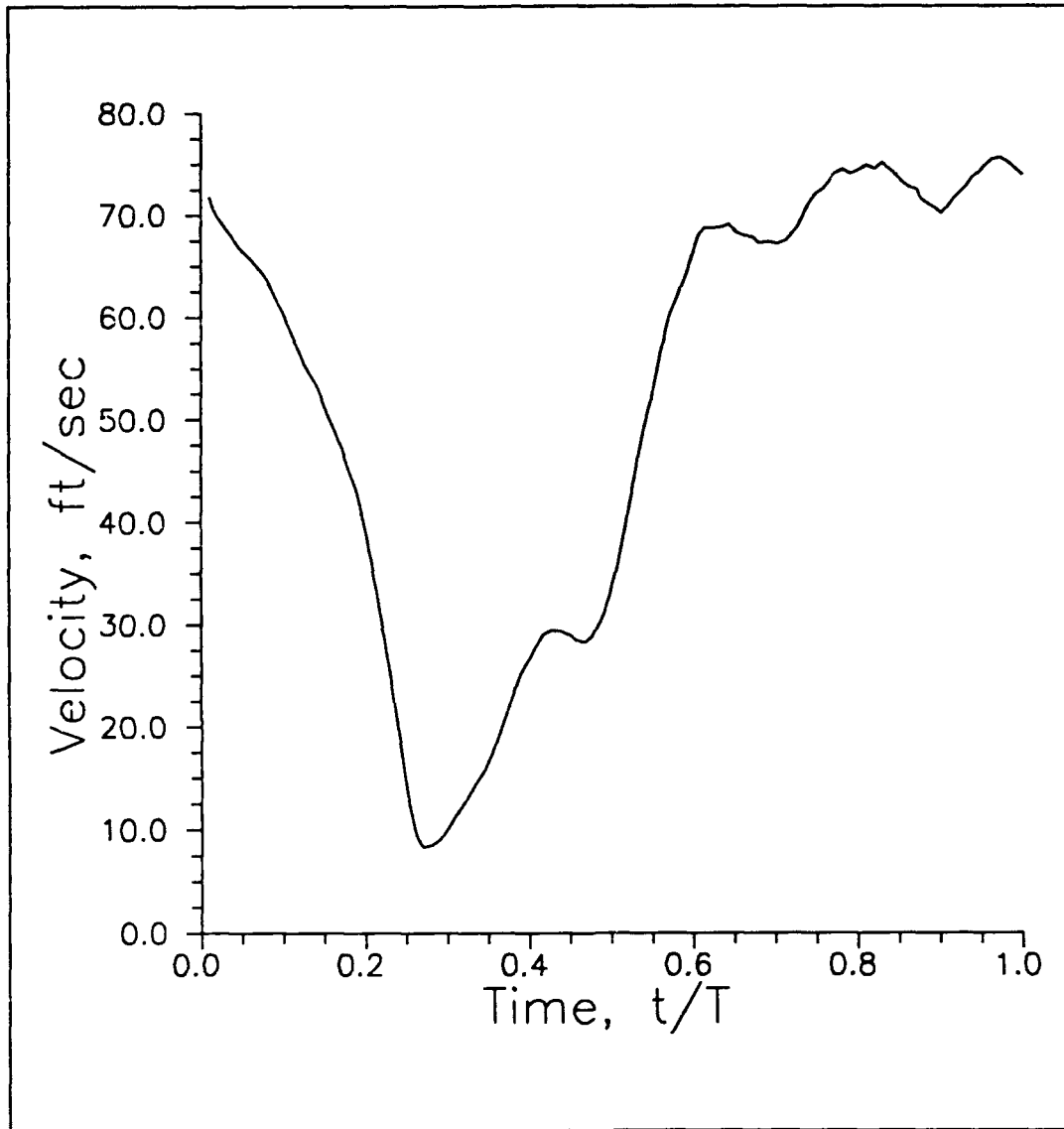
Figure 28. Velocity Profile, RUN110.

41
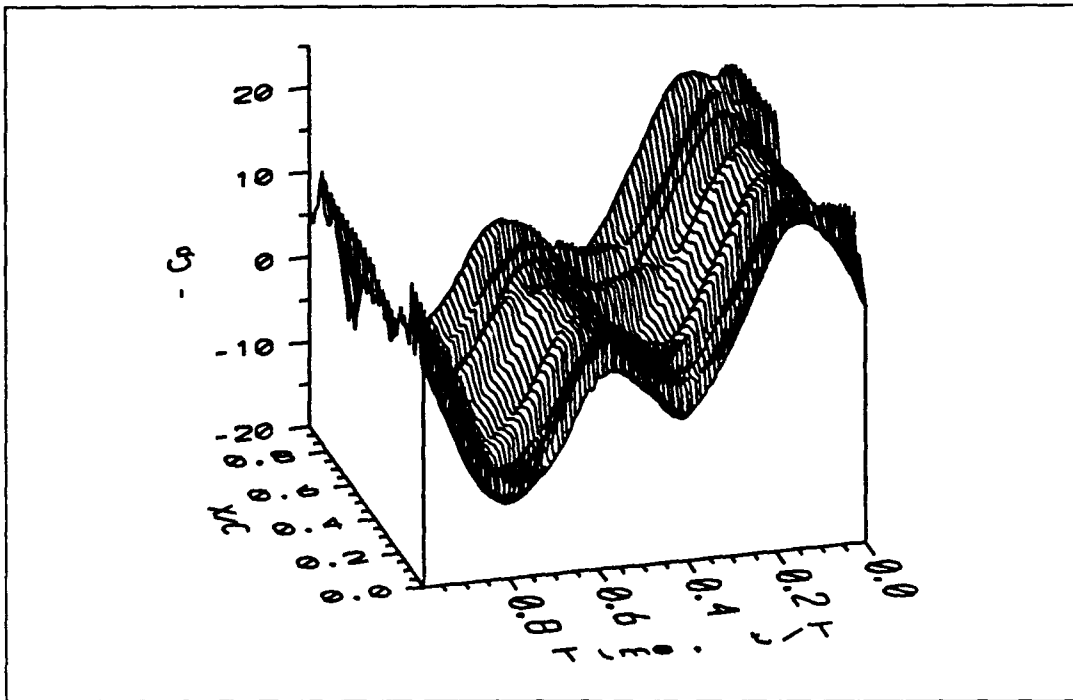
Figure 29. Upper Surface Pressure Coefficient at 15° Angle of Attack in Oscillating Flow.
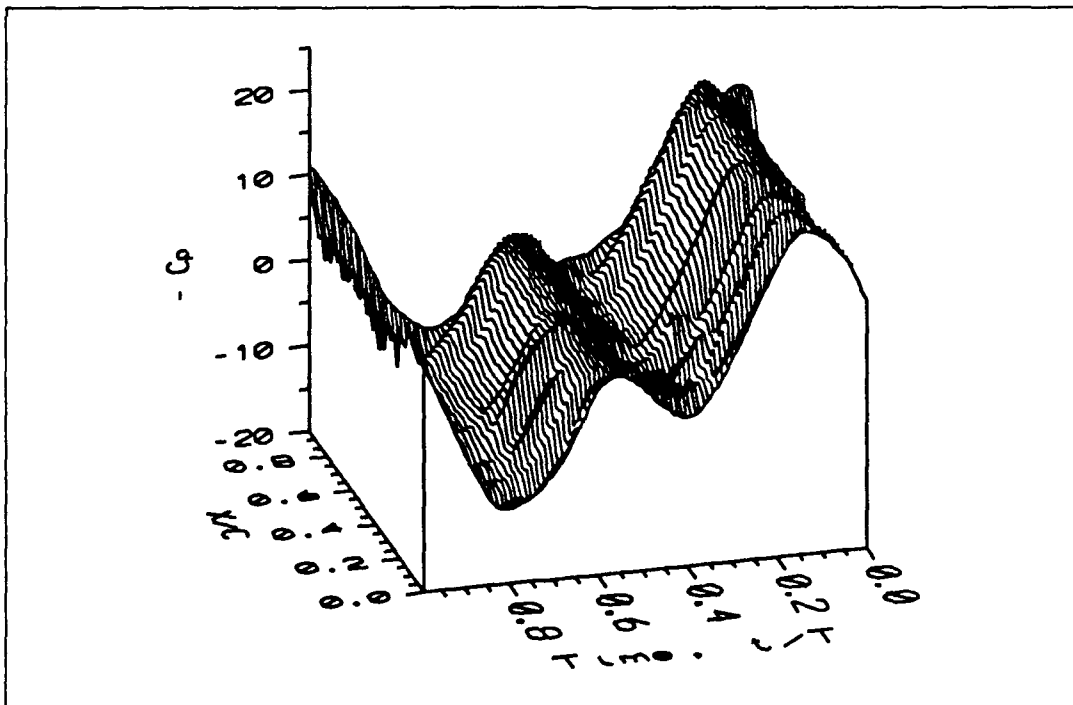
Figure 30. Lower Surface Pressure Coefficient at 15° Angle of Attack in Oscillating Flow.
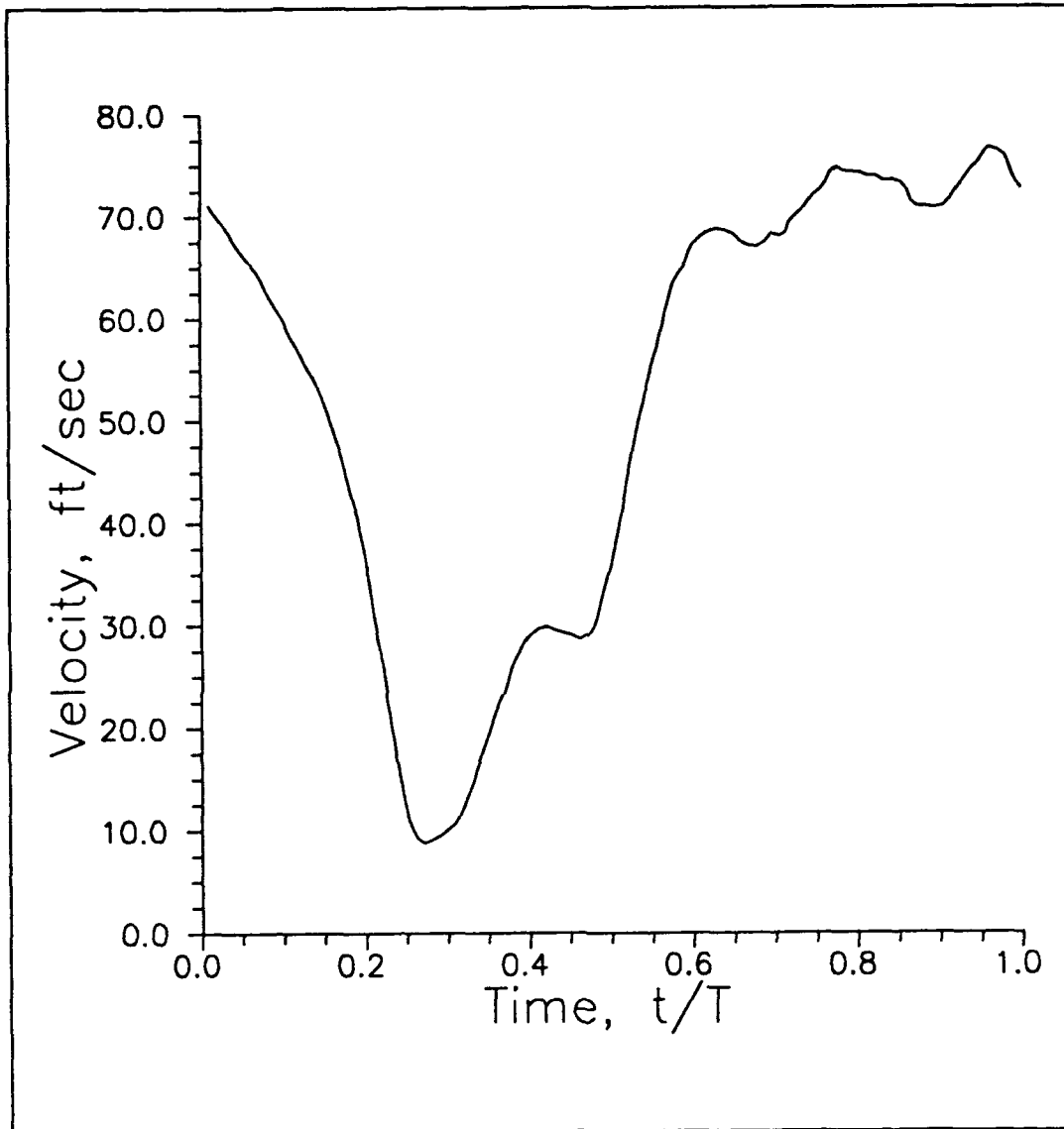
Figure 31. Velocity Profile, RUN115.

43
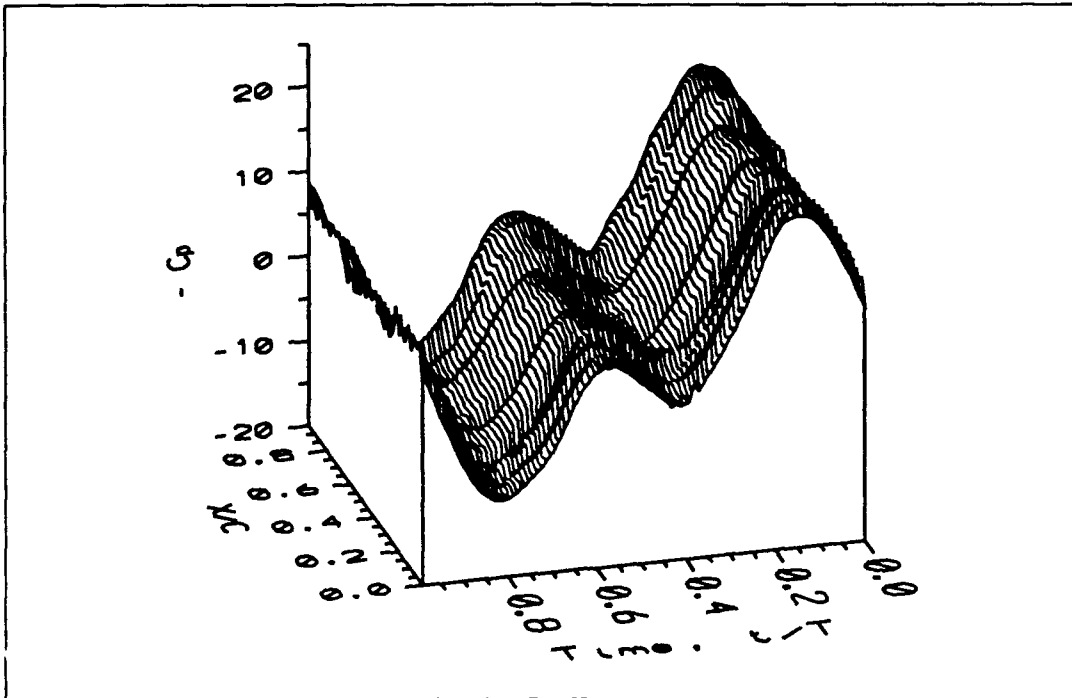
Figure 32. Upper Surface Presure Coefficient at 20° Angle of Attack in Oscillating Flow.
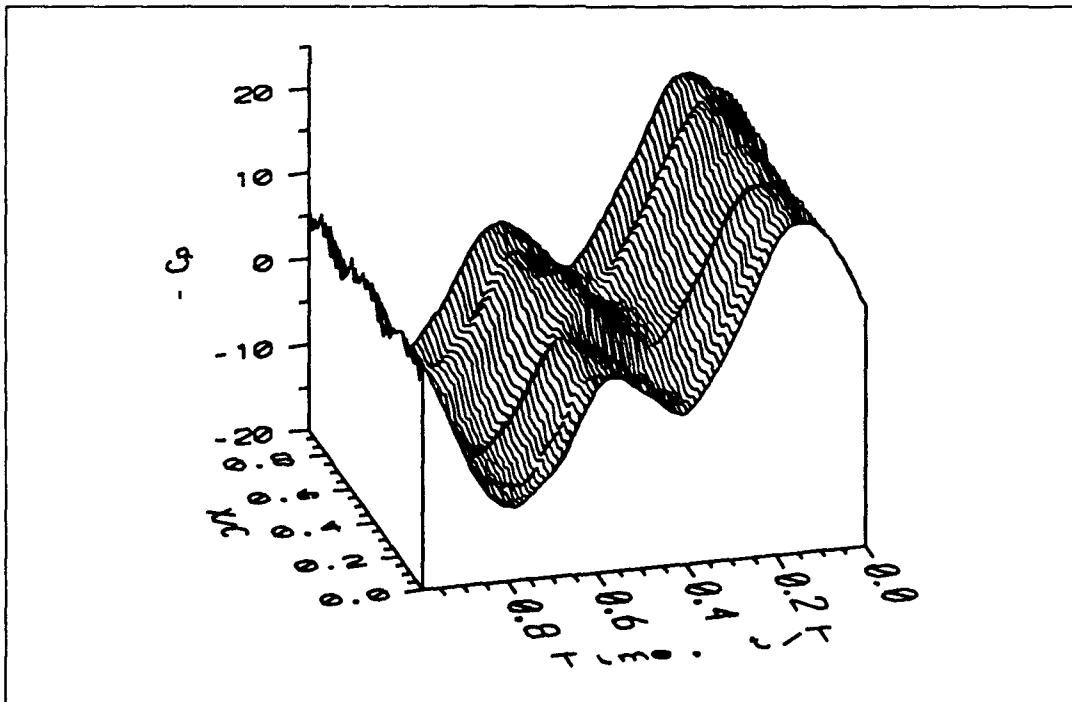
Figure 33. Lower Surface Pressure Coefficient at 20° Angle of Attack in Oscillating Flow.
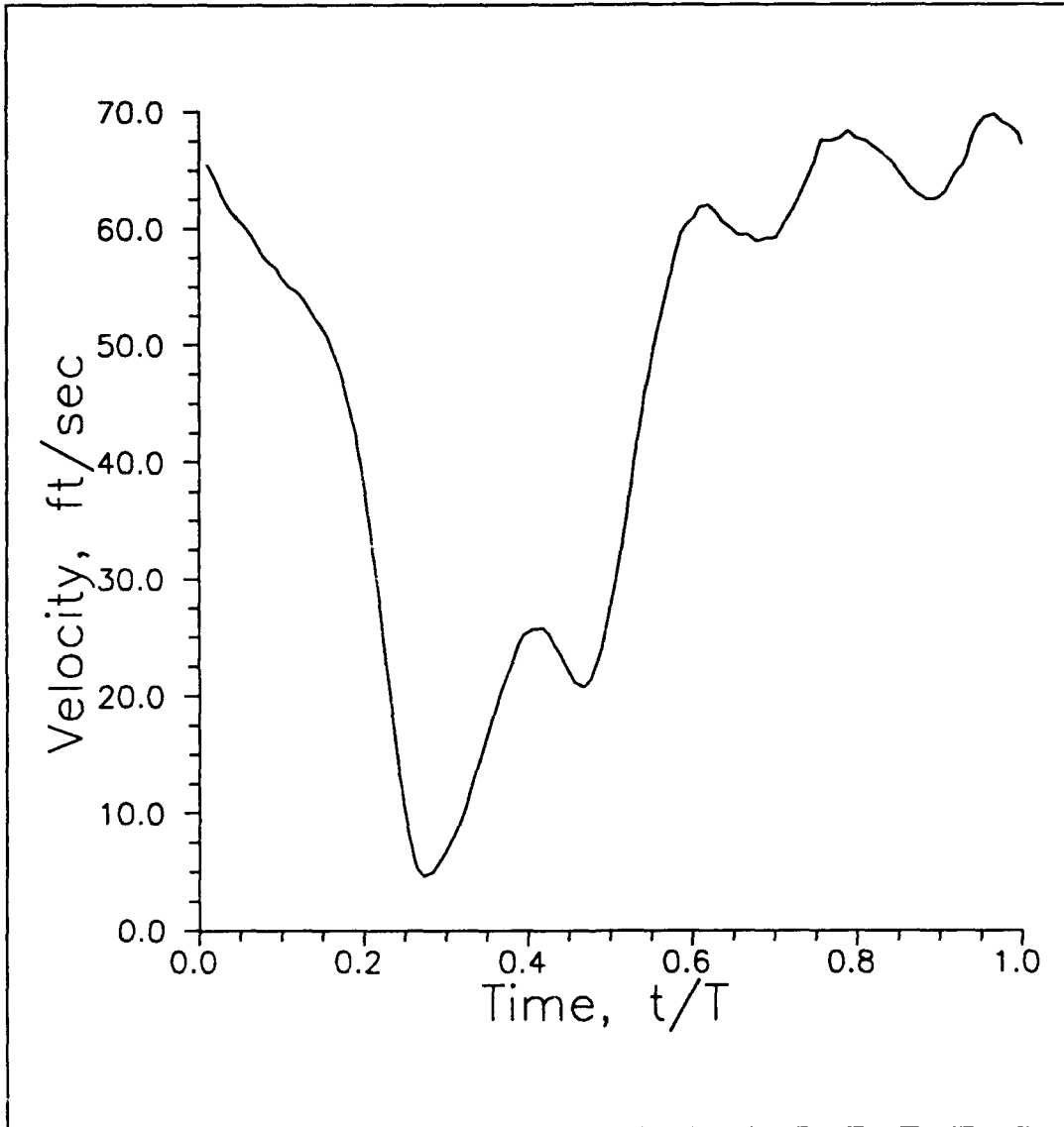
Figure 34. Velocity Profile, RUN120.
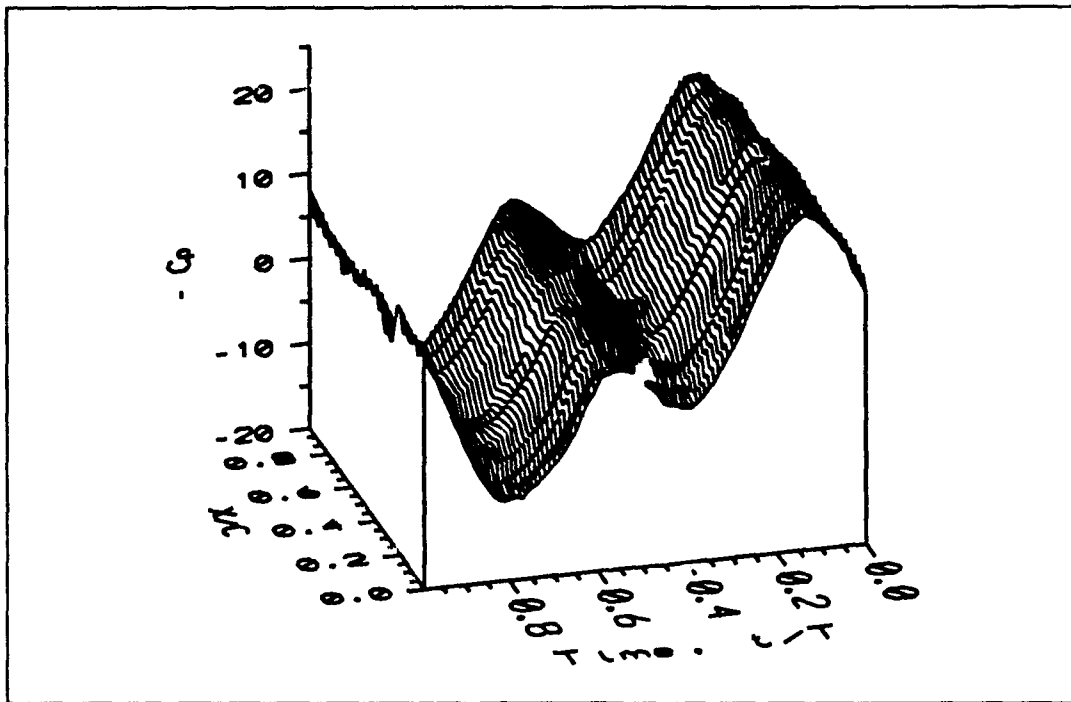
Figure 35. Upper Surface Pressure Coefficient at 25° Angle of Attack in Oscillating Flow.
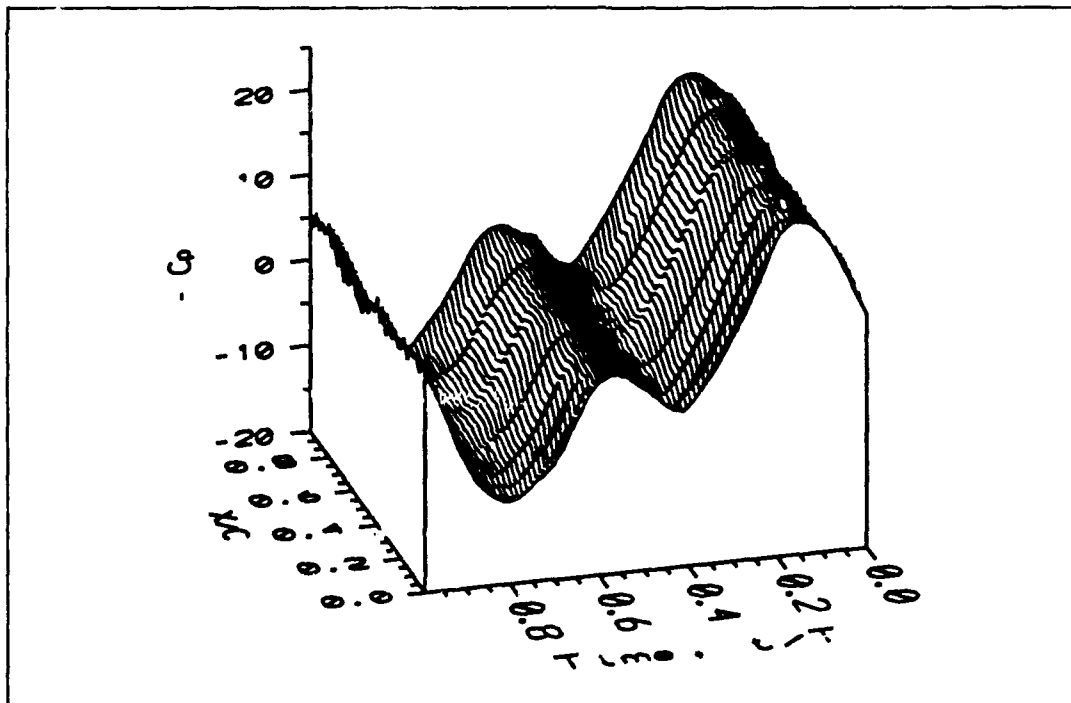
Figure 36. Lower Surface Pressure Coefficient at 25° Angle of Attack in Oscillating Flow.
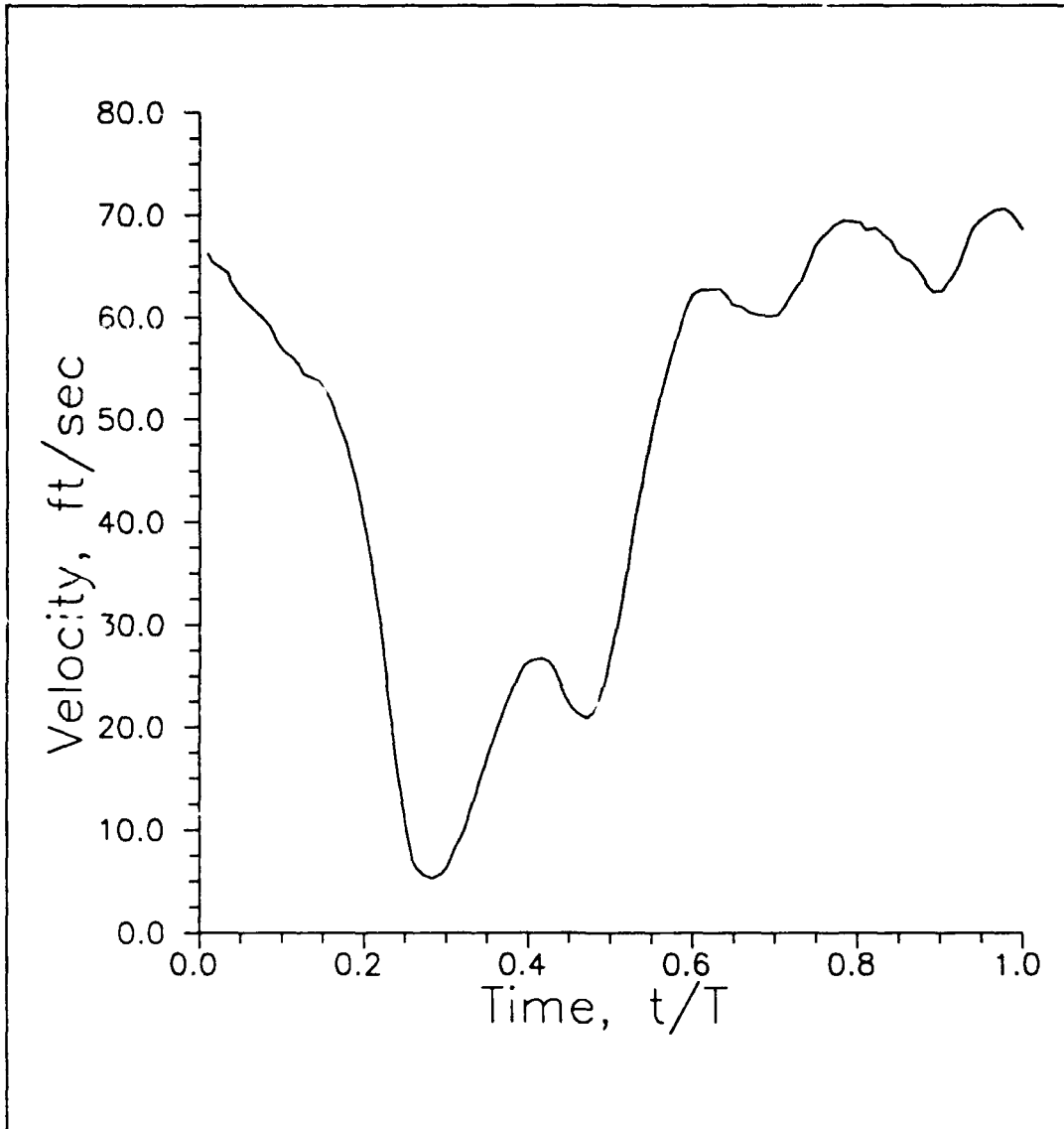
Figure 37. Velocity Profile, RUN125.
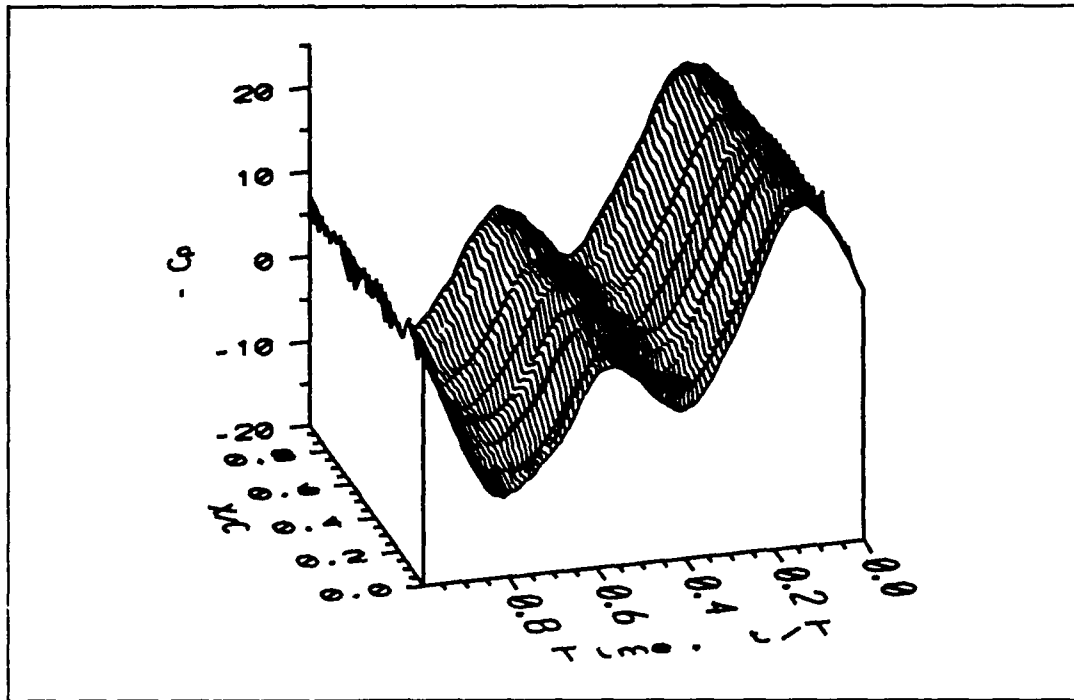
Figure 38. Upper Surface Pressure Coefficient at 30° Angle of Attack in Oscillating Flow.
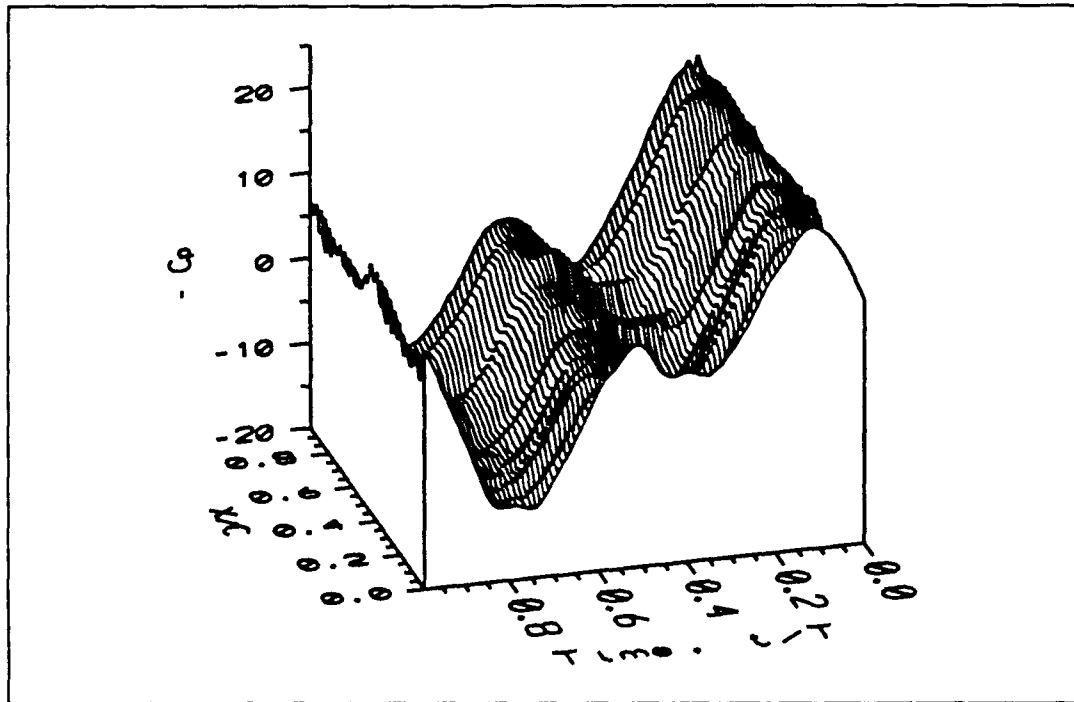
Figure 39. Lower Surface Pressure Coefficient at 30° Angle of Attack in Oscillating Flow.
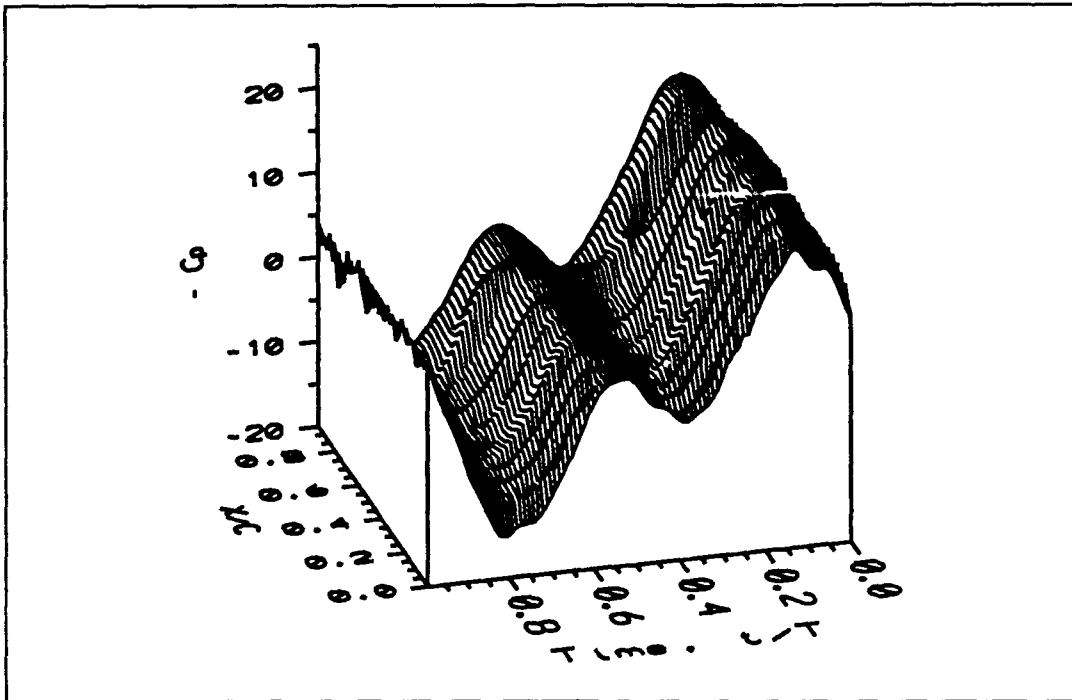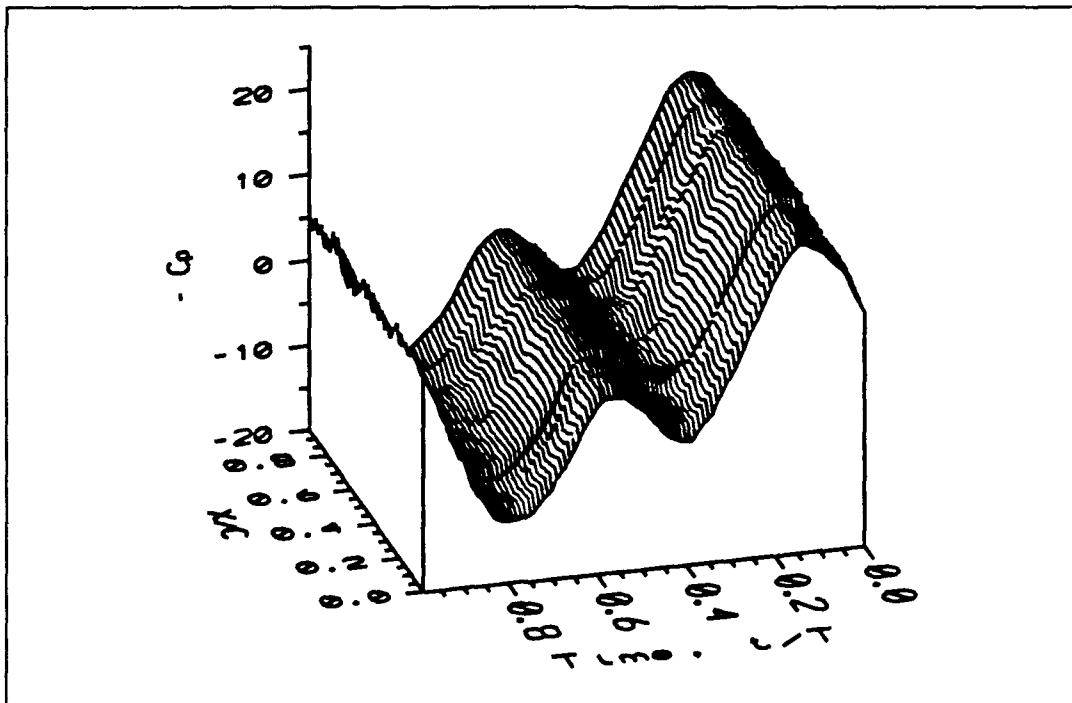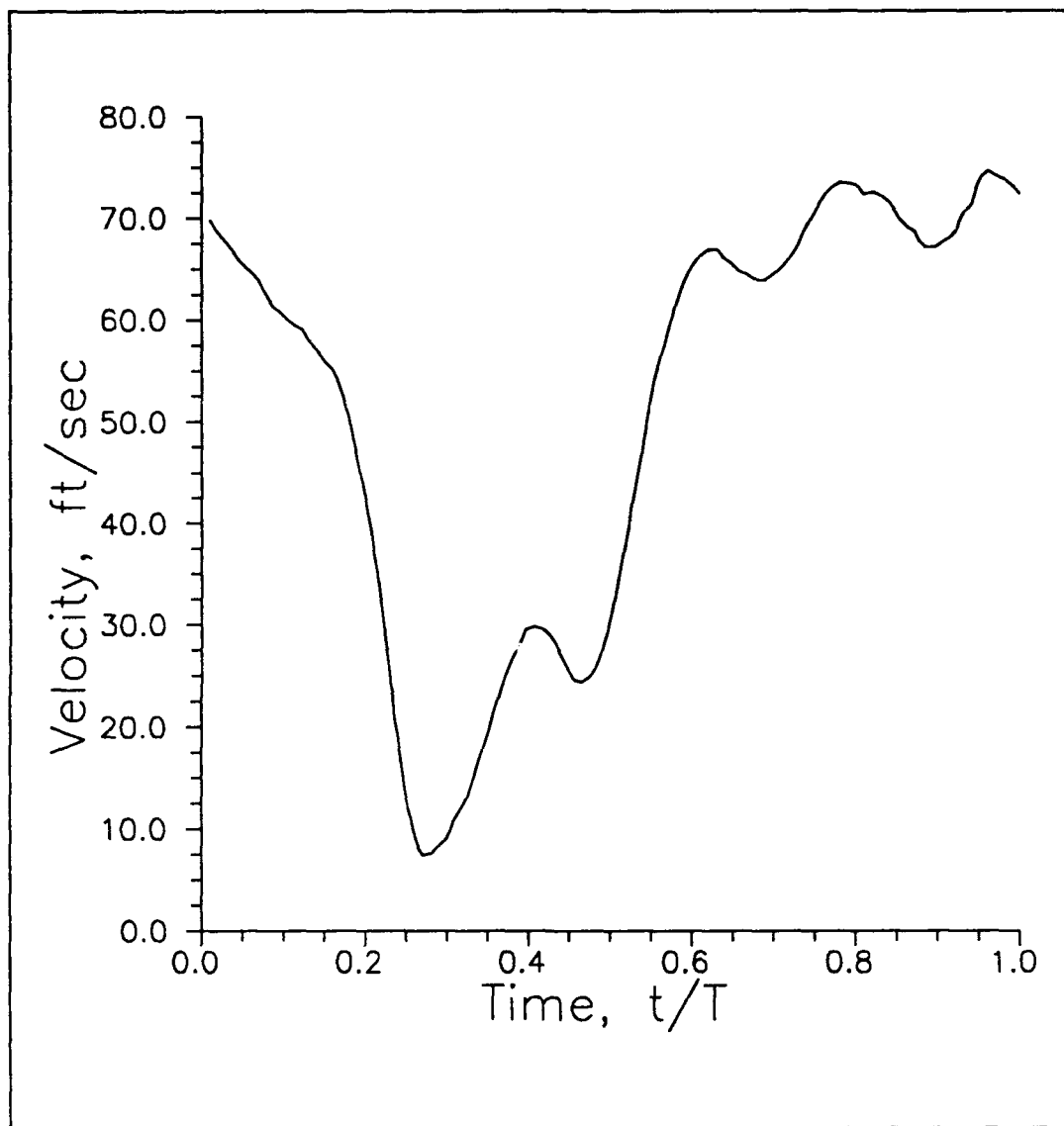
Figure 40. Velocity Profile, RUN130.
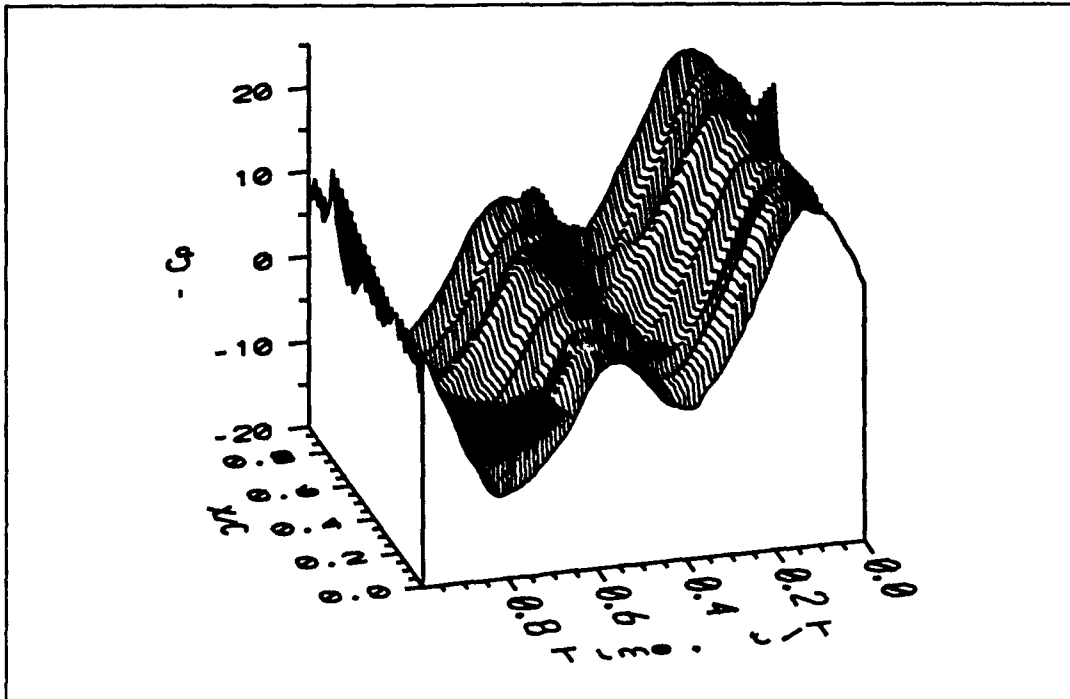
Figure 41. Upper Surface Pressure Coefficient at 35° Angle of Attack in Oscillating Flow.

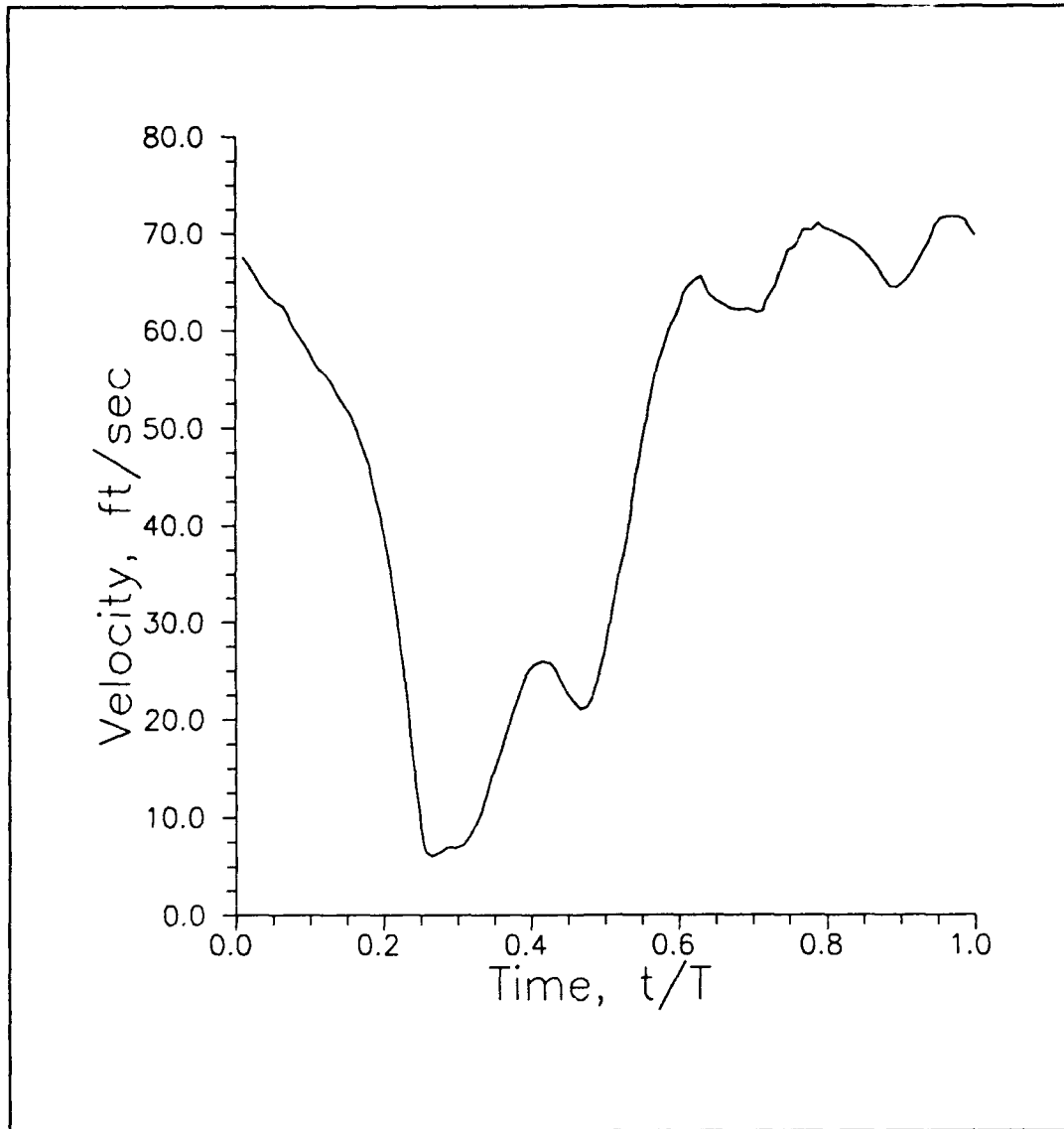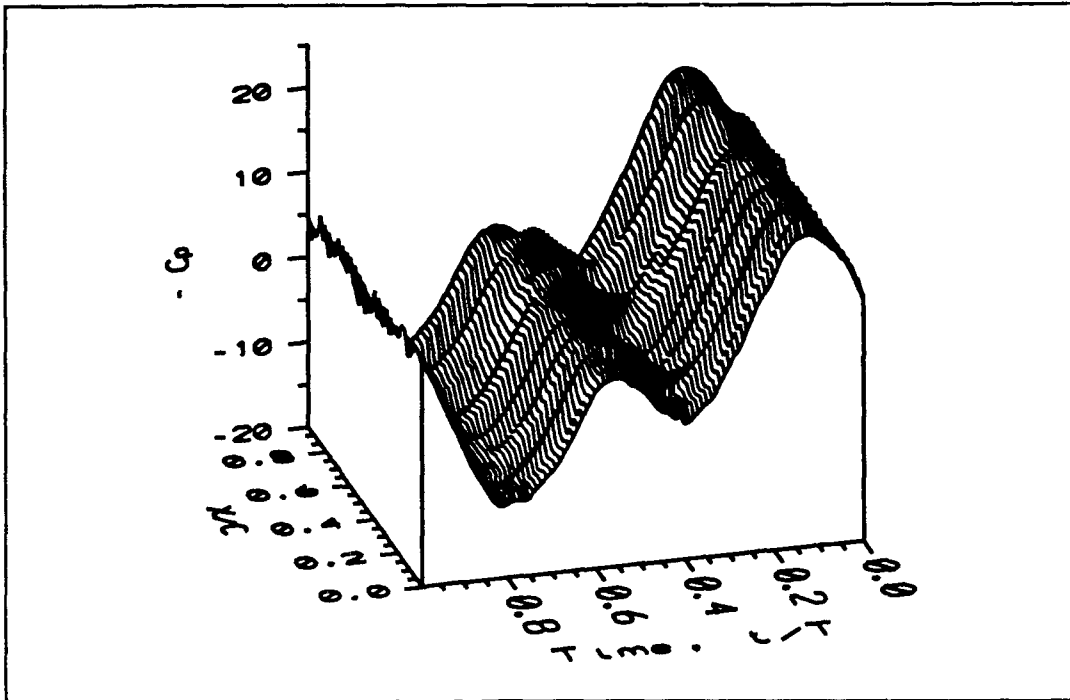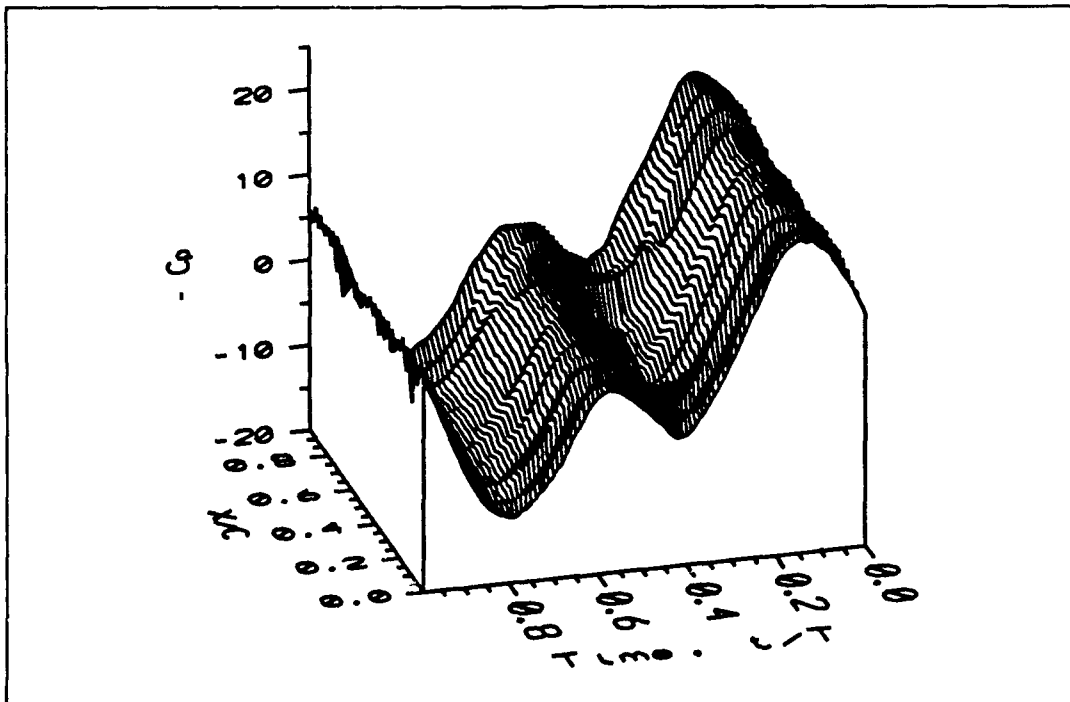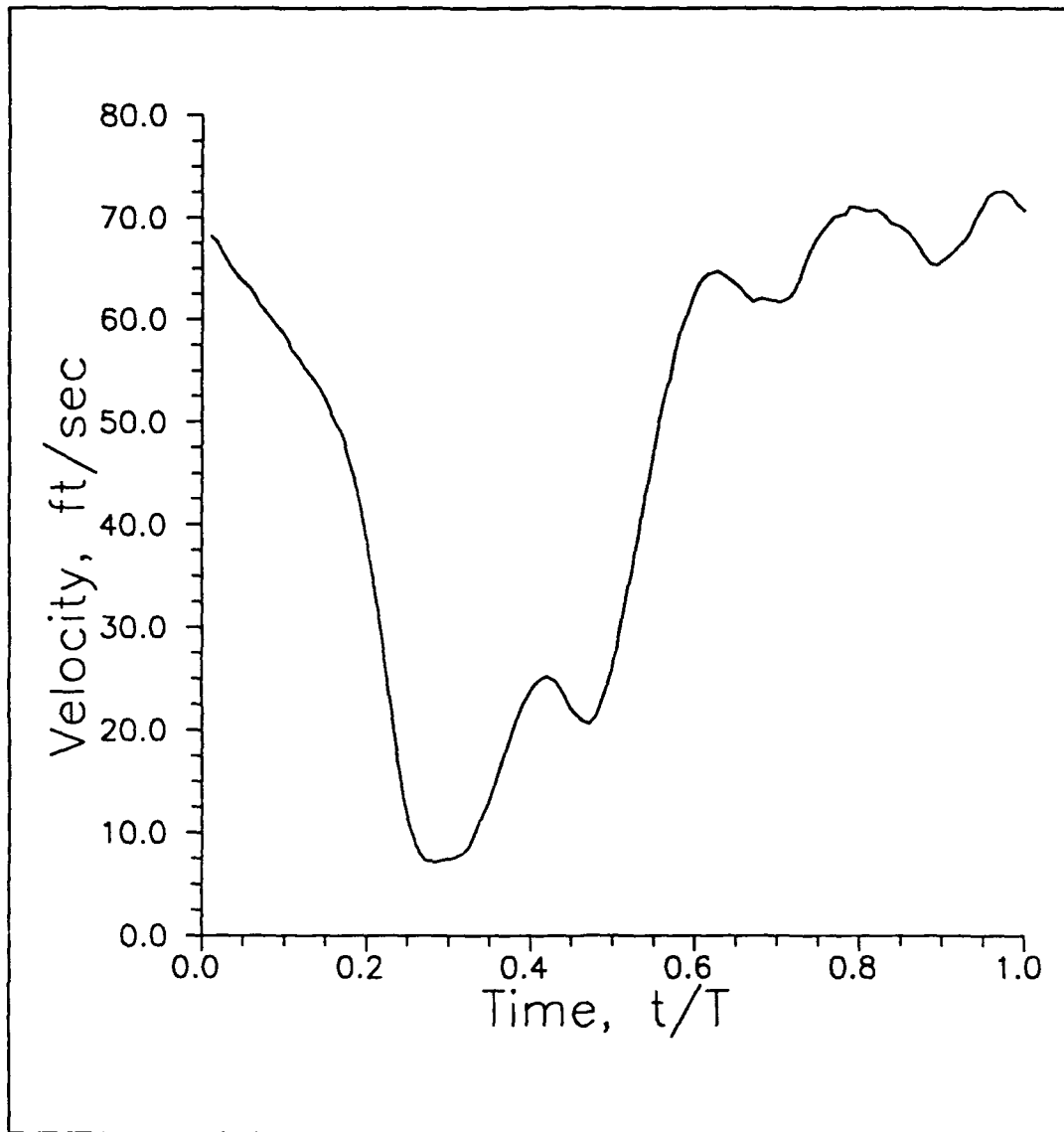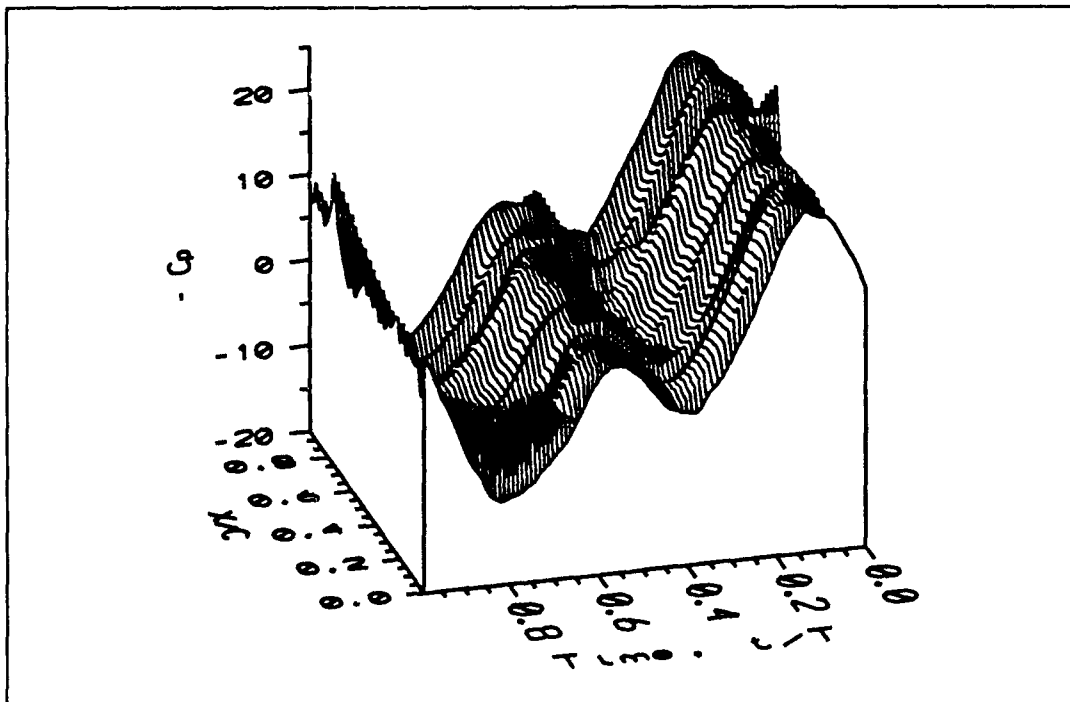**Figure 42.** Lower Surface Pressure Coefficient at 35° Angle of Attack in Oscillating Flow.
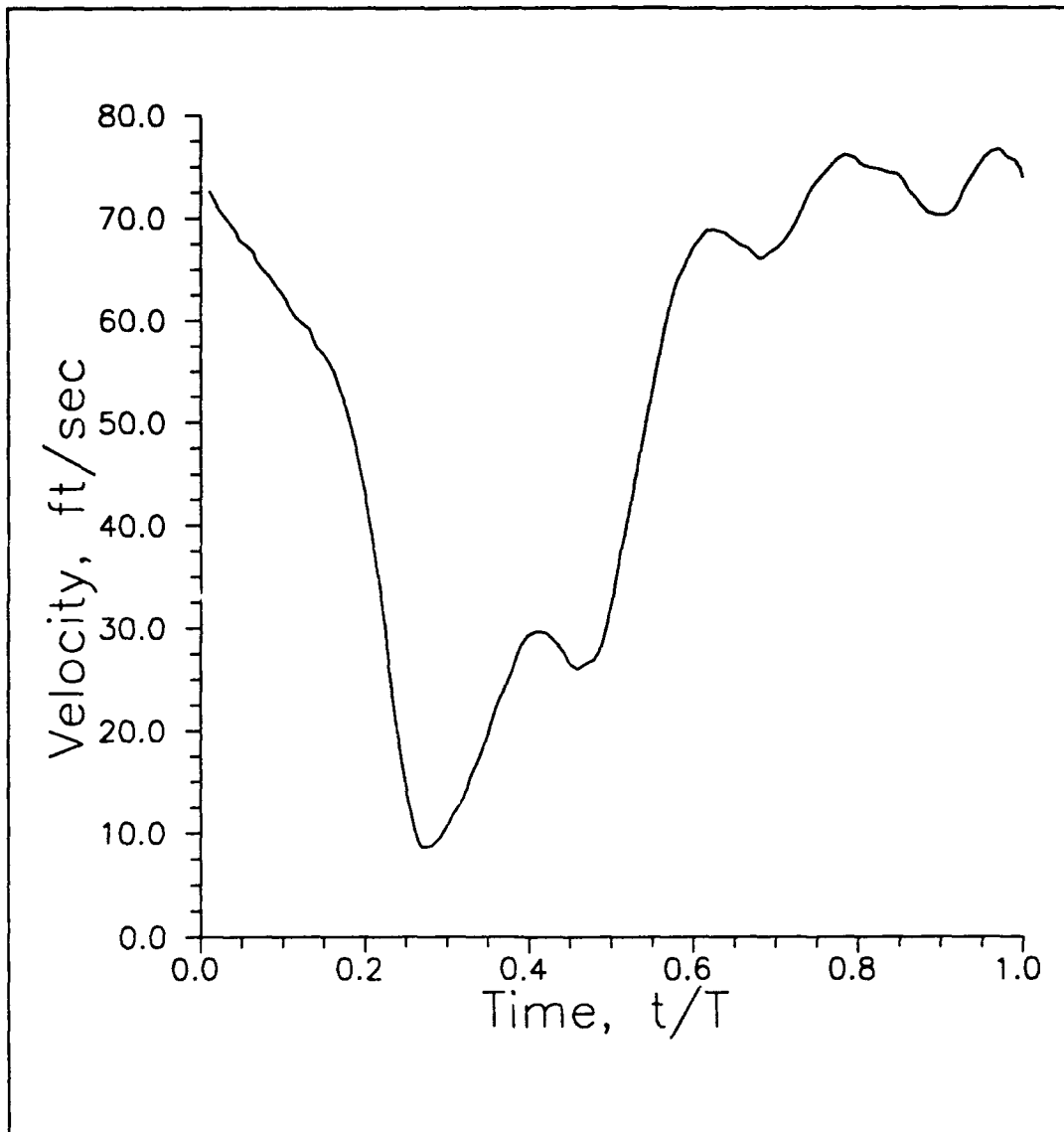
Figure 43. Velocity Profile, RUN135.

51

It is most common in the lower angles of attack, particularly from 5 to 15 degrees, and less so in the higher angles of attack, until its reappearance at 35 degrees. This is consistent with the known phenomenon of increased C_L at high angles of attack, which is thought to be caused by delayed onset of dynamic stall in oscillating flow.

The oscillating flow performance coefficients are plotted as a function of angle of attack and cycle time in Figures 44 - 49. These plots were also generated with Surfer, but with an initial grid size of 8 by 25, which used every fourth data point in the timewise direction. This grid was then spline smoothed, inserting two grid points in between each existing data point in both the angle-of-attack-wise and timewise directions. The lines in the first three surface plots are constant with respect to the cycle time, and the lines in the last three are constant with respect to angle of attack. The most obvious feature of all of the performance coefficient curves is their apparently random, unpredictable appearance. In spite of the closeness with which the coefficient of pressure adhered to the dominant freestream static pressure oscillations, there is no observable pattern that repeats in time at any constant angle of attack. Similarly, there is no evidence of steady or quasi-steady flow behavior. This observation can be verified by following any constant-time line and noting that it does not trace out a reasonable facsimile of a steady flow lift, drag or moment coefficient curve. This result is not surprising in light of the absence of any traditional chordwise pressure variation.

To a small degree, there is a timewise variation in the lift coefficient similar to that in the pressure coefficient plots. In general, the value of the coefficient rises and falls with respect to time in a manner similar to C_p, forming two ridges and two valleys. This

52
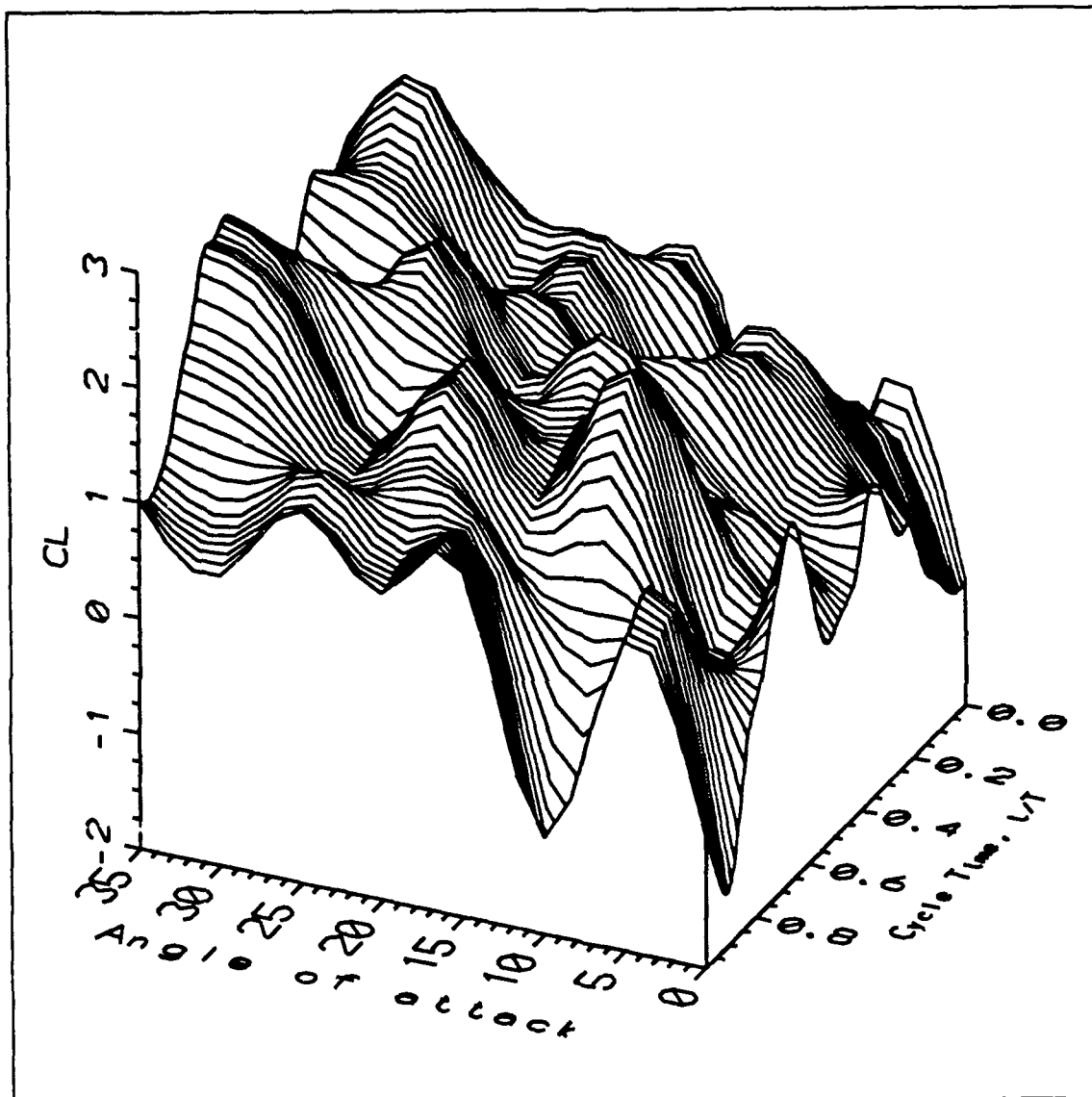
Figure 44. Lift Coefficient in Oscillating Flow (Constant Time Lines).
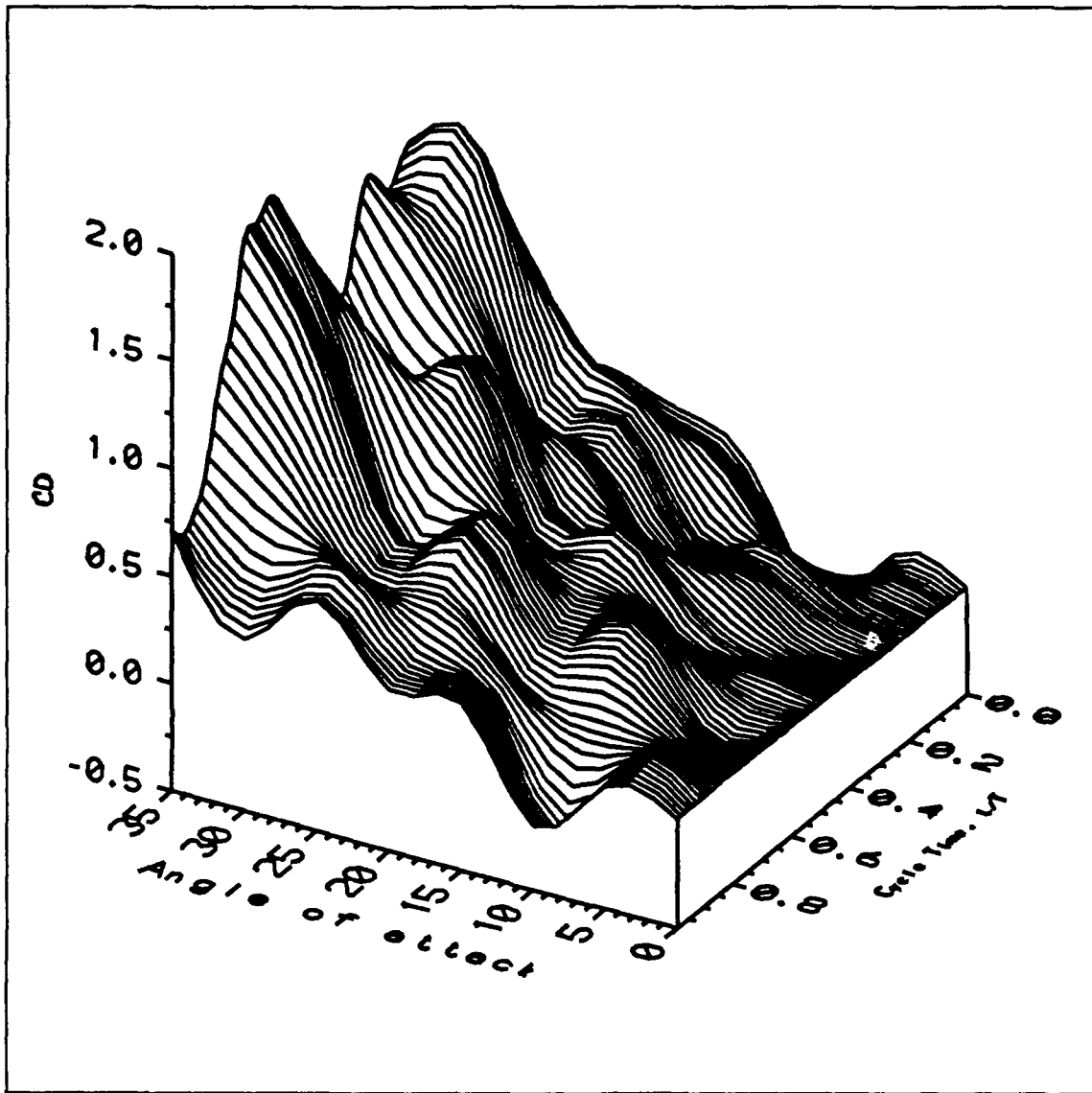
Figure 45. Drag Coefficient in Oscillating Flow (Constant Time Lines).
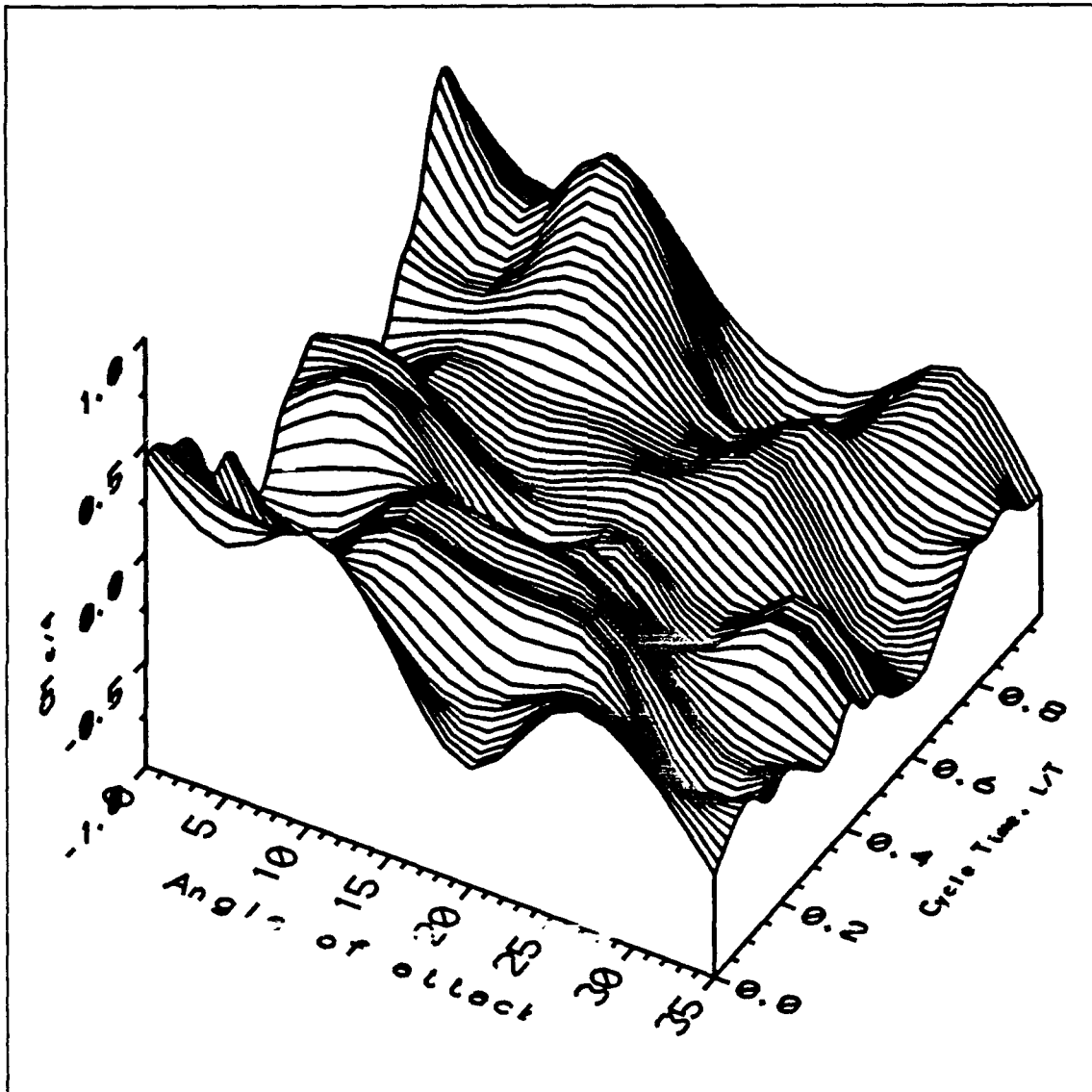
Figure 46. Quarter Chord Moment Coefficient in Oscillating Flow (Constant Time Lines).

55
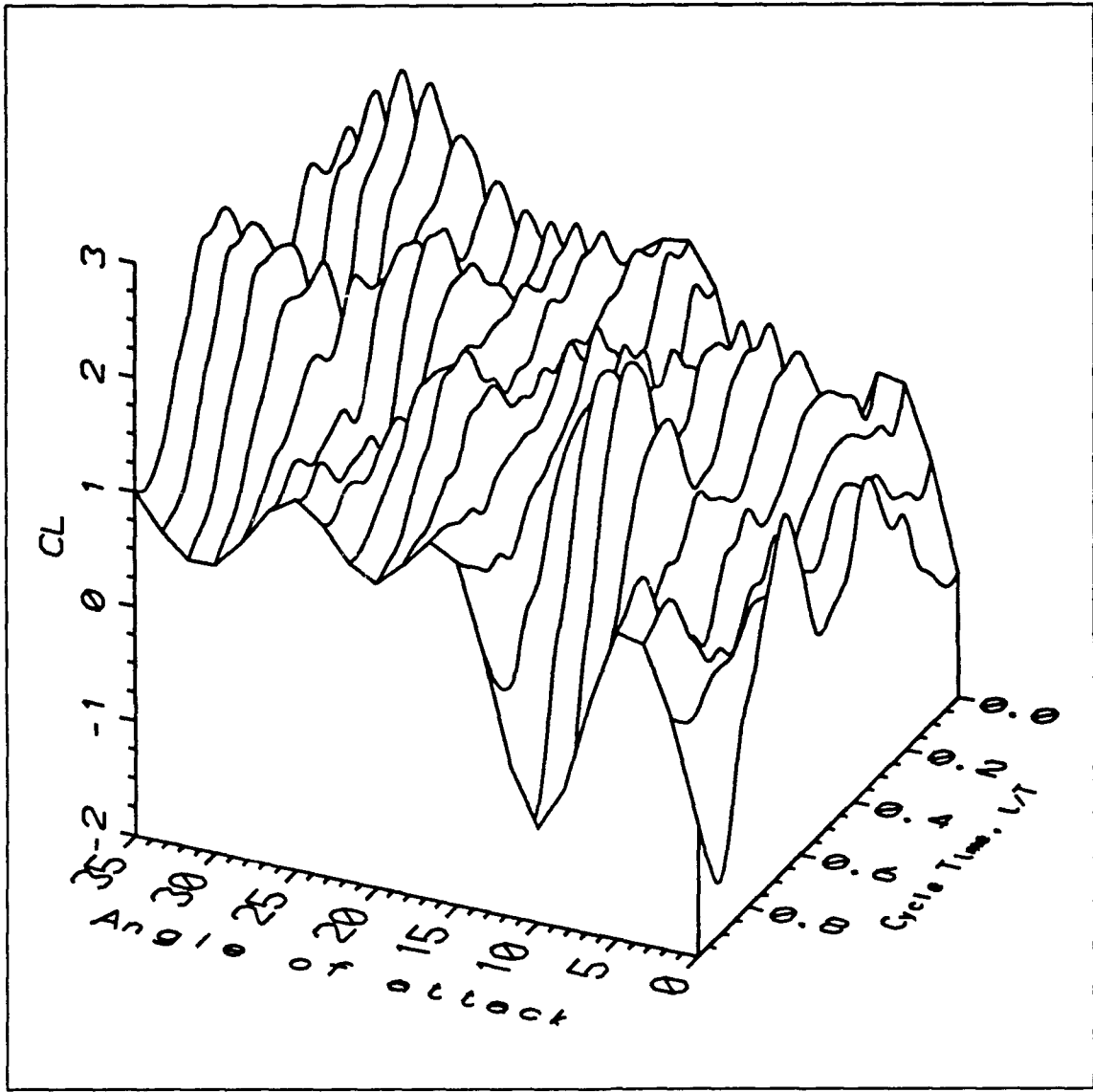
Figure 47. Lift Coefficient in Oscillating Flow (Constant AOA Lines).
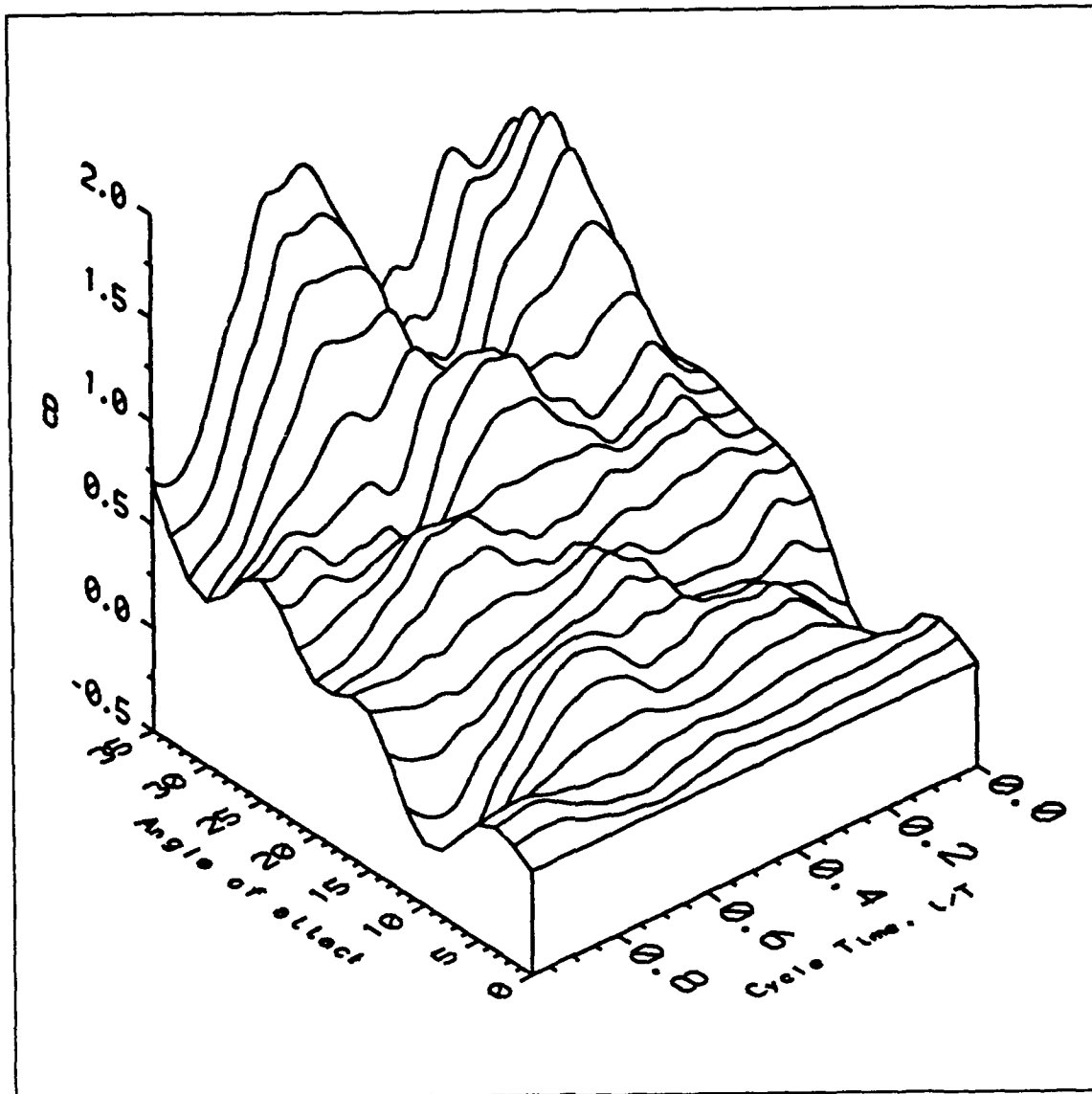
Figure 48. Drag Coefficient in Oscillating Flow (Constant AOA Lines).
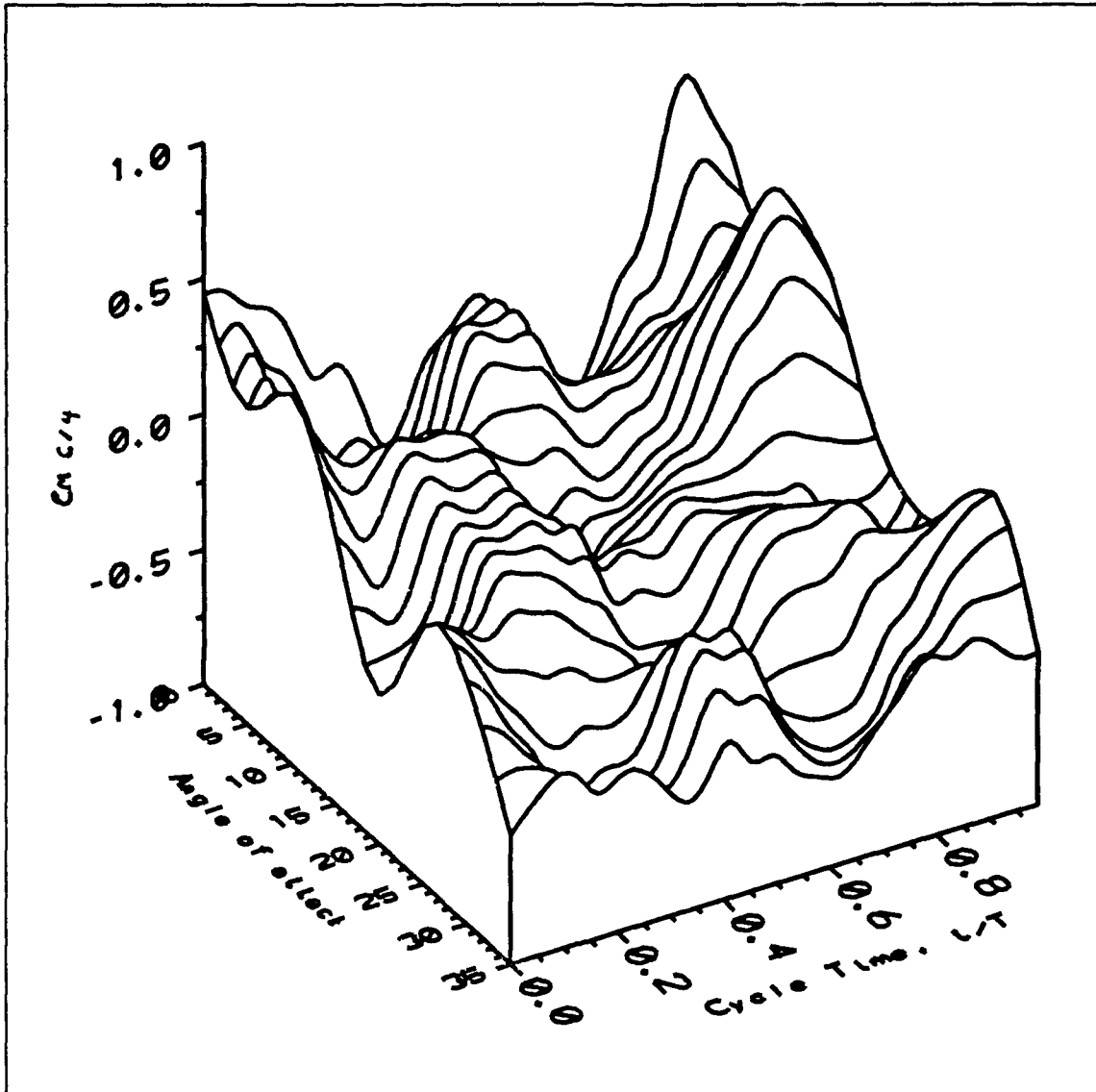
Figure 49. Quarter Chord Moment Coefficient in Oscillating Flow (Constant AOA Lines).

58

pattern is somewhat distorted throughout the region between 5 and 15 degrees angle of attack, which is consistent with the separations noted previously in the pressure coefficient plots. Additionally, there is a general trend of increasing lift coefficient from 0 to 10 degrees, followed by a lower, flat area at 15 to 20 degrees, and a subsequent increasing trend out to 35 degrees angle of attack.

It is apparent that the drag coefficient increases in an exponential fashion with increased angle of attack. Likewise, the moment about the quarter chord appears to become substantially more negative at higher angles of attack. The drag and moment coefficient curves are more difficult to interpret in general.

The mean values of the lift, drag and moment coefficients are presented as a function of angle of attack only in Figures 50 - 52. These graphs are also plotted such
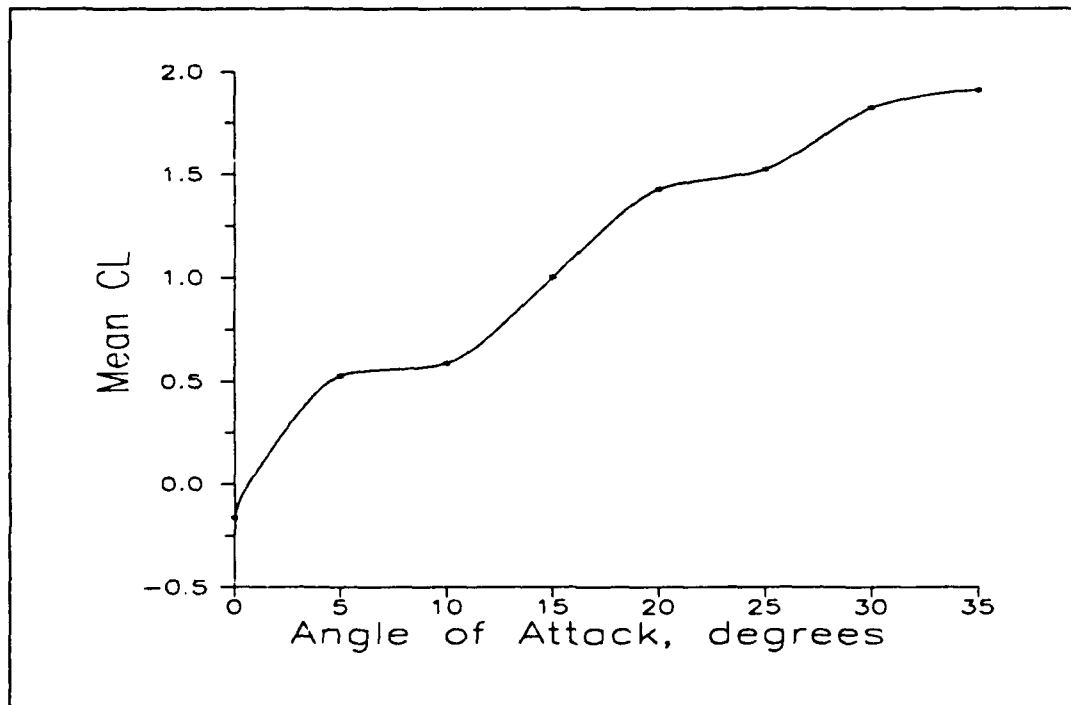
Figure 50. Mean Lift Coefficient in 10 Hz Oscillating Flow.

59
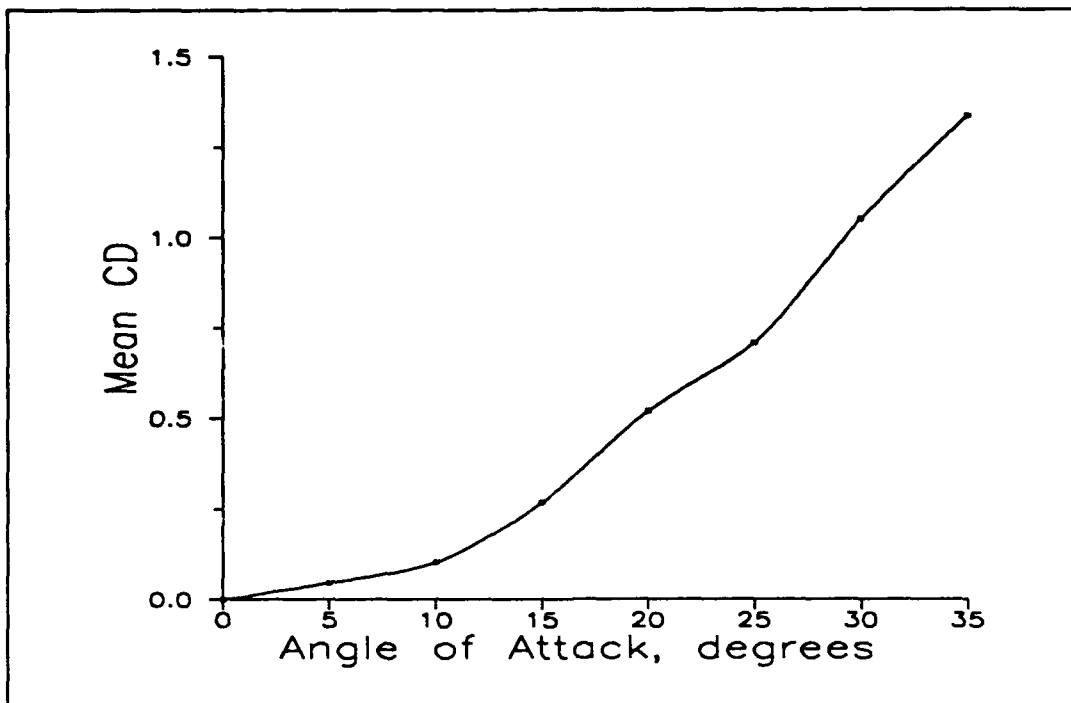
Figure 51. Mean Drag Coefficient in Oscillating Flow.
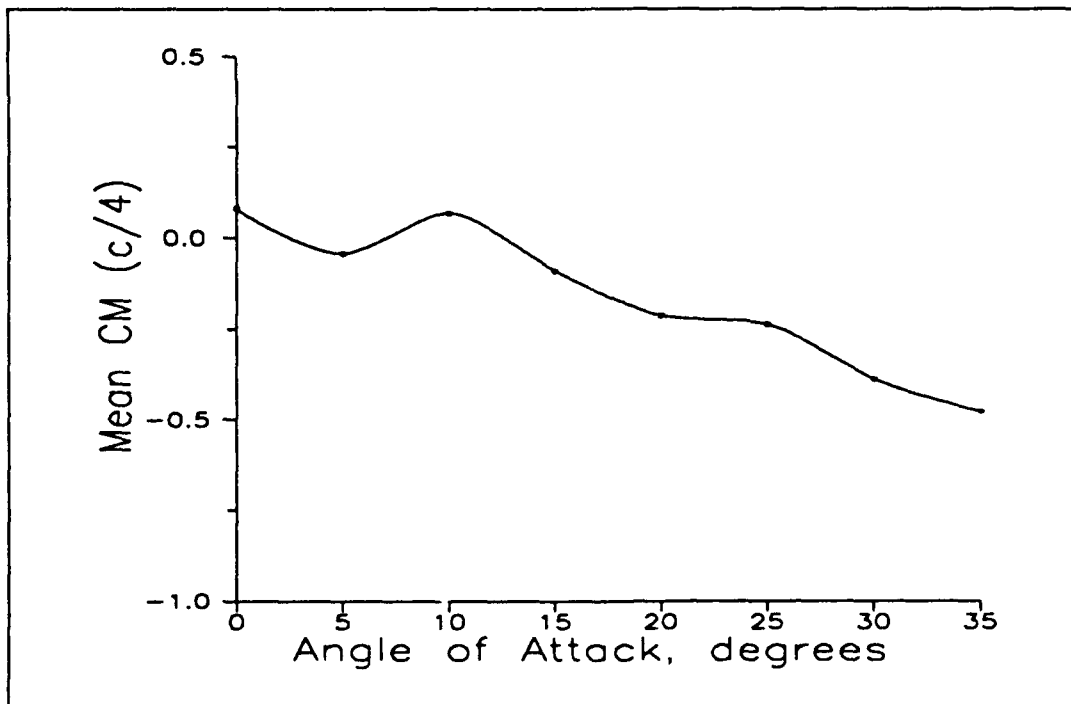
Figure 52. Mean Quarter Chord Moment Coefficient in Oscillating Flow.

that the original data points are displayed, each representing the arithmetic mean value of the respective coefficient at constant angle of attack, connected by a cubic spline curve. These performance coefficients exhibit the well known "average" behavior of a symmetrical airfoil in oscillating flow. In particular, the lift coefficient appears to increase at angles of attack in excess of the steady flow stall angle, even out to 35 degrees. The magnitude of the lift coefficient at lower angles of attack is slightly lower than the corresponding values in steady flow, however the peak value of 2.9 at 35 degrees is well in excess of the steady flow peak C_L of 0.78, found at 10 degrees angle of attack. The oscillating flow mean drag coefficient is increased at every angle of attack over the corresponding steady flow drag coefficient, however the general trend of the two plots is to increase with increasing angle of attack. Likewise, the oscillating flow mean moment coefficient curve is similar in shape to that of the steady flow moment coefficient, but with exaggerated values at the higher angles of attack.

VI. CONCLUSIONS

The steady flow pressure and performance coefficient results lend credence to the experimental procedures, data acquisition and data reduction in this work. The pressure coefficient curves show entirely predictable chordwise variation, which is well known to produce lift on an airfoil. Conversely, the oscillating flow pressure coefficients exhibit no such traditional chordwise variation, and in fact indicate that the instantaneous pressure distribution on a symmetrical airfoil in an oscillating flow is dominated by the freestream static pressure. The noted exceptions to this observation are the occasional flow separations at the lower angles of attack, analagous to the onset of separation in steady flow.

The oscillating flow results also indicate that the aerodynamic performance of a symmetrical airfoil in an oscillating flow is in no way a quasi-steady phenomenon. This is evidenced by the fact that the surface plots of the lift, drag and moment coefficients have a very random appearance, in spite of the closeness with which the pressure coefficient follows the freestream static pressure. It is not intuitively obvious, based on the shape of the velocity profile and the pressure coefficient surface plots, that the performance coefficients would behave as they do.

This randomness notwithstanding, the mean values of the same performance coefficients attest to the known behavior of increased lift coefficients in oscillating flow.

As previously noted, the general trends of the lift, drag and moment coefficients in oscillating flow resemble their steady flow counterparts, albeit with exaggerated values. One must conclude, however, that the instantaneous values of these coefficients cannot be assumed to be a continuum of quasi-steady states.

APPENDIX A

The following is the data acquisition program, TKDATA.BAS, which is written in

Quick BASIC.

```
'PROGRAM TKDATA  This program contains SUBROUTINES to control the SCANIVALVE,
        'setup the SRA 1200 RECORDER boards, take dat, download that
        'data into arrays VRAW, representing the raw velocity data,
        'PRAW, representing the raw pressure data, process that data,
        'and write the output to file and/or plot it on the screen.
        'These SUBROUTINES and their respective functions are:

            'HOME - Returns SCANIVALVE to HOME
            'HEADER - Inputs data for a file header
            'PORTNUM - Returns SCANIVALVE number (PORTNO)
            'INIT - Initializes recorder boards
            'RECORD - Records data
            'getstartadd - Returns starting address of record
            'getdat - Returns data samples
            'PLOT - Plots data on screen
            'LOAD - Loads two arrays, VRAW and PRAW, (in pseudo-
                'complex format)
            'MEAN - Computes the mean of the arrays PRAW and VRAW,
                'subtracts the means and scales the values
            'FFTPRAW - Replaces PRAW with the FFT of PRAW
            'IFFTPRAW - Replaces PRAW with the IFFT of PRAW
            'PORT - Advances the SCANIVALVE port
            'RIRTHETA - Converts an array from real-imaginary pairs
```

```
                       'to r-theta pairs
         'RTHETARI - Converts an array from r-theta pairs to
                'real-imaginary pairs
         'DIVIDE - Replaces PRAW with PRAW/XFER
         'XFERINPUT - Inputs the transfer function array XFER
                'from an external file XFER.FCN
         'NEGALPHA - Halts data collection to allow resetting
                'angle of attack to MINUS ALPHA
         'ARRAY - Loads mean values of pressure and velocity
                'into vectors PMEAN and VMEAN and pressure
                'and velocity data into arrays P and V
         'DISK - Outputs pressure, velocity and header data
                'to a file called DATA.XXX where XXX is the
                'Run Number.

VSCALE = 125!

PSCALE = 144!

OPTION BASE 1                'Sets lower bound of subscripts = 1

DIM physicaladdress(2)       'Physical adresses of the recorders

DIM inputscale(2)            'Full scale input voltage for each
                             'channel. May be 1, 2, 5, or 10 volts

DIM starts(2)                'Array containing starting addresses

DIM VRAW(2048)               'Array containing raw velocity data

DIM PRAW(2048)               'Array containing raw pressure data

DIM XFER(2048)               'Array containing the transfer fncn.

DIM PMEAN(46)                'Vector containing mean pressure

DIM VMEAN(46)                'Vector containing mean velocity

DIM P(46, 100)               'Array containing pressure data

DIM V(46, 100)               'Array containing velocity data

GOSUB XFERINPUT                 'Inputs transfer function XFER

GOSUB HEADER                    'Calls for header information

GOSUB HOME                      'Sets SCANIVALVE to PORT 0

GOSUB PORT                      'Advances SCANIVALVE to PORT 1

GOSUB INIT                   'Use once before using recorders
'/////////////////////////////////////////////////////////////////
'RECORDER CONTROL PARAMETERS --- Set for specific application --
```

'These parameters are:

'   rate, the sample rate in mico-seconds, from 1 to 32767

'   samples, the number of samples per record, from 1 to memmax

'   psamples, the number of samples that preceed the trigger from

'   1 to memmax, but less than samples.

'   mode$, trigger slope control, either "PLUS" or "MINUS"

'   setpoint, for trigger threshold, range is -100 to 100 %

'

'   optional:

'   inputscale(n) controls programmable gain for channel n.

'   Valid with values of 1, 2, 5 and 10; n may be either 1 or 2.

'---

```
        SAMPLES = 1024              'Number of samples per record, (must
                                    'be of the form 2**n for processing)
        psamples = 1000             'Number of pretrigger samples
        setpoint = 10               'Trigger setpoint in % of full scale
        mode$ = "PLUS"              'Trigger mode$, (either "PLUS"
                                    'or "MINUS")
        rate = 1000                 'Sample rate in u-seconds per point
        inputscale(1) = 5           'full scale for channel 1 is 5 volts
        inputscale(2) = 5           'full scale for channel 2 is 5 volts
```

'---

'//

'Begin first data acquisiti n loop through top 23 ports:

```
FOR J = 1 TO 20000              'Delays data acquisition to allow
NEXT J                          'SCANIVALVE to settle
FOR P = 1 TO 2.
        GOSUB PORTNUM               'Reads current port number
        GOSUB RECORD                'Records data for one port
```

'===

```
        'GOSUB PLOT                  'Plots recorded data for debugging
        'END                         'Test point for debugging
```

'===

```
        GOSUB LOAD                  'Loads data into arrays VRAW & PRAW
                                    'of length 2048 in pseudo complex
```

66

```
                        'form
'===========================================================
        'END             'Test point for debugging
'===========================================================
    GOSUB PORT                 'Advances SCANIVALVE to next port
    GOSUB MEAN                  'Computes the mean of each array,
                        'subtracts the means (PBAR and VBAR),
                        'and scales all values with PSCALE
                        'and VSCALE so that output is in psf
                        'and ft/sec. Arrays remain named PRAW
                        'and VRAW
'===========================================================
        'END             'Test point for debugging
'===========================================================
    GOSUB FFTPRAW              'Replaces PRAW with its FFT
    GOSUB RIRTHETA             'Transforms PRAW in pseudo-complex
                        'form
    GOSUB DIVIDE              'Divides FFT of PRAW by XFER
    GOSUB RTHETARI            'Transforms PRAW back r-i pairs
    GOSUB IFFTRAW             'Performs inverse FFT on PRAW
'===========================================================
        'END             'Test point for debugging
'===========================================================
    GOSUB ARRAY               'Loads PBAR and VBAR into vectors
                        'and PRAW and VRAW into arrays
'===========================================================
        'END             'Test point for debugging
'===========================================================
NEXT P
'Next change angle of attack:
GOSUB NEGALPHA               'Stops execution to change ALPHA
'Then we continue with data collection for the bottom 23 ports:
GOSUB HOME                  'Sends SCANIVALVE back to zero
GOSUB PORT                  'Advances SCANIVALVE to first port
FOR J = 1 TO 20000          'Delays data acquisition to allow
```

67

```
NEXT J                      'SCANIVALVE to settle
FOR P = 1 TO 23
        GOSUB PORTNUM               'Reads current port number
        PORTNO = PORTNO + 23        'Adds 23 to PORTNO for no. 24 thru 46
        GOSUB RECORD                'Records data for one port
        GOSUB LOAD                  'Loads data into arrays VRAW & PRAW
                                    'of length 2048 in pseudo complex
                                    'form
        GOSUB PORT                  'Advances Scanivalve to next port
        GOSUB MEAN                  'Computes the mean of each array,
                                    'subtracts the means (PBAR and VBAR),
                                    'and scales all values with PSCALE
                                    'and VSCALE so that output is in psf
                                    'and ft/sec. Arrays remain named PRAW
                                    'and VRAW
        GOSUB FFTPRAW               'Replaces PRAW with its FFT
        GOSUB RIRTHETA              'Transforms PRAW in pseudo-complex
                                    'form
        GOSUB DIVIDE                'Divides FFT of PRAW by XFER
        GOSUB RTHETARI              'Transforms PRAW back r-i pairs
        GOSUB IFFTRAW               'Performs inverse FFT on PRAW
        GOSUB ARRAY                 'Loads PBAR and VBAR into vectors
                                    'and PRAW and VRAW into arrays
NEXT P
GOSUB DISK:                         'Writes all header data, the vectors
                                    'PMEAN and VMEAN and the arrays P and
                                    'V to disk in a file RUNXXX.DAT
END                                 'End of application code......
'=================================================================================
'    SUBROUTINES WHICH MAY BE CALLED AS REQUIRED BY PROGRAM TKDATA
INIT:  'Use this routine once at startup for variable and hardware init.
        k = 1024                    'k is 2 ** 10
        SAMPLES = 2000              'init # of samples per record
        psamples = INT(SAMPLES / 4)     'init presamples at .25 total record
        setpoint = 50               'trigger setpoint in % of full scale
```

68

```
mode$ = "PLUS"              'trigger mode$ is either "PLUS" or "MINUS" or -

rate = 4                    'default sample rate in us.

memmax = (64 * k) - 1       'recorder memory size

tbc = &H205                 'time base control register address

cwr = &H21E                 '8253 control write register

ctr0 = &H206               '8253 counter #1

ctr1 = &H20E               '8263 counter #2

ctr2 = &H216               '8253 counter #3

clkie = 1                   'for hardware hook not being used

FOR J = 1 TO 2

'set up list for physical address starting at &h200

physicaladdress(J) = &H200 + (J * 32) - 32

'Read board jumpers and set inputscale(j)

'Absent boards get inputscale = 0

'Others will return 1, 2, 5 or 10

x = INP(physicaladdress(J) + 5) 'read jumpers w4, w5, w6, w7

x = x AND &H70

SELECT CASE x

    CASE 96

    inputscale(J) = 1

    CASE 80

    inputscale(J) = 2

    CASE 64

    inputscale(J) = 5!

    CASE 48

    inputscale(J) = 10!

    CASE ELSE

    inputscale(J) = 0

END SELECT

NEXT J

'Initialize time base and run through one cycle:

rate = 1000

'Toggle reset line

OUT tbc, &HEF

OUT tbc, &HE3
```

```basic
'Set counter 0 to mode 3
OUT cwr, &H36
'Set sample rate
A = rate * 2
ha = INT(A / 256)
la = A - (ha * 256)
OUT ctr0, la
OUT ctr0, ha
'Set counter 1 to mode 3
'And divide by 4
OUT cwr, &H76
OUT ctr1, 4
OUT ctr1, 0
'Set counter 2 to mode 5
OUT cwr, &HBA
'Set number of samples
B = (SAMPLES - psamples) / 4
hb = INT(B / 256)
lb = INT(B - (hb * 256))
OUT ctr2, lb
OUT ctr2, hb
'Set trigger level
A = INT(setpoint * 1.28 * -1) + 127 - 7
OUT &H203, A
'Arm time base in chosen mode
'First create time base control word
tbctw = &HF7
'clkie always equals 1
IF clkie = 1 THEN tbctw = tbctw AND &HFB
'Test mode$( PLUS or MINUS )
IF mode$ = "MINUS" THEN tbctw = tbctw AND &H7F
OUT tbc, tbctw AND &HED
'Wait loop to get pretrigger data
ptdelay = psamples * rate * .000001
A = TIMER + ptdelay + 1
```

70

```
'This loop generates and error when timing through midnight
WHILE (TIMER < A AND INKEY$ = ""): WEND
OUT tbc, tbctw AND &HFE
OUT tbc, tbctw
'Force a trigger


OUT tbc, tbctw OR 8
FOR J = 1 TO 500: NEXT J
OUT tbc, tbctw
'This ends the time base first cycle and initialization..
RETURN
RECORD: 'recording sequence control
    CLS
    'VIEW
    'WINDOW (0, 2048)-(511, -2048)
    'LINE (50, 2000)-(461, -1850), lincol, B
    LOCATE 3, 27
    PRINT "RECORDING SEQUENCE CONTROL "
    LOCATE 8, 25
    PRINT "1. ) Initialize Recorder Hardware "


    'First set the gain on boards 1 and 2.
    FOR J = 1 TO 2
    SELECT CASE inputscale(J)
        CASE 1
        mask = &H9F
        CASE 2
        mask = &HDF
        CASE 5
        mask = &HBF
        CASE 10
        mask = &HFF
        CASE ELSE
        mask = &HFF
    END SELECT
```

71

```
IF J = 1 THEN mask1 = mask
OUT physicaladdress(J) + 5, mask
NEXT J
'Toggle reset line
OUT tbc, &HEF AND mask1
OUT tbc, &HE3 AND mask1
'Set counter 0 to mode 3
OUT cwr, &H36
'Set sample rate
A = rate * 2
ha = INT(A / 256)
la = A - (ha * 256)
OUT ctr0, la
CUT ctr0, ha
'Set counter 1 to mode 3
'And divide by 4
OUT cwr, &H76
OUT ctr1, 4
OUT ctr1, 0
'Set counter 2 to mode 5
OUT cwr, &HBA
'Set number of samples
B = (SAMPLES - psamples) / 4
hb = INT(B / 256)
lb = INT(B - (hb * 256))
OUT ctr2, lb
OUT ctr2, hb
'Set trigger level
A = INT(setpoint * 1.28 * -1) + 127 - 7
OUT &H203, A
'Arm timebase in chosen mode
'First create time base control word
tbctw = &HF7 AND mask1
'clkie always equals 1
IF clkie = 1 THEN tbctw = tbctw AND &HFB
```

```
'Test mode$( PLUS or MINUS )

IF mode$ = "MINUS" THEN tbctw = tbctw AND &H7F

LOCATE 10, 25

PRINT "2. ) Pretrigger Delay "

OUT tbc, tbctw AND &HED

REM out tbc,tbctw

'Wait loop to get pretrigger data

ptdelay = psamples * rate * .000001

A = TIMER + ptdelay + 1

'This loop generates and error when timing through midnight

WHILE (TIMER < A AND INKEY$ = ""): WEND

LOCATE 12, 25

PRINT "3. ) Arm Recorder...wait for trigger"

OUT tbc, tbctw AND &HFE

OUT tbc, tbctw

B$ = ""

A = 1

WHILE (A <> 0 AND INKEY$ = "")

A = INP(&H205) AND 3

        IF INKEY$ <> "" THEN

        OUT tbc, tbctw OR 8

        FOR J = 1 TO 500: NEXT J

        OUT tbc, tbctw

        END IF

WEND

LOCATE 14, 25

PRINT "4. ) Triggered ...wait until recording complete"

WHILE (A = 1 AND INKEY$ = "")

A = INP(&H205) AND 1

WEND

LOCATE 16, 25

        PRINT "6. ) Recording port number..."; PORTNO

'   PRINT "6. ) Hit any Key to Exit"

'   A$ = ""

'   WHILE A$ = ""
```

```
'       A$ = INKEY$
'       WEND
'       CLS
RETURN
getstartadd: 'read 18 bit stop address from 3 ports and convert to number
            'only 16 bit stop address from 2 ports for 64 k sample memory
        stopadd = 0
        'bits 16 and 17
        REM sadd1 = INP(physicaladdress(i) + 2) AND &H3
        REM sadd1 = sadd1 * 64 * 1024
        REM stopadd = sadd1
        'bits 8 thru 15
        sadd1 = INP(physicaladdress(I) + 1)
        sadd1 = sadd1 * 256
        stopadd = stopadd + sadd1
        'bits 0 thru 7
        stopadd = INP(physicaladdress(I)) + stopadd
        'Calculate starting address by subtracting samples
        starts(I) = stopadd - SAMPLES - 4
        'Starting address is never negative, so fix it if needed.
        IF starts(I) < 0 THEN starts(I) = starts(I) + 65536
RETURN


        'GETDAT and gd1 both get a single value from the recorder memory,
        'this value is returned in the variable DAT.  DAT is a
        'number that ranges from -2048 to +2047.
        'GETDAT will initialize the memory pointer with the value in BRDST.
        'However, this memory pointer need only be initialized one time per
        'record transfer.  Then the calling routine can continuously call gd1
        'and the recorder memory pointer will auto increment after each call.
getdat:
        IF brdst > memmax THEN brdst = brdst - memmax
        IF brdst < 0 THEN brdst = brdst + memmax
        REM str = INT(brdst / 65536)
        REM brdst = brdst - (str * 64 * 1024)
```

74

```
        strhi = INT(brdst / 256)

        strlo = brdst - (strhi * 256)

        OUT physicaladdress(I), strlo

        OUT physicaladdress(I) + 1, strhi

        REM OUT physicaladdress(i) + 2, str

gd1:

        A = INP(physicaladdress(I) + 3)

        B = INP(physicaladdress(I) + 4)

        B = (B AND &HF) * 256

        DAT = (A + B - 2048) * -1

RETURN

'------------------------------------------------------------------

PLOT:  'This routine will plot the arrays to the screen.

        SCREEN 2                        '2 for cga, 3 for      iles and
                                        '9 for ega. (Only required for
                                        'plotting)

        WINDOW (0, 2048)-(SAMPLES, -2048)

        'plot data on screen

        FOR I = 1 TO 2


                GOSUB getstartadd       'Determines the starting point of
                                        'the record in memory

                brdst = starts(I)       'Initializes memory pointer and gets
                GOSUB getdat            'first data point

        PSET (0, DAT)

                FOR J = 1 TO SAMPLES

                        GOSUB gd1       'gets sample in dat and auto inc. ptr

                        LINE -(J, DAT)

                NEXT J

        NEXT I

RETURN

'------------------------------------------------------------------

LOAD:  'Load data into an array

        FOR I = 1 TO 2

                GOSUB getstartadd       'Initializes boards for data
```

75

```basic
        brdst = starts(I)       'downloading and gets first
        GOSUB getdat            'sample
    FOR J = 1 TO (2 * SAMPLES) STEP 2
            GOSUB gd1       'Gets sample and increments pointer
            IF I = 1 THEN
                VRAW(J) = DAT
                VRAW(J + 1) = 0!
            END IF
            IF I = 2 THEN
                PRAW(J) = DAT
                PRAW(J + 1) = 0!
            END IF


    NEXT J
    NEXT I
RETURN
'-----------------------------------------------------------------------
HEADER:
    'This subroutine collects run information for a file header and
    'data processing
        CLS                     'Clear screen
        LOCATE 2, 10            'Locates text
        PRINT "RUN DATA INPUT"
        LOCATE 4, 20
        PRINT "1. Input a Run Number consisting of the Model"
        LOCATE 5, 25
        PRINT "Number and the Angle of Attack in the form XYY."
        INPUT RUNNO$
        LOCATE 6, 25
        PRINT "RUN NUMBER - "; RUNNO$
        LOCATE 8, 20
        PRINT "2. Input Date"
        INPUT DAIT$
        LOCATE 9, 25
        PRINT "DATE-"; DAIT$
```

```
LOCATE 11, 20
PRINT "3. Input Time"
INPUT TYMES
LOCATE 12, 25
PRINT "TIME-"; TYMES
LOCATE 14, 20
PRINT "4. Input Model Description"
INPUT MODELS
LOCATE 15, 25
PRINT "MODEL-"; MODELS
LOCATE 17, 20
PRINT "5. Input Angle of Attack (+/-) in Degrees"
INPUT AOA
LOCATE 18, 25
PRINT "ANGLE OF ATTACK-"; AOA; "DEGREES"
LOCATE 20, 20
PRINT "6. Input Mean Dynamic Pressure in CM of Water"
INPUT PDYN
LOCATE 21, 25
PRINT "MEAN DYNAMIC PRESSURE-"; PDYN; "CM-WATER"
LOCATE 23, 20
PRINT "7. Input Indicated Oscillation Frequency (Hz)"
INPUT OMEGA
LOCATE 24, 25
CLS
LOCATE 3, 20
PRINT "OSCILLATION FREQUENCY-"; OMEGA; "Hz"
LOCATE 5, 20
PRINT "8. Input Barometric Pressure (In Hg)"
INPUT PREF
LOCATE 7, 25
PRINT "BAROMETRIC (REF) PRESSURE-"; PREF; "IN HG"
LOCATE 9, 20
PRINT "9. INPUT Freestream Temperature (F)"
INPUT TEMP
```

```
            LOCATE 10, 25
            PRINT "FREESTREAM TEMPERATURE-"; TEMP; "F"
RETURN
'---------------------------------------------------------------
HOME:
    'This subroutine advances the SCANIVALVE to HOME
                FOR I = 1 TO 50000
                OUT &H2A8, 2
                NEXT I
                OUT &H2A8, 0
RETURN
'---------------------------------------------------------------
PORTNUM:
'This file contains the text of the subprogram PORTNUM in TKDATA
            'This subprogram reads the BCD output of the SCANIVALVE,
            '(PORTNO)
                PORTNO = 255 - INP(&H2A9)
RETURN
'---------------------------------------------------------------
PORT:
'This file contains the text of the subprogram PORT in TKDATA
        'This subprogram advances the SCANIVALVE one port
                FOR I = 1 TO 1000
                OUT &H2A8, 1
                NEXT I
                OUT &H2A8, 0
RETURN
'---------------------------------------------------------------
MEAN: 'Determines the mean values of the two arrays PRAW and VRAW, PBAR
        'and VBAR, and replaces the arrays with the mean values subtracted
        VBAR = 0!
        PBAR = 0!
        FOR I = 1 TO (2 * SAMPLES) STEP 2
                VBAR = CDBL(VRAW(I) * (VSCALE / 2048) + VBAR)
                PBAR = CDBL(PRAW(I) * (PSCALE / 2048) + PBAR)
```

```
        NEXT I
        VBAR = VBAR / SAMPLES
        PBAR = PBAR / SAMPLES
        FOR I = 1 TO (2 * SAMPLES) STEP 2
                VRAW(I) = VRAW(I) * (VSCALE / 2048) - VBAR
                PRAW(I) = PRAW(I) * (PSCALE / 2048) - PBAR
        NEXT I
        RETURN
```

FFTPRAW: 'Replaces PRAW with the FFT of PRAW, (in pseudo-complex form)

```
        n = 2 * SAMPLES
        J = 1
        FOR I = 1 TO n STEP 2
                IF J > I THEN
                        TEMPR = PRAW(J)
                        TEMPI = PRAW(J + 1)
                        PRAW(J) = PRAW(I)
                        PRAW(J + 1) = PRAW(I + 1)
                        PRAW(I) = TEMPR
                        PRAW(I + 1) = TEMPI
                END IF
        m = n / 2
1       IF m > 2! AND J > m THEN
                        J = J - m
                        m = m / 2
                        GOTO 1
                END IF
                J = J + m
        NEXT I
                mmax = 2
2       IF n > mmax THEN
                        ISTEP = 2 * mmax
                        THETA = 6.283185307# / mmax
                        WPR = -2! * SIN(.5 * THETA) ^ 2
                        WPI = SIN(THETA)
```

79

```
                    WR = 1!
                    WI = 0!
            FOR m = 1 TO mmax STEP 2
                FOR I = m TO n STEP ISTEP
                    J = I + mmax
                    TEMPR = WR * PRAW(J) - WI * PRAW(J + 1)
                    TEMPI = WR * PRAW(J + 1) + WI * PRAW(J)
                    PRAW(J) = PRAW(I) - TEMPR
                    PRAW(J + 1) = PRAW(I + 1) - TEMPI
                    PRAW(I) = PRAW(I) + TEMPR
                    PRAW(I + 1) = PRAW(I + 1) + TEMPI
                NEXT I
                    WTEMP = WR
                    WR = WR * WPR - WI * WPI + WR
                    WI = WI * WPR + WTEMP * WPI + WI
                NEXT m
                    mmax = ISTEP
                    GOTO 2
            END IF
RETURN
'_____

XFERINPUT: 'This subroutine loads transfer function values in the standard
            'format from an external file XFER.FCN (which MUST be in the
            'same directory as TKDATA).
        OPEN "XFER.FCN" FOR INPUT AS #1
        FOR I = 1 TO 2048
            INPUT #1, XFER(I)
        NEXT I
RETURN
'_____

IFFTRAW: 'Replaces PRAW with the IFFT of PRAW, (in pseudo-complex form)
        n = 2 * SAMPLES
        J = 1
        FOR I = 1 TO n STEP 2
            IF J > I THEN
```

80

```basic
                    TEMPR = PRAW(J)
                    TEMPI = PRAW(J + 1)
                    PRAW(J) = PRAW(I)
                    PRAW(J + 1) = PRAW(I + 1)
                    PRAW(I) = TEMPR
                    PRAW(I + 1) = TEMPI
              END IF
        m = n / 2
3          IF m > 2! AND J > m THEN
                    J = J - m
                    m = m / 2
                    GOTO 3
              END IF
                    J = J + m
        NEXT I
              mmax = 2
4          IF n > mmax THEN
                    ISTEP = 2 * mmax
                    THETA = -6.283185307# / mmax
                    WPR = -2! * SIN(.5 * THETA) ^ 2
                    WPI = SIN(THETA)
                    WR = 1!
                    WI = 0!
              FOR m = 1 TO mmax STEP 2
                    FOR I = m TO n STEP ISTEP
                          J = I + mmax
                          TEMPR = WR * PRAW(J) - WI * PRAW(J + 1)
                          TEMPI = WR * PRAW(J + 1) + WI * PRAW(J)
                          PRAW(J) = PRAW(I) - TEMPR
                          PRAW(J + 1) = PRAW(I + 1) - TEMPI
                          PRAW(J) = PRAW(I) + TEMPR
                          PRAW(I + 1) = PRAW(I + 1) + TEMPI
                    NEXT I
                          WTEMP = WR
                          WR = WR * WPR - WI * WPI + WR
```

```
                    WI = WI * WPR + WTEMP * WPI + WI
            NEXT m
                mmax = ISTEP
                GOTO 4
        END IF
    FOR I = 1 TO n
        PRAW(I) = PRAW(I) / SAMPLES
    NEXT I
RETURN
'_____

RIRTHETA: 'This subroutine converts an array from real-imaginary pairs to
          'r-theta pairs.
    FOR I = 1 TO 2047 STEP 2
        IF I = 1 THEN
            R = 0!
            THETA = 0!
        ELSE
            R = SQR(PRAW(I) ^ 2 + PRAW(I + 1) ^ 2)
            IF PRAW(I) = 0! THEN
                THETA = 1.570796327# * SGN(PRAW(I + 1))
            ELSE
                THETA = ATN(PRAW(I + 1) / PRAW(I))
            END IF
        END IF

        PRAW(I) = R
        PRAW(I + 1) = THETA
    NEXT I
RETURN
'_____

RTHETARI: 'This subroutine converts an array from r-theta pairs to real-
          'imaginary pairs.
    FOR m = 1 TO 2047 STEP 2
        R = PRAW(m) * COS(PRAW(m + 1))
        I = PRAW(m) * SIN(PRAW(m + 1))
```

```
                    PRAW(m) = R
                    PRAW(m + 1) = I
                    NEXT m
    RETURN
    '----------------------------------------------------------------

    DIVIDE: 'This subroutine replaces PRAW with PRAW/XFER.
            FOR m = 1 TO 2047 STEP 2
                    PRAW(m) = PRAW(m) / XFER(m)
                    PRAW(m + 1) = PRAW(m + 1) + XFER(m + 1)
            NEXT m
    RETURN
    '----------------------------------------------------------------

    ARRAY: 'This subroutine loads PBAR and VBAR into vectors PMEAN and VMEAN,
            'and PRAW and VRAW into arrays P and V. (It truncates the 1024
            'records in PRAW and VRAW into 1000 records, performs an ensemble
            'average over the 10 ensembles and loads a 100 sample array )
            PMEAN(PORTNO) = PBAR
            VMEAN(PORTNO) = VBAR
            FOR k = 1 TO 100
                    PEA = 0!
                    VEA = 0!
                        FOR J = 0 TO 9
                            PEA = PRAW((2 * k) - 1 + (J * 100))
                            VEA = VRAW((2 * k) - 1 + (J * 100))
                        NEXT J

                    P(PORTNO, k) = PEA
                    V(PORTNO, k) = VEA
            NEXT k
    RETURN
    '----------------------------------------------------------------

    NEGALPHA: 'This subroutine halts data acquisition to allow the angle of
             'attack to be changed from positive alpha to negative alpha.
             CLS
```

```
BEEP
LOCATE 8, 10
PRINT "1.) Program execution has been interrupted to allow angle of"
LOCATE 10, 15
PRINT "attack change."
LOCATE 15, 10
PRINT "3.) Change ALPHA to MINUS ALPHA then press any key to continue"
LOCATE 17, 15
PRINT "data collection."
A$ = ""
WHILE A$ = ""
A$ = INKEY$
WEND
CLS
LOCATE 10, 30
PRINT "Data collection continuing."

RETURN
'-------------------------------------------------------------------------
DISK:    'This subroutine creates a file A:RUNXXX.DAT, (where XXX is the
         'RUN NUMBER), and writes all HEADER DATA, the mean pressure and
         'velocity vectors, PMEAN and VMEAM and the pressure and velocity
         'arrays, P and V to that file.
CLS
LOCATE 10, 15
PRINT "INSERT A FORMATTED DISK IN DRIVE A:"

LOCATE 15, 15
PRINT "PRESS ANY KEY TO WRITE RUN DATA TO FILE"
A$ = ""
WHILE A$ = ""
A$ = INKEYS
WEND
CLS
LOCATE 10, 10
```

```
PRINT "Run data is being written to file A:RUN" + RUNNO$ + ".DAT"
'Open the file:
OPEN "A:RUN" + RUNNO$ + ".DAT" FOR OUTPUT AS #2
'Write header data:
        WRITE #2, RUNNO$, DAITE$, TYME$, MODEL$: 'Strings!
        WRITE #2, AOA: 'degrees
        WRITE #2, PDYN * 2.048: 'psf
        WRITE #2, OMEGA: 'hertz
        WRITE #2, PREF * 70.73: 'psf
        WRITE #2, TEMP + 459.7: 'deg Rankine
'Next write pressure and velocity data:
FOR I = 1 TO 46
        WRITE #2, PMEAN(I): 'Mean pressure vector
NEXT I
FOR I = 1 TO 46
        WRITE #2, VMEAN(I): 'Mean velocity vector
NEXT I
FOR I = 1 TO 46
        FOR J = 1 TO 100
                WRITE #2, P(I, J): 'Pressure array
        NEXT J
NEXT I
FOR I = 1 TO 46
        FOR J = 1 TO 100
                WRITE #2, V(I, J): 'Velocity array
        NEXT J
NEXT I
CLS
BEEP
LOCATE 10, 10
PRINT "   RUN COMPLETE AND DATA WRITTEN TO FILE"
RETURN
'-----------------------------------------------------------------------
```

85

APPENDIX B

The following is the data reduction program, REDUCE.FOR, which is written in
FORTRAN.

```
      PROGRAM REDUCE
c
c******** ********* ********* ********* ********* ********* ********* **
c
c   DATA REDUCTION PROGRAM FOR THESIS.  WRITTEN BY
c   LT G. A. CHAMBERLAIN, JR., USN, JANUARY 1990.
c   CURRENT UPDATED VERSION, 11 JULY 1990
c
c   THIS VERSION IS INTENDED FOR USE ON THE PC IN BLDG 230.
c   PROGRAM "REDUCE.FOR" SHOULD BE INSTALLED IN THE ROOT DIRECTCRY OF
c   DRIVE C:, AND FILE "XOVRC.DAT" SHOULD BE IN THE ROOT DIRECTORY OF
c   DRIVE D:. THIS PROGRAM WILL ASK FOR THE APPROPRIATE RUN NUMBER.
c   ONCE INPUT, THE PROGRAM WILL READ FILE "RUNXXX.DAT" FROM DRIVE
c   C:\DATA AND CUE THE USER AS TO THE PROGRESS OF THE DATA REDUCTION.
c   OUTPUT FILES ARE PUPXXX.DAT, PLOXXX.DAT, VUPXXX.DAT, VLOXXX.DAT,
c   CPUXXX.DAT AND CPLXXX.DAT. ALL OF THESE WILL BE WRITTEN TO DRIVE C:
c   IN DIRECTORY C:\DATA\filename.ext. ADDITIONALLY, FILES CL.DAT,
c   CD.DAT AND CMC4.DAT (WHICH MUST EXIST ALREADY IN C:\DATA\ ) WILL
c   BE APPENDED WITH THE CL, CD AND CMC4 INFORMATION FOR THE AOA
c   USED IN THAT EXPERIMENTAL RUN.
c
c
c   FURTHER DATA MANIPULATION WITH SURFER AND GRAPHER IS FACILITATED
c   BY SETTING THE DEFAULT DRIVES IN THE "ENVIRONMENT" MENUS OF THESE
c   PROGRAMS TO LOOK FOR INPUT AND WRITE OUTPUT TO C:\DATA\ ALSO,
```

```
c    DATA FROM EACH RUN CAN BE EASILY COLLECTED AND COPIED TO THE
c    APPROPRIATE DISK FOR STORAGE BY USING THE DOS COPY COMMAND, AS
C    FOLLOWS: " \> COPY ???123.* A: ". IT IS IMPORTANT TO NOTE THAT
c    AS LONG AS ALL FILES THAT ARE CREATED HAVE THE SAME NAME STRUCTURE,
c    ie, ABC123.ext, THEN A SINGLE COPY COMMAND WILL CAUSE THE COMPUTER
c    TO RETRIEVE ALL FILES THAT ARE SO NAMED.
c
c    *** KEY TO LOGICAL UNIT NUMBERS ***
c    7...RUNXYY.DAT, INPUT DATA FILE
c    8...XOVRC.DAT, CONTAINS PRESSURE PORT LOCATION DATA
c    9...CPUXYY.DAT, OUTPUT FILE CONTAINING CPupper vs TIME AND LOCATION
c    10...CL.DAT, OUTPUT FILE CONTAINING CL vs TIME AND LOCATION
c    11...CD.DAT, OUTPUT FILE CONTAINING CD
c    12...CMC4.DAT, OUTPUT FILE CONTAINING CMC4
c    19...CPLXYY.DAT, OUTPUT FILE CONTAINING CPlower vs TIME AND LOCATION
c    20...PUPXYY.DAT, OUTPUT FILE W/ UPPER SURFACE PRESSURE PROFILE
c    21...PLOXYY.DAT, OUTPUT FILE W/ LOWER SURFACE PRESSURE PROFILE
c    22...VUPXYY.DAT, OUTPUT FILE W/ UPPER SURFACE VELOCITY PROFILE
c    23...VLXYY.DAT, OUTPUT FILE W/ LOWER SURFACE VELOCITY PROFILE
c
c******** ********* ********* ********* ********* ********* ********* **
c
      IMPLICIT DOUBLE PRECISION (A-H,L,O-Z)
      DIMENSION PRM(46),VM(46),CPUPR(23,100),CPLWR(23,100),X(23),
     \   CPU2(23),CPL2(23),PRES(46,100),
     \    VELOC(46,100), V2D(100), CL(100), CD(100), CMC4(100)
      CHARACTER XXX*3, RUNID*10
      WRITE (6,*) 'RUN IDENTIFICATION NUMBER CONSISTS OF THREE'
      WRITE (6,*) 'DIGITS, XYY, WHERE X = MODEL SHAPE (1,2 or 3)'
      WRITE (6,*) 'AND YY = ANGLE OF ATTACK IN DEGREES.'
      DO 321 I = 1,10
      WRITE (6,*)
321   CONTINUE
4     WRITE (6,*) 'INPUT RUN IDENTIFICATION NUMBER XYY ?'
      READ (5,1000) XXX
```

```
1000   FORMAT (A3)
       DO 999 I = 1,30
       WRITE (6,*)
999    CONTINUE
       RUNID = 'RUN'//XXX//'.DAT'
5      WRITE (6,*) 'ENSURE DATA DISK CONTAINING FILE ',RUNID,' IS ON'
       WRITE (6,*) 'HARD DISK AT C:\DATA\ . REDUCE.FOR WILL LOOK FOR'
       WRITE (6,*) RUNID,' AT THAT LOCATION. IF NOT, PRESS CTRL-C NOW'
       WRITE (6,*) 'AND COPY ',RUNID,' TO C:\DATA\ THEN BEGIN AGAIN.'
       WRITE (6,*)
       WRITE (6,*)
       WRITE (6,*) 'OUTPUT FILES WILL BE PUPXYY.DAT, PLOXYY.DAT, '
       WRITE (6,*) 'VUPXYY.DAT, VLOXYY.DAT, CPUXYY.DAT AND CPLXYY.DAT.'
       WRITE (6,*) 'ALL OF THESE WILL BE WRITTEN TO C:\DATA\ AND MAY'
       WRITE (6,*) 'BE RETRIEVED TO THE "A" DISK BY THE DOS COMMAND'
       WRITE (6,*) '... COPY C:\DATA\???123.* A: ... '
       WRITE (6,*) 'CL, CD AND CMC4 FILES WILL ALSO BE AMENDED TO'
       WRITE (6,*) 'THIS NEW AOA.'
       DO 6 I = 1,6
       WRITE (6,*)
6      CONTINUE
       PAUSE 'STRIKE <ENTER> TO CONTINUE'
       WRITE (6,*) 'PROCEEDING...'
       OPEN (UNIT = 7, FILE = 'C:\DATA\RUN'//XXX//'.DAT',
     / STATUS = 'OLD')
       OPEN (UNIT = 8, FILE = 'D:\XOVRC.DAT', STATUS = 'OLD')
       OPEN (UNIT = 9, FILE = 'C:\DATA\CPU'//XXX//'.DAT',
     / STATUS = 'NEW')
       OPEN (UNIT = 19, FILE = 'C:\DATA\CPL'//XXX//'.DAT',
     / STATUS = 'NEW')
       OPEN (UNIT = 20, FILE = 'C:\DATA\PUP'//XXX//'.DAT',
     / STATUS = 'NEW')
       OPEN (UNIT = 21, FILE = 'C:\DATA\PLO'//XXX//'.DAT',
     / STATUS = 'NEW')
       OPEN (UNIT = 22, FILE = 'C:\DATA\VUP'//XXX//'.DAT',
```

```
        / STATUS = 'NEW')
         OPEN (UNIT = 23, FILE = 'C:\DATA\VLO'//XXX//'.DAT',
        / STATUS = 'NEW')
         READ (7,*)
         READ (7,*) A
40    CONTINUE
        PI = 3.141592653589
        ALFA = A * PI / 180.0
        READ (7,*) Q
        READ (7,*)
        READ (7,*) P
        READ (7,*) T
        WRITE (6,*) 'READING MEAN PRESSURES'
        DO 60 I = 1,46
        READ (7,*) PRM(I)
60    CONTINUE
        WRITE (6,*) 'READING MEAN VELOCITIES'
        DO 70 I = 1,46
        READ (7,*) VM(I)
70    CONTINUE
        WRITE (6,*) 'READING DYNAMIC PRESSURE CORRECTIONS & COMPUTING'
        WRITE (6,*) 'PRESSURE DISTRIBUTION AND COEFFICIENTS.'
        DO 80 I = 1,23
        DO 90 J = 1,100
            READ (7,*) PRD
            PRES (I,J) = PRD + PRM(I)
            CPUPR(I,J) = 1+(PRES(I,J)/Q)
90    CONTINUE
80    CONTINUE
        DO 100 I = 1,23
        K = I + 23
        DO 110 J = 1,100
            READ (7,*) PRD
            PRES (K,J) = PRD + PRM (K)
            CPLWR(I,J) = 1 + (PRES(K,J) / Q)
```

89

```
110   CONTINUE
100   CONTINUE
      WRITE (6,*) 'READING DYNAMIC VELOCITY CORRECTION & COMPUTING'
      WRITE (6,*) 'VELOCITY DISTRIBUTION OVER THE AIRFOIL.'
      DO 120 I = 1,46
      DO 130 J = 1,100
         READ (7,*) VD
         VELOC(I,J) = VD + VM(I)
130   CONTINUE
120   CONTINUE
      DO 140 I = 1,23
      READ (8,*) X(I)
140   CONTINUE
      DO 998 I = 1,30
      WRITE (6,*)
998   CONTINUE
      WRITE (6,*) 'WRITING PRESSURE/VELOCITY ARRAYS TO HARD DISK:'
      WRITE (6,*) 'UPPER SIDE PRESSURE...'
      DO 997 I = 1,23
      DO 996 J = 1,100
         WRITE (20,6996) X(I),J/100.0,(P+PRES(I,J))
996   CONTINUE
997   CONTINUE
      CLOSE (20)
      WRITE (6,*)'                 ...LOWER SIDE PRESSURE...'
      DO 995 I = 1,23
      K = I + 23
      DO 994 J = 1,100
         WRITE (21,6996) X(I),J/100.0,(P+PRES(K,J))
994   CONTINUE
995   CONTINUE
      CLOSE (21)
      WRITE (6,*) 'UPPER SIDE VELOCITY...'
      DO 222 I = 1,23
      DO 223 J = 1,100
```

```fortran
          WRITE (22,6996) I/23.0, J/100.0, VELOC(I,J)
223   CONTINUE
222   CONTINUE
      CLOSE (22)
      WRITE (6,*) '              ...LOWER SIDE VELOCITY...'
      DO 224 I = 1,23
      K = I + 23
      DO 225 J = 1,100
          WRITE (23,6996) I/23.0, J/100.0, VELOC(K,J)
225   CONTINUE
224   CONTINUE
      CLOSE (23)
6996  FORMAT (1X,F20.5,1X,F20.5,5X,F20.5)
      WRITE (6,*) 'WRITING UPPER SURFACE CP FILE TO HARD DISK:'
      WRITE (6,*) '(DONT FORGET THAT THIS WILL BE NEGATIVE CP)'
      DO 150 I = 1,23
      DO 155 J = 1,100
          WRITE (9,6000) X(I),J/100.0,(-CPUPR(I,J))
155   CONTINUE
150   CONTINUE
      WRITE (6,*) 'WRITING LOWER SURFACE CP FILE TO HARD DISK:'
      WRITE (6,*) '(ALSO NEGATIVE CP)'
      DO 160 I = 1,23
      DO 165 J = 1,100
          WRITE (19,6000) X(I),J/100.0,(-CPLWR(I,J))
165   CONTINUE
160   CONTINUE
6000  FORMAT (1X, F6.3,F20.5,E15.5)
      CLOSE (9)
      CLOSE (19)
      WRITE (6,*) 'PRESS CONTROL-C NOW TO AVOID SPLINE INTERPOLATION.'
      WRITE (6,*) 'PRESSURE, VELOCITY AND Cp FILES ARE ALREADY SAVED.'
      WRITE (6,*)
      WRITE (6,*) 'BEGINNING PERFORMANCE COEFFICIENT CALCULATIONS'
      DO 777 J = 1,100
```

```fortran
          CMC4(J)=0.0
          CL(J)=0.0
          DO 778 I = 1,22
             DCL=(CPLWR(I,J)-CPUPR(I,J))*.5*(X(I+1)-X(I-1))
             CL(J)=CL(J)+DCL
             CMC4(J)=CMC4(J)+(0.25-X(I))*DCL
778       CONTINUE
          CLF=(CPLWR(1,J)-CPUPR(1,J))*.0125
          CMF=CLF*(0.25-X(1))
          CLB=(CPLWR(23,J)-CPUPR(23,J))*.075
          CMB=CLB*(0.25-X(23))
          CL(J)=(CL(J)+CLF+CLB)*COS(ALFA)
          CD(J)=CL(J)*TAN(ALFA)
          CMC4(J)=CMC4(J)+CMF+CMB
777       CONTINUE
c ****** STEADY FLOW LOOP, PRINTS AVG CL, CD, CM ********
c ****** VALUE TO SCREEN FOR IMMEDIATE FEEDBACK *********
c     AVG1 = 0.0
c     AVG2 = 0.0
c     AVG3 = 0.0
c     DO 1234 I = 1,100
c        AVG1 = AVG1 + CL(I)
c        AVG2 = AVG2 + CD(I)
c        AVG3 = AVG3 + CMC4(I)
c        WRITE (6,8888) A,I,CL(I),CD(I),CMC4(I)
c1234    CONTINUE
c     AVG1 = AVG1/100.0
c     AVG2 = AVG2/100.0
c     AVG3 = AVG3/100.0
c     WRITE (6,8889) A, AVG1, AVG2, AVG3
c8888    FORMAT (1X,F5.1,I4,3F10.4,/)
c8889    FORMAT (1X,'AVG VALUES AT ',F8.1,' DEGREES AOA:',/,1X,'CL = ',
c     / F10.4,'  CD = ',F10.4,'  CMC4 = ',F10.4,/)
C ********END OF STEADY FLOW LOOP *********************
      OPEN (UNIT = 10, FILE = 'C:\DATA\CL.DAT', STATUS = 'OLD')
```

```fortran
      DO 10 I = 1,4000
      READ (10,*,ERR=1,IOSTAT = J10) Z
      IF (J10.LT.0) GO TO 1
10    CONTINUE
1     BACKSPACE (10)
      DO 220 J = 1,100
       WRITE (10,6100) A, J, CL(J)
220   CONTINUE
      OPEN (UNIT = 11, FILE = 'C:\DATA\CD.DAT', STATUS = 'OLD')
      DO 20 I = 1,4000
      READ (11,*,ERR=2,IOSTAT = J11) Z
      IF (J11.LT.0) GO TO 2
20    CONTINUE
2     BACKSPACE (11)
      DO 230 J = 1,100
      WRITE (11,6100) A, J, CD(J)
230   CONTINUE
      OPEN (UNIT = 12, FILE = 'C:\DATA\CMC4.DAT',STATUS = 'OLD')
      DO 30 I = 1,4000
      READ (12,*,ERR=3,IOSTAT = J12) Z
      IF (J12.LT.0) GO TO 3
30    CONTINUE
3     BACKSPACE (12)
      DO 240 J = 1,100
       WRITE (12,6100) A, J, CMC4(J)
240   CONTINUE
6100  FORMAT (1x, F6.2, I4,E15.5)
      CLOSE (7)
      CLOSE (8)
      CLOSE (10)
      CLOSE (11)
      CLOSE (12)
      WRITE (6,*) 'DATA REDUCTION COMPLETE.'
C******** ********* ********* ********* ********* ********* ********* **
      END
```

93

LIST OF REFERENCES

1. Karlsson, S. F., An Unsteady Turbulent Boundary Layer, Ph.D. Thesis, Johns Hopkins University, 1958.

2. Miller, J. A., Transition in Oscillating Blasius Flow, Ph.D. Thesis, Illinois Institute of Technology, 1963.

3. Banning, M. R., The Unsteady Normal Force on an Airfoil in Oscillating Flow, M.S. Thesis, United States Naval Postgraduate School, 1969.

4. Miller, J. A., A Simple Linearized Hot-Wire Anemometer, Journal of Fluids Engineering, Vol. 98, December, 1976.