

AD-A120 456

PRELIMINARY MEASUREMENTS AND CODE CALCULATIONS OF FLOW
THROUGH A CASCADE OF DCA BLADING AT A SOLIDITY OF 167
(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA W D MOLLOY

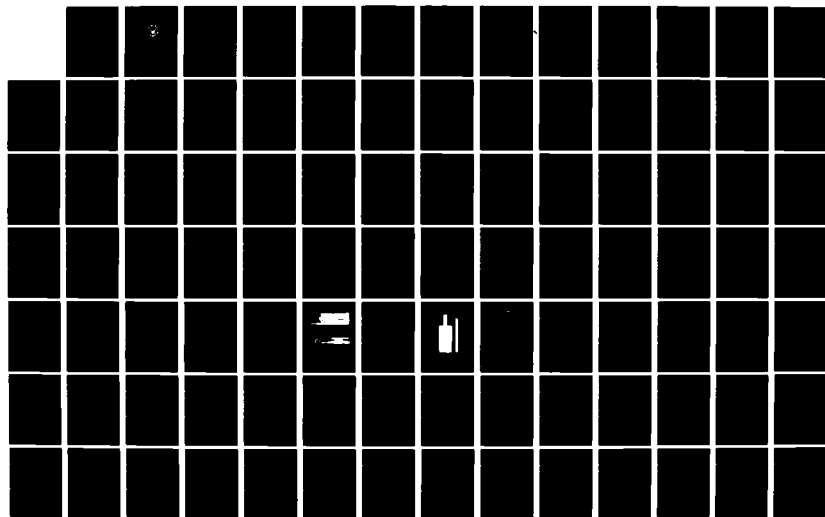
1/3

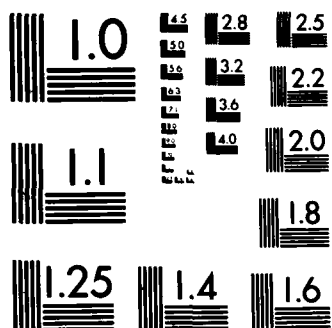
UNCLASSIFIED

JUN 82

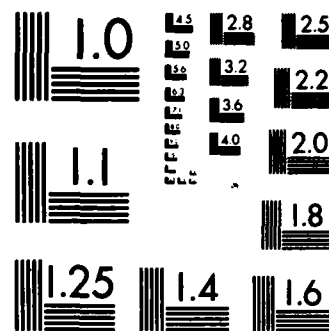
F/G 21/5

NL

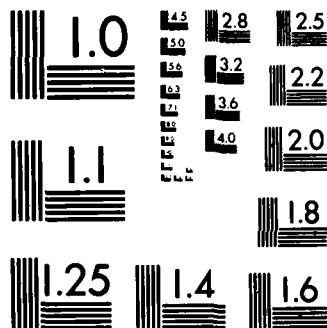




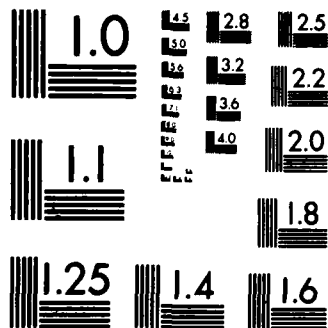
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



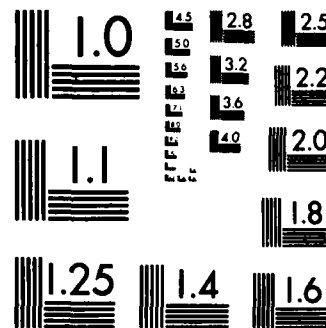
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

PRELIMINARY MEASUREMENTS AND CODE
CALCULATIONS OF FLOW THROUGH A CASCADE
OF DCA BLADING AT A SOLIDITY OF 1.67

by

William D. Molloy Jr.

June 1982

Thesis Advisor:

Raymond P. Shreeve

Approved for public release; distribution unlimited

AD A120456

DTIC FILE COPY

82 10 18 05 5

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. <i>A120 456</i>	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Preliminary Measurements and Code Calculations of Flow through a Cascade of DCA Blading at a Solidity of 1.67		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1982
7. AUTHOR(s) William D. Molloy Jr.		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE June 1982
		13. NUMBER OF PAGES 280
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Cascade DCA Blades QSONIC <i>+ on -</i> <i>Double Circular ARC (DCA)</i>		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) An experimental program to obtain uniform inlet flow to the test blading in a large cascade facility designed to use inlet turning vanes, and to measure the conventional blade element performance, is described. Attempts to reduce non-uniformities ($\pm 1\%$ in velocity) using screens were unsuccessful and so abandoned. Preliminary DCA blade element performance data were obtained without screens at one		

DD FORM 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE/When Data Entered

incidence angle before aero-mechanical problems with the inlet guide vane assembly curtailed testing. The blade surface pressure distribution at the one test condition compared very favorably with the distribution predicted using the NASA computer code QSONIC. Recommendations were made that would avoid the aero-mechanical problems encountered.



Approved for public release; distribution unlimited

Preliminary Measurements and Code
Calculations of Flow through a Cascade
of DCA Blading at a Solidity of 1.67

by

William D. Molloy Jr.
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1974

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1982

Author:

W D Molloy Jr.

Approved by:

Raymond P. Skene
Thesis Advisor

Daniel Hollins
Chairman, Department of Aeronautics

William M. Jolly
Dean of Science and Engineering

ABSTRACT

An experimental program to obtain uniform inlet flow to the test blading in a large cascade facility designed to use inlet turning vanes, and to measure the conventional blade element performance, is described. Attempts to reduce non-uniformities ($\pm 1\%$ in velocity) using screens were unsuccessful and so abandoned. Preliminary DCA blade element performance data were obtained without screens at one incidence angle before aero-mechanical problems with the inlet guide vane assembly curtailed testing. The blade surface pressure distribution at the one test condition compared very favorably with the distribution predicted using the NASA computer code QSONIC. Recommendations were made that would avoid the aero-mechanical problems encountered.

TABLE OF CONTENTS

I.	INTRODUCTION	15
II.	FACILITY DESCRIPTION AND MEASUREMENT APPROACH . .	20
	A. SUBSONIC CASCADE WIND TUNNEL	20
	B. INSTRUMENTATION	22
	C. REFERENCE MEASUREMENTS	23
	D. TEST BLADING	23
	E. DATA ACQUISITION, REDUCTION AND ANALYSIS . . .	24
III.	EXPERIMENTAL PROGRAM AND RESULTS	25
	A. PROGRAM OF TESTS	25
	B. TEST PROCEDURES	25
	1. Cascade Adjustments	25
	2. Measurements	27
	C. VERIFICATION OF INLET GUIDE VANE (IGV) ASSEMBLY	27
	D. TESTING WITH WIRE GAUZE SCREENS	28
	E. PRELIMINARY TESTING OF DCA BLADES	30
IV.	DISCUSSION OF EXPERIMENTAL RESULTS	31
	A. EFFECT OF INLET GUIDE VANE (IGV) MODIFICATION	31
	B. EFFECT OF WIRE GAUZE SCREENS	32
	C. PRELIMINARY TESTING OF DCA BLADES	34
	1. Inlet Uniformity	34
	2. Two-Dimensionality	34
	3. Periodicity	36

4. Blade Performance	36
5. Aero-Mechanical Problems Encountered	37
V. COMPUTATIONAL PROGRAM	40
A. DESCRIPTION OF QSONIC	40
B. APPLICATION TO THE TEST CASCADE	42
C. COMPARISON OF CODE CALCULATIONS AND MEASURED DATA	43
VI. CONCLUSIONS AND RECOMMENDATIONS	44
APPENDIX A: MODIFICATION TO THE INLET GUIDE VANE SECTION OF THE SUBSONIC CASCADE WIND TUNNEL	119
APPENDIX B: SELECTION AND INSTALLATION OF SCREEN MATERIAL	126
APPENDIX C: CASCADE PERFORMANCE PARAMETERS (by F. S. Cina; reproduced with minor changes from Ref. 7)	130
APPENDIX D: INSTRUCTIONS FOR PREPARING INPUT AND OPERATING QSONIC USING A RECTILINEAR CASCADE CONFIGURATION	132
D.1. BACKGROUND INFORMATION	132
D.2. INPUT DESCRIPTION	134
D.3. PREPARING INPUT FILES	146
D.4. PROGRAM OUTPUT	150
D.5. RUNNING THE PROGRAM	151
D.6. QSONIC UPDATE	153
APPENDIX E: QSONIC PROGRAM LISTING	169
LIST OF REFERENCES	278
INITIAL DISTRIBUTION LIST	280

LIST OF TABLES

I.	MEASUREMENT UNCERTAINTY	47
II.	CASCADE CONFIGURATION FOR DCA BLADE TESTS	48
III.	SUMMARY OF MEASUREMENTS WITHOUT SCREENS	49
IV.	SUMMARY OF MEASUREMENTS WITH SCREENS ($\beta_1 = 35^\circ$)	50
V.	PROBE DATA, UPPER PLANE AT MIDSPAN ($i = 5.3^\circ$)	51
VI.	PROBE DATA, LOWER PLANE AT MIDSPAN ($i = 5.3^\circ$)	52
VII.	CENTER BLADE DATA ($i = 5.3^\circ$)	53
VIII.	ADJACENT BLADES DATA ($i = 5.3^\circ$)	54
IX.	BLADE PERFORMANCE DATA	55
D.1.	INPUT DATA FOR MESH GENERATION RUN OF QSONIC	154
D.2.	TEST BLADE COORDINATES	155
D.3.	INPUT DATA FOR FLOW SOLUTION RUN OF QSONIC	156
D.4.	DATA FILE FOR QUASI-3D SOLUTION	157
D.5.	SAMPLE OUTPUT FROM MESH GENERATION RUN	158
D.6.	SAMPLE OUTPUT FROM FLOW SOLUTION RUN	161

LIST OF FIGURES

1.	Subsonic Cascade Facility	56
2.	Plenum Chamber as Modified by Bartocci	57
3.	Plenum Chamber as Modified by Moebius	58
4.	Lower Plane Survey Probe	59
5.	Upper Plane Survey Probe	59
6.	Blade Edge Detail	60
7.	Photograph of Instrumented Blade	61
8.	Instrumented Blade	62
9.	Instrumented Blade Tap Locations	63
10.	Data Acquisition System	64
11.	Probe Survey Data at Midspan of Lower Plane, End Walls at 35°, Points 1 to 50, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$.	65
12.	Probe Survey Data at Midspan of Lower Plane, End Walls at 35°, Points 51 to 100, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$.	66
13.	Probe Survey Data at Midspan of Upper Plane, End Walls at 35°, Points 1 to 50, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$.	67
14.	Probe Survey Data at Midspan of Upper Plane, End Walls at 35°, Points 51 to 100, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$.	68
15.	Probe Survey Data at Midspan of Lower Plane, End Walls at 30°, Points 1 to 50, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$.	69
16.	Probe Survey Data at Midspan of Lower Plane, End Walls at 30°, Points 51 to 100, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$.	70
17.	Probe Survey Data at Midspan of Upper Plane, End Walls at 30°, Points 1 to 50, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$.	71
18.	Probe Survey Data at Midspan of Upper Plane, End Walls at 30°, Points 51 to 100, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$.	72

19.	Probe Survey Data at Midspan of Lower Plane, End Walls at 50°, Points 1 to 50, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$.	73
20.	Probe Survey Data at Midspan of Lower Plane, End Walls at 50°, Points 51 to 100, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	74
21.	Probe Survey Data at Midspan of Upper Plane, End Walls at 50°, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	75
22.	Probe Survey Data at Midspan at Lower Plane, End Walls at 50°, Two Runs, Points 1 to 50, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	76
23.	Probe Survey Data at Midspan at Lower Plane, End Walls at 50°, Two Runs, Points 51 to 100, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	77
24.	Probe Survey Data at Midspan at Lower Plane, End Walls at 30°, Two Runs, Points 1 to 50, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	78
25.	Repetitive Samples with Fixed Probe Position (10" Left of CTR Midspan, End Walls 30°, Lower Plane, $(P_{\text{PLENUM}} - P_{\text{AMB}})/Q_{\text{ref}}$)	79
26.	Repetitive Samples with Fixed Probe Position, 10" Left of CTR, Midspan, End Walls at 30°, Lower Plane $(P_t - P_{\text{AMB}})/Q_{\text{ref}}$	80
27.	Repetitive Samples with Fixed Probe Position, 10" Left of CTR, Midspan, End Walls at 30°, Lower Plane $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	81
28.	Repetitive Samples with Fixed Probe Position (on Centerline at Midspan, End Walls at 30°, Lower Plane $(P_{\text{PLENUM}} - P_{\text{AMB}})/Q_{\text{ref}}$)	82
29.	Repetitive Samples with Fixed Probe Position (on Centerline at Midspan, End Walls at 30°, Lower Plane $(P_t - P_{\text{AMB}})/Q_{\text{ref}}$)	83
30.	Repetitive Samples with Fixed Probe Position (on Centerline at Midspan, End Walls at 30°, Lower Plane $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$)	84
31.	Repetitive Samples with Fixed Probe Position (10" Right of CTR Midspan, End Walls at 30°, Lower Plane $(P_{\text{PLENUM}} - P_{\text{AMB}})/Q_{\text{ref}}$)	85
32.	Repetitive Samples with Fixed Probe Position (10" Right of CTR Midspan, End Walls at 30°, Lower Plane $(P_t - P_{\text{AMB}})/Q_{\text{ref}}$)	86

33.	Repetitive Samples with Fixed Probe Position (10" Right of CTR Midspan, End Walls at 30°, Lower Plane $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$)	87
34.	Probe Survey Data at Midspan, Lower Plane, 16 Mesh Screen, Walls at 35°, Points 1 to 50, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	88
35.	Probe Survey Data at Midspan, Lower Plane, 16 Mesh Screen, Walls at 35°, Points 51 to 100, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	89
36.	Probe Survey Data at Midspan, Upper Plane, 16 Mesh Screen, Walls at 35°, Points 1 to 50, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	90
37.	Probe Survey Data at Midspan, Upper Plane, 16 Mesh Screen, Walls at 35°, Points 51 to 100, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	91
38.	Probe Survey Data Span Traverse, Lower Plane, 16 Mesh Screen, Walls at 35°, 10" Left of CTR, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	92
39.	Probe Survey Data Span Traverse, Lower Plane, 16 Mesh Screen, Walls at 35°, Center of Test Section, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	93
40.	Plane Survey Data Span Traverse, Lower Plane, 16 Mesh Screen, Walls at 35°, Center of Test Section, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	94
41.	Probe Survey Data at Midspan, Lower Plane, 16 Mesh and 2 Mesh, Walls at 35°, Points 1 to 50, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	95
42.	Probe Survey Data at Midspan, Lower Plane, 16 Mesh and 2 Mesh, Walls at 35°, Points 51 to 100, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	96
43.	Probe Survey Data at Midspan, Lower Plane, 4 Mesh Screen, Walls at 35°, Points 1 to 50, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	97
44.	Probe Survey Data at Midspan, Lower Plane, 4 Mesh Screen, Walls at 35°, Points 51 to 100, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	98
45.	Probe Survey Data at Midspan, Lower Plane, 5 Mesh Screen, Walls at 35°, Points 1 to 50, $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$	99

46.	Probe Survey Data at Midspan, Lower Plane, 5 Mesh Screen, Walls at 35°, Points 51 to 100, ($P_{\text{PLENUM}} - P_t$)/ Q_{ref}	100
47.	Wall Static Pressure Distribution	101
48.	Probe Survey Data at Upstream Midspan ($i = 5.3$, $P_{\text{PLENUM}} - P_t$)/ $\bar{Q}_1$, Lower Plane)	102
49.	Blade Surface Pressure Distribution on Three Centermost Blades ($i = 5.3$)	103
50.	Probe Survey Data at Midspan ($i = 5.3$, ($P_{\text{PLENUM}} - P_t$)/ $\bar{Q}_1$, Upper Plane)	104
51.	Probe Survey Data at Midspan ($i = 5.3$, ($X/\bar{X}$), Upper Plane)	105
52.	Probe Survey Data at Midspan ($i = 5.3$, ($P_s - P_{w1}$)/ $\bar{Q}_1$, Upper Plane)	106
53.	Probe Survey Data at Midspan ($i = 5.3$, Outlet Angle, Upper Plane)	107
54.	Spanwise Probe Data Surveyed 1 in. from Suction Side of Centermost Blade ($i = 5.3$, ($P_{\text{PLENUM}} - P_t$)/ $\bar{Q}_1$, Upper Plane)	108
55.	Spanwise Probe Data Surveyed 1 in. from Suction Side of Centermost Blade ($i = 5.3$, $X/\bar{X}$, Upper Plane)	109
56.	Spanwise Probe Data Surveyed 1 in. from Pressure Side of Centermost Blade ($i = 5.3$, ($P_{\text{PLENUM}} - P_t$)/ $\bar{Q}_1$, Upper Plane)	110
57.	Spanwise Probe Data Surveyed 1 in. from Pressure Side of Centermost Blade ($i = 5.3$, $X/\bar{X}$, Upper Plane)	111
58.	Resultant Blade Force Vectors by Momentum Balance (-----) and from Surface Pressure Integration (-----), $i = 5.3$	112
59.	Measured Blade Surface Pressure Distribution ($i =$ 5.3, * = Pressure Side, + = Suction Side)	113
60.	Measured Blade Surface Velocity Distribution ($i =$ 5.3, * = Pressure Side, + = Suction Side)	114
61.	2D Code Blade Surface Mach Number Distribution ($i = 5.3$)	115

62.	3D Code Blade Surface Mach Number Distribution (i = 5.3)	116
63.	Measured Blade Surface Mach Number Distribution (i = 5.3)	117
64.	Blade Surface Mach Number Distribution (i = 5.3) .	118
A.1.	Inlet Guide Vane Assembly	121
A.2.	Cascade Wind Tunnel Sub-Assemblies	122
A.3.	View of th IGV Adjustment Mechanism	123
A.4.	Side View of the IGV Assembly	124
A.5.	View of the Subsonic Cascade Wind Tunnel (North Wall Removed)	125
B.1.	Screen Installation	129
D.1.	Blade Coordinates	167
D.2.	Blade Coordinates Translated and Rotated	167
D.3.	Mesh Points on Blade Surface, Horizontal Chord . .	168
D.4.	Mesh POints on Blade Surface, Chord at Stagger Angle	168

LIST OF SYMBOLS

AVDR	Axial velocity-density ratio
C_{P1}	Coefficient of pressure at the inlet
C_{P2}	Coefficient of pressure at the outlet
$C_{P\text{STATIC}}$	Coefficient of static pressure rise
C	Blade chord (inches)
D	Diffusion factor
i	Incidence angle (degrees)
P	Pressure (in. H ₂ O)
Q	Dynamic pressure (in. H ₂ O)
T	Temperature (°R)
X	Non dimensional velocity
β	Air angle, measured in the cascade midspan plane with respect to the axial direction (degrees)
γ	Stagger angle
σ	Solidity (C/S)
$\bar{w}$	Loss coefficient

Subscripts

amb	Ambient
P	Pressure
PLENUM	Plenum (supply)
s	Static
w1	North wall, lower plane

ACKNOWLEDGMENT

I would like to express my sincere thanks and appreciation to Dr. R. P. Shreeve, Director, Turbopropulsion Laboratory, for his guidance during the project. His affable nature and professional expertise made this study a most enjoyable learning experience.

A special note of thanks to Jim Hammer, Al McGuire, John Morris and Kelly Harris. Without their prompt and efficient help, this study would not have been possible.

I. INTRODUCTION

The need for lightweight, fuel efficient gas turbines that are capable of developing large amounts of thrust or power has motivated a continuing drive to obtain more accurate predictions of the flow through turbomachinery. Cascade testing of blade rows has, in the past, been a logical and relatively inexpensive way to learn more about the phenomena involved in the flow through compressor and turbine stages. It is required more today in order to verify two dimensional and near-two dimensional analysis codes for flow through cascades. Such testing also provides two-dimensional blade element performance data which, in the absence of reliable analytical predictions, are required in the design of compressors and turbine stages. Reference 1 describes how cascade measurements are obtained using a cascade wind tunnel and then used in the design process.

Before subsonic cascade wind tunnel data can be accepted as being valid, the flow conditions must meet three requirements. These criteria are discussed in detail in Refs. 1, 2 and 3. First, any disturbance in the airflow should be caused by the test blades; that is, the inlet flow to the test section must be acceptably uniform.

Secondly, the measured flow characteristics should, ideally, be independent of spanwise position along the test

blades. The flow, ideally, should be two dimensional. Duval [Ref. 3] demonstrated that excellent flow conditions could be achieved in the Naval Postgraduate School Turbo-propulsion Laboratory (NPS/TPL) Subsonic Cascade Wind Tunnel using test blades with an aspect ratio of approximately two. The absence of suction along the walls results however, in some degree of streamline contraction which is measured in terms of an Axial Velocity Density Ratio (AVDR).

The third requirement which must be satisfied is the periodicity of the inlet flow to the test section and of the outlet flow. Within one chord length of the leading edges of the test blades an upstream perturbation occurs as the streamlines adjust to negotiate the blade passages. Since the rectilinear cascade is simulating an infinite cascade of blades, the flow characteristics should be the same at corresponding axial and blade-to-blade positions within each blade passage. This same condition should be true at any measurement plane downstream of the test blading.

As described by Rose and Guttormsen [Ref. 4] several unique features were incorporated into the design of the NPS/TPL Cascade to ensure a two-dimensional and periodic flow at the test blading. Initial evaluations of the facility were conducted and reported in Refs. 3, 4, 5 and 6. Work by Moebius [Ref. 6] involved modifications to the tunnel plenum chamber which established satisfactory uniform

flow at the exit of the bellmouth contraction into the test section.

In order to maintain an aspect ratio close to 2.0 at a solidity of 1.67 Cina [Ref. 7], following the work of Duval [Ref. 3], used a cascade configuration of 20 blades with 3 inch spacing. Cina conducted a program of tests of DCA blading at five (5) different air incidence angles. With this cascade configuration, Cina found that the inlet flow to the test section was uniform in direction and of uniform static pressure, but with an imposed variation in velocity and stagnation pressure resulting from the wakes of inlet guide vanes. Although excellent periodicity was found over pairs of test blades, departure from strictly periodic conditions were detected from one blade passage to another. Cina explained this condition as being the result of the inlet guide vane wakes being separated at two inch intervals and entering a test section configured with a three inch blade spacing. Because of these flow conditions, Cina considered his results to be preliminary.

As a result of these findings the Cascade Wind Tunnel was modified so that inlet guide vanes were provided at one inch intervals. The object of the study reported herein was to obtain blade performance data on Cina's cascade with good periodic and uniform flow conditions. A necessary condition was to obtain agreement in the results for blade forces evaluated from surface pressures and from a momentum

balance. A second objective was to compare measured blade surface Mach numbers with the results of code calculations.

At the outset, it was first necessary to carry out an extensive testing program to verify the new inlet guide vane section and the effect the new spacing had on flow uniformity and periodicity. It was found that the uniformity of dynamic pressure was improved with the inlet guide vanes spaced at one inch intervals. Attempts were made to further improve the flow by the use of (various) wire screens placed downstream of the inlet guide vanes. These methods proved unsuccessful for the range of parameters tested and in fact aggravated the situation.

Cina's testing of the Double Circular Arc blading was repeated without screens and with the Cascade Wind Tunnel configured with the modified inlet guide vane arrangement. Limited measurements were obtained before aero-mechanical problems with the new IGV arrangement, at the higher tunnel speeds, were encountered.

The overall purpose of the testing program initiated by Cina was to obtain data with which to verify design optimization computer codes developed by NASA. Towards this goal a fast, reliable computer analysis code (QSOMIC) for calculating the flow field about a cascade of arbitrary 2-D airfoils was obtained from NASA. The code was adapted and modified to run on the Naval Postgraduate School's IBM 370/3033 computer.

The program QSONIC was developed by NASA to overcome the Mach number limitations of the earlier program TSONIC [Ref. 8]. QSONIC is described in Ref. 9. Procedures for using the program QSONIC at the Naval Postgraduate School's computer facility are given in Appendix D. The procedures are documented for the case of the DCA blading in the NPS/TPL cascade wind tunnel. A program listing is included to document changes made to the code in order to adapt to the operating system of the NPS computer.

Preliminary results show that experimental measurements and code predictions are in very good agreement.

II. FACILITY DESCRIPTION AND MEASUREMENT APPROACH

A. SUBSONIC CASCADE WIND TUNNEL

The Naval Postgraduate School's Rectilinear Cascade Facility is shown in Fig. 1. A description of the facility as it was originally configured is given in Ref. 4. The test facility is an open cycle wind tunnel, designed for the purpose of testing cascades of axial-flow turbomachinery compressor or turbine blades. The unique design of the test section ensures that the airflow paths from the inlet guide vanes to all of the blades of the cascade test section are of equal length. This particular design was intended to eliminate the problems found in other cascade wind tunnels caused by having wall boundary layers of different thicknesses entering the cascade at different points.

As a result of the work reported in Ref. 1, two fine mesh screens were installed at the bellmouth entrance to improve flow stability. A follow-on study into the cascade performance was conducted by Bartocci and is reported in Ref. 5. As a result of Bartocci's findings, plenum turning vanes were installed to direct plenum inlet air towards the bellmouth entrance and to decrease the total pressure fluctuations present at the bellmouth entrance. Figure 2 shows the configuration of the plenum chamber as modified by Bartocci. Reference 6 describes work by Moebius that resulted

in further modification to the plenum chamber in which the original contraction was changed to two two-dimensional contractions in series. After this modification, acceptably small variations in velocity and flow angle were measured at the inlet guide vane station. Figure 3 shows the internal arrangement of the plenum chamber as modified by Moebius and as it was configured for the work presented here.

Using the plenum configuration shown in Fig. 3, Duval [Ref. 3] found that the wakes from the inlet guide vanes were not mixed out at the lower measuring plane of the test section but gave a well defined periodic variation in the impact pressure. The peak-to-peak variation was $\pm 4\%$ of dynamic pressure over two-inch periodic intervals. This condition was undesirable, but was tolerated while looking only to establish the values of parameters required to achieve two-dimensionality and periodicity. Since the inlet flow conditions were not uniform, mass averages were used to calculate properties at the inlet plane from probe measurements.

In order to achieve a solidity of 1.67 and aspect ratio of about 2, a blade spacing of 3 inches was required for the tests carried out by Cina [Ref. 7]. The tests showed unacceptable departures from blade-to-blade periodicity under conditions of high blade loading and the installation of additional guide vanes was recommended. The modification

to the inlet guide vane section of the tunnel resulting from Cina's findings is described in detail in Appendix A.

In the present work, several tests were completed with the tunnel further modified by the introduction of wire screens between the inlet guide vanes and the lower plane of the test section. Appendix B describes the screen material and the criteria used to select the particular screens used in this study.

B. INSTRUMENTATION

The instrumentation used in the present study is that which is described in detail in Ref. 7. Twenty static pressure taps were located on the north and south side walls. The taps on the south wall were connected to a water manometer board so that the uniformity of the static pressure distribution of the inlet and outlet could be monitored visually. Additionally, one upstream tap on each wall and one downstream tap on each wall (near the centerline) were also connected to the Scanivalve so that these static pressures were recorded.

Figure 4 shows the probe that was used for the upstream survey (at the lower plane). The probe was a United Sensor Corporation DA 125 probe, serial number A847-1, calibrated earlier at various Mach numbers and yaw angles in a calibration facility. The United Sensor Corporation DC-125-24-F-22-CD probe, serial number A981-2 (Fig. 5), which was used

at the upper plane was similarly calibrated. The characteristics of the probes were approximated analytically to facilitate automatic data reduction procedures. The calibration and application procedures were those given by Duval [Ref. 3]. Appendix B of Ref. 3 describes both the upstream and downstream probes in detail. The mounting and traversing mechanisms are described in Ref. 7.

C. REFERENCE MEASUREMENTS

Plenum chamber (supply) pressure and temperatures, and atmospheric pressure were recorded on each data scan. Plenum pressure was also displayed on a water manometer board. The total temperature in the test cascade was assumed to be the same as the plenum chamber temperature.

D. TEST BLADING

The double circular arc test blading modeled the midspan section of the stator of the compressor stage reported in Ref. 10. Coordinates describing the profile of the blading are listed in Table D-2. The leading edge and trailing edge are shown in detail in Fig. 6. A photograph of the centermost blade is shown in Fig. 7.

The three blades centrally located in the cascade were constructed with surface pressure taps along the midspan section as shown in Fig. 8. The centermost blade had 19 ports on each of the pressure and suction surfaces and one tap at the leading edge. The two blades adjacent to the

center blade had 3 surface pressure taps located on each of the pressure and suction surfaces. The surface pressure tap locations for the centermost blade are given in Fig. 9.

E. DATA ACQUISITION, REDUCTION AND ANALYSIS

Data were recorded, reduced, and plotted using the modified Hewlett Packard HP-3052A Data Acquisition System shown in Fig. 10. Reference 11 describes the system in detail. The system incorporated a HP-9845A desktop computer as a controller, with all components connected on the HP-98034A HP-IB Interface Bus. A NPS/TPL HG-78K Scanivalve Controller with two 48 port Scanivalves allowed the programmed acquisition of probe and blade surface pressure measurements.

The software used in the present study for acquisition, reduction and plotting of data were developed from software originally created by Duval and Cina. The programs are listed and described separately in Ref. 12.

The uncertainties in the measurements are listed in Table 1.

III. EXPERIMENTAL PROGRAM AND RESULTS

A. PROGRAM OF TESTS

The test program was in three phases. First, in order to verify the new inlet guide vane assembly, tests were conducted with no blading in the test section and with the upper and lower endwalls set parallel at 35° (design condition), 30° and 50° with respect to axial.

Secondly, tests were made of the effect of wire gauze screen materials in reducing non-uniformities in the flow entering the test section. Appendix B describes the type of screens used and how they were installed.

The last phase of the test program was a continuation of the work initiated by Cina. Table II lists the cascade configuration tested. One test was completed successfully before aero-mechanical problems were encountered and testing was halted until the causes were analyzed.

B. TEST PROCEDURES

1. Cascade Adjustments

In the first and second phases of testing, the same procedures were used to realign the cascade for each new configuration. The lower and upper end walls were set to the desired flow angle and the inlet guide vanes were set so that their trailing edges were approximately aligned with

the end walls. The flow was started, and the desired inlet dynamic pressure was set. All tests were run at an average dimensionless inlet velocity (X) of about .13, corresponding to an inlet flow dynamic pressure of 18 inches water. Before recording data, the water manometer board was checked to ensure that the distributions of wall static pressures at the inlet plane and outlet plane were acceptably uniform. If required, the inlet guide vanes were adjusted to obtain uniform static pressure to within ± 0.5 inches of water.

In the third phase of testing, initially the procedures used by Cina [Ref. 7] were followed, namely: the lower end walls were set to the desired inlet air angle and the upper end walls were set approximately to the expected exit air angle. The inlet guide vanes were set very approximately and the cascade was turned on and set to an inlet dynamic pressure of 18 inches water. The upper end walls and the inlet guide vanes were adjusted in turn to obtain wall static pressure distributions upstream and downstream which were acceptably uniform. Using this procedure however it was found on occasion that the inlet air angle sensed by the probe at the lower plane at mid-span could be 2 or 3 degrees different from the setting of the end walls.

The following procedures was subsequently adopted. The lower end walls were set to the desired inlet air angle. The upper end walls were adjusted to be "wide open", to form a diverging passage in which, when the cascade was turned

on (to an inlet dynamic pressure of 18 inches water), the flow was completely separated. The inlet guide vanes were adjusted to obtain the required inlet air angle on the channel center line over the center 24 inches in the blade-to-blade direction. The upper end walls were then moved individually towards the vertical until the lower plane static pressure distribution was uniform and the upper plane static pressure distribution was acceptably uniform at a value close to atmospheric pressure. No readjustment of the inlet guide vanes was made.

2. Measurements

Probe surveys were carried out in the blade-to-blade direction at midspan at the upper and lower planes. In the first and second phases of testing, data were taken over approximately 24 inches of the test section at intervals of 0.25 inches. Also, in order to test the repeatability of measurements, repetitive samples were taken with the probe held fixed at midspan at the lower plane at the center, 10 inches to the right and 10 inches to the left of center.

During the third phase of testing, data were taken using the procedures established by Cina in Ref. 7.

C. VERIFICATION OF INLET GUIDE VANE (IGV) ASSEMBLY

The results of the first phase are presented (as shown in Table III) in Figs. 11 to 32. The results are arranged into groups. The first group (Figs. 11 to 14) are

measurements of tunnel conditions with the end walls set at 35 degrees. Plots of conditions at the lower plane are followed by plots of conditions at the upper plane.

Results for a wall angle of 30 degrees are given next (Fig. 15 to Fig. 18), followed by results for a wall angle of 50 degrees (Fig. 19 to Fig. 21). Data were taken over 24 inches at the lower plane, and also at the upper plane at 30°. At 50°, at the upper plane, only the center 12 inches were surveyed.

The degree of repeatability of conditions in the wind tunnel from test to test (with no change in wall setting) is demonstrated by the results plotted in Figs. 22, 23 and 24.

The last group of plots, Figs. 25 to 33, shows the degree of repeatability in the probe data from scan to scan. Data for these plots were obtained by holding the probe stationary at midspan in three specific blade-to-blade locations in turn and taking 50 repetitive scans of the channels normally recorded for survey profile data. The time interval for each scan was approximately 20 seconds.

D. TESTING WITH WIRE GAUZE SCREENS

The selection and installation of the wire gauze screens is described in Appendix B.

The measurements obtained with the various screen configurations are given (as shown in Table IV) in Figs. 34 to 45. The results are arranged in four groups.

The first group of plots (Fig. 34 to 40) give data obtained with the 16 mesh .0105 inch diameter wire screen installed. Over a blade-to-blade distance from -1.0 to 7.0 inches (Fig. 35) a peak-to-peak variation in velocity of about 1 percent was noted. This is slightly greater than the less than 1 percent (0.9 percent) variation noted over the same survey region without a screen installed (Fig. 12).

The results shown plotted in Figs. 41 and 42 were obtained with two screens installed. One screen (16 mesh, .0105 inch diameter) was installed as discussed in Appendix B, while the second screen (2 mesh, .0400 inch diameter) was attached across the duct at the leading edges of the inlet guide vanes.

The fourth group of plots (Figs. 43 to 46) show the results for two different single screens. Probe survey data for these screens was taken only at the lower plane. The results shown plotted in Figs. 43 and 44 are data obtained with a 4 mesh, .041 inch diameter wire screen installed. The variation in velocity in the blade-to-blade direction was as much as ± 1.1 percent, peak-to-peak. The results shown in Figs. 45 and 46 are for a 5 mesh, .041 inch diameter wire screen. The variation in flow velocity was approximately ± 1.5 percent, peak-to-peak.

E. PRELIMINARY TESTING OF DCA BLADES

The results contained in Tables V to IX and Figs. 47 to 60 are arranged in the following manner.

The results shown plotted in Figs. 47 to 60 are divided into two separate groups. The first group (Figs. 47 to 57) contain results which exhibit the quality of the wind tunnel flow conditions. The second group (Figs. 58 to 60) shows the blade forces (and surface pressures) from survey data. In the first group of figures, results are presented first to examine the inlet flow uniformity (Figs. 47 and 48); second, to examine the outlet flow periodicity (Figs. 49 to 53); and, finally, to examine outlet flow two dimensionality (Figs. 54 to 57).

All points are shown connected with straight lines.

IV. DISCUSSION OF EXPERIMENTAL RESULTS

A. EFFECT OF INLET GUIDE VANE (IGV) MODIFICATION

The probe survey data shown in Figs. 11 and 12 were taken at the lower plane, with the end walls set at 35 degrees. A turning angle of 35 degrees corresponded to the "design point" of the inlet guide vanes, when the airflow from the plenum chamber was at zero angle of incidence to the leading edge of the IGV's. These two figures show that the inlet plane total pressure at midspan in the blade-to-blade direction had a peak-to-peak periodic variation of about ± 2 percent and therefore about a ± 1 percent peak-to-peak variation in the velocity. Corresponding data from probe surveys at the upper plane (Figs. 13 and 14) show that periodic variations in total pressure were reduced to about 25% of the value at the lower plane by the mixing of the inlet guide vane wakes.

It can be seen in Figs. 15 to 21 that at "off-design" conditions for the IGV's (endwalls at 30° and 50°) there is a greater periodic variation in total pressure at the lower plane in the blade-to-blade direction than at the design point conditions. In Fig. 15 and Fig. 16, it can be seen that the periodic variations in total pressure are more pronounced with the flow from the plenum at a negative incidence angle to the IGV's (endwalls at 30°). Except for the first

8 data points in Fig. 15, there is a well defined period of about 1 inch of travel.

Figures 19 and 20 show the probe surveys conducted with endwalls at 50 degrees. At this off design condition the periodic variation in total pressure was considerably greater than the design point and the individual wakes from the inlet guide vanes were much less well defined.

The repeatability of the survey was examined at wall angles of both 50° and 30° . Figs. 22 to 24 show that the non-uniformities in the flow conditions were repeated to (generally) better than 0.5 percent of total pressure. The question was then, to what accuracy could the individual data points be repeated in successive samples. This was examined at several probe positions and the results given in Figs. 25 to 33 explain the departures in Figs. 22 to 24.

B. EFFECT OF WIRE GAUZE SCREENS

All testing with screens was conducted with the end walls and inlet guide vanes set to yield a flow angle of 35 degrees. The data obtained with screens installed were therefore compared with the data obtained without screens, shown in Figs. 11-14. The effect of the pressure drop across the screen on the pressure coefficient plotted in Figs. 34-35 should be noted. With the first screen installed the drop in total pressure from plenum to the probe in the lower plane was about 10 inches of water (plenum pressure minus total

pressure measured by the probe at the lower plane). Without the screen (at design conditions), the pressure drop from the plenum to the probe at the lower plane was approximately 2.0 inches of water. Since Q_{ref} was defined as the difference between plenum pressure and lower wall static pressure, the value of Q_{ref} with the first screen installed was about 28 inches water and without the screen installed, about 20 inches water. In comparing the peak-to-peak variation in P1 seen in Fig. 11 with that obtained with the first screen installed in Fig. 34, the difference in the values of Q_{ref} must be considered. Examinations showed that the peak-to-peak variation in velocity remained at approximately 1% when the screen was installed.

Figures 38 to 40 are plots of data obtained during a spanwise traverse of the probe at the lower plane. These figures show that the pressure drop through the screen and turning vanes was nearly uniform over approximately 8.0 inches of the 10.0 inch span of the tunnel.

In an attempt to generate upstream disturbances that might trigger early boundary layer transition on the IGV's and increase the rate of mixing of the wakes, a second screen was attached to the leading edge of the IGV's. The results in Figs. 41 and 42 showed that this was not the case and in fact the second screen increased the magnitude of the non-uniformities at the lower survey plane. Measurements made with a single 4 mesh (Figs. 43 and 44) and a single 5 mesh

screen (Figs. 45 and 46) of similar blockage showed that neither screen influenced the flow in a particularly favorable manner. The 4 mesh screen caused the peak-to-peak variation in velocity to be about ± 1.1 percent, while the 5 mesh screen caused the variation to be about ± 1.5 percent. This compared unfavorably with the variation obtained without any screen installed which was less than ± 1 percent.

It was therefore decided to proceed with measurements of the test cascade without using screens.

C. PRELIMINARY TESTING OF DCA BLADES

1. Inlet Uniformity

The probe survey at the lower plane in Fig. 48 shows that the inlet plane total pressure at midspan varied in the blade to blade direction less than 0.5 inches of water, with no well-defined spatial period. This was an improvement in the inlet conditions found by Cina [Ref. 7: Fig. 16]. That the spatial period was not well defined agreed with the findings presented earlier in this report. The wall static pressure distribution (Fig. 47) showed small variations (less than 0.5 ins. water peak-to-peak at the lower plane, .4 ins. water peak-to-peak at the upper plane).

2. Two-Dimensionality

The data in Figs. 54 to 57 show that, at the downstream plane, an area of (spanwise) nearly uniform conditions

existed near the centerline of the cascade. Reference 2 points out that at higher loadings it is difficult to establish a substantial spanwise area of uniform flow in the region near the suction side of the blade. This difficulty is evident in the data shown in Figs. 54 and 55 which show that only about 20% of the spanwise distance is acceptably uniform. It is noted that Cina also found reduced areas of uniform flow at this incidence angle; however, 30-40% of the spanwise distance was found to be acceptably uniform in his case. The difference could be the result of the reduced spacing of the IGV's and its effect on the side wall boundary layers.

Figure 58 shows results for inlet and outlet flow angles and blade force vectors derived in two ways as shown in Appendix B of Ref. 7. These two methods are first the applications of momentum conservation to probe survey data and second, the integration of surface pressures measured over the blade area. Reference 2 points out that for truly two dimensional flow the blade forces derived from the two methods should be the same. As shown in Fig. 58 the magnitudes and the directions of the two vectors representing the blade forces are in reasonable agreement. It is noted however that at this particular incidence angle Cina [Ref. 7] measured blade forces that were in total agreement in direction but disagreed slightly in magnitude. The values of the

force magnitudes were about 1.5% lower than those measured by Cina.

3. Periodicity

As can be seen in Figs. 50 and 51 the total pressure and velocity qualitatively repeated fairly well over three central blade passages. Acceptably small quantitative differences are noted. There was also a small but measurable difference in the surface pressures on adjacent blades, as is evidenced in Fig. 49.

4. Blade Performance

Figures 59 and 60 are plots of the pressure and velocity distributions respectively over the centermost blade. These results compare favorably to those obtained by Cina for an incidence angle of 5.3° .

Table IX contains the blade performance parameters deduced from the probe survey data listed together with the data obtained for corresponding test parameters in Ref. 7. While differences in two sets of data are evident, the differences are not large. It is noted that the value of the loss coefficient was only 10% lower than was measured by Cina, but the AVDR was less than 2% different from unity rather than the 6.5% measured by Cina. Further measurements need to be made, particularly in the light of the following discussion, before stronger conclusions can be drawn.

5. Aero-Mechanical Problems Encountered

Fifteen cascade tests were made while evaluating the new inlet guide vane assembly and testing the wire screens. All runs were made without test blades installed, with a plenum total pressure of about 20 inches of water and with the upper and lower end walls parallel. No difficulties were encountered in establishing the desired flow conditions or in using the inlet guide vanes to arrive at a satisfactory distribution of wall static pressures.

The first time the Cascade Wind Tunnel was set up with test blades installed to take data at an air inlet angle of 39.2° , the tunnel operated normally and the test was completed. (The data from this test were subsequently found to be highly suspect and are not reported here.) During the next test, with an inlet air angle of 42.4 , the start-up appeared normal and previously established procedures were used to arrive at a satisfactorily uniform wall static pressure distribution. Tunnel operation appeared to be normal while taking data, but on shutdown a very noticeable high frequency vibration was encountered. Examination of the inlet guide vanes revealed that about 40% of the 60 blades were damaged. Damage included chips missing from the trailing edges, blades bent, cracks at the weld where the blade is joined to its support and indications that the suction side near the leading edge of one blade had been vibrating

against the pressure side near the leading edge of an adjacent blade.

After the inlet guide vanes had been repaired and reinstalled extreme care was used at the beginning of the next test to adjust the inlet guide vanes, with two individuals monitoring the movement of the adjustment mechanism. (IGV adjustment mechanism is described in detail in Appendix A.) A lack of stiffness in the mechanism was suspected as having been a contributing factor to the failure.

One successful test was completed at an inlet flow angle of 42.4° and these data were discussed above.

With the next cascade configuration set, at an air inlet angle of 45.9° , when the IGV's were adjusted after starting up, high frequency vibrations were again experienced. The wind tunnel was shut down and no further testing was attempted at plenum total pressures as high as 20 inches of water gauge.

The difficulty encountered with the IGV's is not fully understood, however the lack of stiffness present in the actuation of the two separate rows of vanes is suspected of having allowed the problem to occur. Certainly, the possibility of an aerodynamic flutter condition being present (due to the misadjustment perhaps) can not be ignored. It was noticed after the initial failure that the lead screw which adjusts the IGV's could be turned but the blades mounted from only one side would be caused to rotate. This

could lead to the trailing edge of one blade contacting the trailing edge of an adjacent blade and effectively closing the blade passages.

Also, the holes in which the cylindrical shanks of the IGV's were held were found not to be uniformly machined. As much as 0.1 inches of movement at the tip of some vanes was possible while others could barely move. (The most seriously damaged blades were found in or adjacent to the larger holes.)

The tendency for the mechanism to "hang-up" on one side would be greater as the vanes became more highly loaded. It is noted that the IGV problem was encountered first when going to increased incidence angles with the compressor test cascade installed. In setting a constant plenum total pressure of 20 inches of water gauge, the static pressure increase across the test blades to a constant atmospheric pressure at the downstream side implies that a progressively increasing dynamic pressure was being generated out of the turning vanes. This can be seen in Table IX, where Q_1 for the test at $\beta_1 = 42.4$ was 25 inches of water.

V. COMPUTATIONAL PROGRAM

A. DESCRIPTION OF QSONIC

The computational code, QSONIC, was developed by the staff at the NASA Lewis Research Center.¹ This code is able to calculate the blade-to-blade flow conditions in turbomachinery blade rows assuming inviscid flow but including streamtube convergence and radius change in the throughflow direction. QSONIC is flexible enough to allow the input of the appropriate boundary conditions to calculate the flow through the test blading in the Subsonic Cascade Wind Tunnel. The program uses a fully conservative solution of the full potential equation combined with the finite volume method on a body-fitted mesh. QSONIC uses an artificial density imposed in the transonic region, if such a region exists, to ensure stability and the capture of shock waves.

The analysis used by QSONIC is a combination of transonic analysis methods to calculate the flow conditions in the vicinity of a cascade of airfoils. A conservative form of the full potential equation is discretized at every point of a body fitted periodic mesh and a mass balance is calculated through the finite volume surrounding the point. The volume

¹The help and advice received from Charles Farrell at NASA Lewis R.C. in the process of adapting QSONIC to the NPS computer is gratefully acknowledged.

is corrected three dimensionally for any change in stream-tube thickness along a streamtube, if a quasi-3D solution is desired. Either elliptic or hyperbolic non-linear partial differential equations are used, depending on the local Mach number.

The analysis used in developing QSONIC made the following assumptions:

- 1) The airflow is inviscid and adiabatic.
- 2) The airflow relative to the test blades is steady.
- 3) Air is a perfect gas with constant specific heat.
- 4) The airflow is isentropic and any discontinuities such as shocks are so weak that they may be approximated as isentropic jumps.
- 5) There is no velocity component normal to the streamsurface.
- 6) The airflow relative to a fixed reference frame (i.e. absolute velocity) is completely irrotational.

Assumption 4 requires that the peak local relative Mach number on a blade surface be 1.4 or less. The Mach numbers measured in test blades in the Subsonic Cascade Wind Tunnel would be well within this limit. However, this limitation would probably preclude the use of QSONIC for analysis of the flow field in the NPS transonic cascade wind tunnel.

There are some combinations of blading geometry and flow conditions which cause unsatisfactory results to be generated. For example, because of assumptions 1 and 6, sharp leading edges at high incidence angles (more than a few degrees) cause large velocity peaks in the blade surface as

the flow tries to turn from the stagnation point to the suction surface.

Reference 9 gives a detailed description of QSONIC and the solution method used including the governing equations. Appendix D describes the operating procedures to use QSONIC on the Naval Postgraduate School's IBM 370/3033 computer. Appendix D also describes the input and output required as applicable to the Subsonic Cascade Wind Tunnel.

B. APPLICATION TO THE TEST CASCADE

Appendix D describes in detail the generation of the input required for QSONIC when applied to test blading in the Subsonic Cascade Wind Tunnel facility. In the present work one comparison of code calculations and measured data was made before testing was stopped. The comparison was for an inlet flow angle (β_1) of 42.4° .

Tables D.1 and D.3 show the input data generated. Table D.6 shows the flow solution output by QSONIC. The flow calculated on the blade surface, using a 15 by 97 mesh, was examined. Figure 61 is a plot of the calculated Mach number along the blade surface using two dimensional inputs. Figure 62 is a plot of computed Mach number incorporating quasi-three dimensional effects. The method of incorporating quasi-three dimensional effects is explained in Appendix D.

C. COMPARISON OF CODE CALCULATIONS AND MEASURED DATA

Table VII lists the data measured in the cascade wind tunnel. C_{p1} , C_{p2} and X_{vel} are defined in Appendix C. The surface Mach number distribution measured on the center blade is shown plotted in Fig. 63.

For comparison, the computed two dimensional, computed quasi-three dimensional, and the Mach number measured in the cascade wind tunnel are plotted together in Fig. 64.

Excellent agreement between all three cases is seen. As would be expected, the greatest difference between measured and calculated data is near the leading edge in the suction side and at the trailing edge of the blade.

VI. CONCLUSIONS AND RECOMMENDATIONS

Based on the results of the first part of the present study, to evaluate the effects of the altered inlet guide vane spacing on flow uniformity and periodicity, the following conclusions were drawn:

1. With the inlet guide vanes operating at design, the peak-to-peak variation in velocity was about $\pm 1\%$, and there was a well defined spatial period of about 1 inch.
2. Operating the cascade wind tunnel in a configuration that requires the inlet guide vanes to be set to other than zero incidence resulted in peak-to-peak variations in velocity greater than $\pm 1\%$ and a spatial period that was less well defined.

The second part of the study, to evaluate the use of wire gauze screens to further reduce the non-uniformities in the flow field, led to the following conclusions:

1. A 16 mesh screen with a blockage factor of .69 had a slightly aggravating effect in the variation in velocity. The peak-to-peak variation in velocity with the screen installed was slightly greater than with no screen installed. This occurred at the expense of a pressure drop of about 10" of water across the screen.
2. The use of screens with similar blockage but with larger mesh and larger diameter wires resulted in larger peak-to-peak variations in the velocity at the lower plane.

The overall objective of the present study was to measure the performance of the DCA test blading. Because of aero-mechanical problems encountered with the inlet turning vanes the performance of the blades was obtained at one

incidence angle only. The following was concluded from the limited test program:

1. As a result of the reduced inlet guide vane spacing the variations in velocity and total pressure at the inlet plane were much less than those reported by Cina.
2. Good periodicity was found from one blade passage to another.
3. An acceptable region of spanwise uniformity (20-40% of blade span) was found at the downstream plane at the one test condition reported. However, this was less than was previously reported for the same incidence angle.
4. The blade forces derived from the integration of surface pressure measurements and probe survey data were in close agreement in both magnitude and direction.
5. The Mach number measured by surface pressure taps over the surface of the blade and Mach number calculated using the program QSONIC were in excellent agreement qualitatively and reasonable agreement quantitatively.
6. The specific reasons for the aero-mechanical problems experienced with the inlet turning vanes have not been identified completely.

Based on these conclusions and other observations, the following recommendations are made:

1. Use of the present inlet guide vane assembly and adjustment mechanism for testing at inlet dynamic pressures higher than about 15 inches of water is unsafe. There are three possible solutions to this problem.
 - a) Operate only within the dynamic pressure range of 10-15 inches of water.
 - b) Modify the new inlet guide vane assembly so that the vanes are supported at both ends on their axes of rotation. [Supporting the IGV's from both ends would prevent flapping vibrations of the (presently) cantilevered vanes. Such vibrations, when the vanes are supported alternately from opposite ends and the gaps are small compared to the chord, might lead to a potentially destructive flutter mode at particular flow velocities.]

- c) Replace the entire inlet turning vane section with one of entirely new design.
2. The procedure should be adopted immediately of adjusting the vanes and walls of the cascade at lower values of the dynamic pressure before increasing the blower speed to the desired operating condition.
 3. More time needs to be spent to examine the flow field produced between the guide vanes and the test blades, and to establish the effects of the movement of the tail boards. The uniqueness of the flow field when the wall static pressure uniformity is used as a criterion of good inlet flow, needs to be examined by conducting repetitive tests at nominally similar test parameters. Only when the adjustment of the flow in the facility and the quality of the flow itself is fully understood should the measured blade performance data be accepted as final.
 4. The upper electrical yaw adjustment mechanism should be replaced with a manual system to greatly decrease the time required to achieve probe pressure (angle) balancing.
 5. Develop the computer code necessary to take advantage of the plotting data created by QSONIC.
 6. Modify the data acquisition and reduction software for the HP 9845 so that real time plots of blade performance parameters can be displayed.

TABLE I. MEASUREMENT UNCERTAINTY

Item	Description	Method	Uncertainty
x	Blade-to-Blade dimension x = 0 in. West end x = 60 in. East end	Position Potentiometer	± 0.01 in.
z	Spanwise dimension z = 0 in. North wall z = 10 in. South wall	Position Potentiometer on probe mount	± 0.01 in.
β_1	Inlet flow yaw angle	Angle Potentiometer on probe mount (hand adjustment)	± 0.2 deg.
β_2	Outlet flow yaw angle	Angle Potentiometer on probe mount (motor driven adjustment)	± 0.5 deg.
P_{plen}	Plenum total pressure	Static tap in plenum chamber $V \approx 0$	± 0.01 in. H_2O gauge
P_s	Static pressure at the test plane	Calibrated pneumatic probe	± 0.1 in. H_2O gauge
P_{wl}	Static pressure at x = 0 in., y = -16.25 in., z = 0 in.	Static tap on North wall	± 0.01 in. H_2O gauge
P_{ATM}	Atmospheric pressure	Absolute Strain Gauge Transducer	± 0.3 in. H_2O
P	Pressure	Scanivalve Transducer	± 0.01 in. H_2O gauge

TABLE II. CASCADE CONFIGURATION FOR DCA BLADE TESTS

Constant Parameters

Number of Blades	20
Spacing (Pitch)	3 inches
Chord	5.01 inches
Solidity	1.67
Thickness	7.0 percent of chord
Camber Angle	45.72 degrees
Stagger Angle	14.72 degrees

Variable Parameters

β_1	42.4 degrees
i	5.3 degrees

TABLE III. SUMMARY OF MEASUREMENTS WITHOUT SCREENS

<u>β_1</u>	<u>Survey Plane</u>	<u>Survey Direction</u>	<u>Fig. Nos.</u>	<u>Purpose</u>
35	Lower	B-B (24 inches)	11 & 12	Flow Field Determination
	Upper	B-B (24 inches)	13 & 14	
30	Lower	B-B (24 inches)	15 & 16	
	Upper	B-B (24 inches)	17 & 18	
50	Lower	B-B (24 inches)	19 & 20	
	Upper	B-B (12 inches)	21	
50	Lower	B-B (24 inches)	22 & 23	Survey Repeatability
30	Lower	B-B (24 inches)	24	
30	Lower	Fixed Probe (10" L. of ζ)	25-27	Point Repeatability
		(on ζ)	28-30	
		(10" R. of ζ)	31-33	

TABLE IV. SUMMARY OF MEASUREMENTS WITH SCREENS
 $(\beta_1 = 35^\circ)$

	<u>Screen</u>	<u>Survey Plane</u>	<u>Survey Direction</u>	<u>Fig. Nos.</u>
1.	16 mesh .0105 wire	Lower	B-B	34 & 35
		Upper	B-B	36 & 37
		Upper	Spanwise	38-40
2.	16 mesh .0105 wire + 2 mesh ahead of IGV's	Lower	B-B	41 & 42
3.	4 mesh .041 wire	Lower	B-B	43 & 44
4.	5 mesh .041 wire	Lower	B-B	45 & 46

TABLE V. PROBE DATA, UPPER PLANE AT MIDSPAN ($i = 5.3^\circ$)

DATA FROM FILE DBRED2:T14
BLADE TO BLADE TRAVERSE MIDSPAN

UPPER PLANE

Point	Loc (in)	Beta	P ₀ Q1bar	P ₉₀ Q1bar	P ₁₈₀ Q1bar	M ₁ Xbar
1	-7.31	-1.59	.5686	.2614	.0920	.6475
2	-6.84	-1.84	.5532	.2669	.1041	.6331
3	-6.37	-1.83	.4276	.2561	.2396	.5630
4	-5.37	-2.56	.5096	.2707	.1380	.6149
5	-5.67	-2.56	.5622	.2638	.0995	.6428
6	-5.48	-2.56	.5780	.2534	.0902	.6530
7	-5.23	-1.79	.5739	.2573	.0916	.6503
8	-5.04	-1.79	.5747	.2626	.0927	.6483
9	-4.80	-1.78	.5697	.2579	.0952	.6480
10	-4.60	-1.78	.5705	.2588	.0901	.6495
11	-4.41	-1.78	.5687	.2575	.0955	.6478
12	-4.16	-1.36	.5650	.2591	.0994	.6451
13	-3.96	-1.35	.5629	.2606	.0987	.6443
14	-3.78	-1.36	.5462	.2682	.1123	.6333
15	-3.58	-1.36	.4863	.2684	.1739	.5978
16	-3.39	-1.35	.4349	.2611	.2269	.5677
17	-3.19	-1.88	.4239	.2571	.2456	.5597
18	-3.00	-3.16	.4886	.2738	.1668	.5989
19	-2.80	-2.41	.5467	.2744	.1108	.6318
20	-2.60	-2.19	.5717	.2636	.0962	.6461
21	-2.40	-1.92	.5751	.2608	.0951	.6482
22	-2.20	-1.92	.5773	.2597	.0926	.6498
23	-2.01	-1.92	.5814	.2599	.0926	.6507
24	-1.81	-1.92	.5810	.2612	.0951	.6493
25	-1.61	-1.48	.5765	.2606	.0901	.6502
26	-1.42	-1.47	.5755	.2644	.0912	.6483
27	-1.22	-1.48	.5748	.2626	.0947	.6476
28	-1.02	-1.46	.5679	.2661	.1004	.6430
29	-.83	-1.47	.5378	.2726	.1217	.6267
30	-.64	-1.46	.4704	.2693	.1954	.5862
31	-.44	-1.47	.4140	.2553	.2698	.5497
32	-.25	-1.47	.4295	.2658	.2442	.5594
33	-.05	-3.00	.5011	.2740	.1616	.6040
34	.15	-2.32	.5562	.2718	.1040	.6371
35	.36	-1.74	.5776	.2672	.0879	.6488
36	.55	-1.71	.5795	.2636	.0865	.6509
37	.75	-1.83	.5880	.2591	.0872	.6541
38	.95	-1.83	.5915	.2613	.0833	.6554
39	1.15	-1.83	.5902	.2591	.0844	.6556
40	1.35	-1.83	.5903	.2587	.0833	.6561
41	1.55	-1.83	.5810	.2601	.0922	.6506
42	2.35	-1.83	.4490	.2598	.2202	.5749
43	2.85	-1.84	.4421	.2670	.2278	.5681
44	3.34	-2.01	.5722	.2600	.0990	.6465
45	3.85	-2.01	.5763	.2597	.0972	.6481

TABLE VI. PROBE DATA, LOWER PLANE AT MIDSPAN ($i = 5.3^\circ$)

DATA FROM FILE LBRED2:T14
BLADE TO BLADE TRAVERSE MIDSPAN

LOWER PLANE

Point	Loc(in)	Beta	Q/Q1bar	Ps/Q1bar	Pt/Q1bar	X-Xbar

1	-4.00	-42.43	.9533	-.1372	.0932	.8457
2	-3.50	-42.42	.9603	-.1444	.0967	.8474
3	-3.00	-42.42	.9567	-.1427	.0996	.8454
4	-2.50	-42.44	.9614	-.1519	.1031	.8480
5	-2.00	-42.43	.9741	-.1469	.0871	.8524
6	-1.50	-42.43	.9767	-.1493	.0875	.8532
7	-1.00	-42.43	.9722	-.1481	.0889	.8522
8	-.50	-42.43	.9743	-.1564	.0921	.8546
9	0.00	-42.43	.9799	-.1567	.0935	.8540
10	.50	-42.42	.9849	-.1592	.0854	.8585
11	1.00	-42.43	.9814	-.1604	.0850	.8593
12	1.50	-42.43	.9871	-.1628	.0779	.8635
13	2.00	-42.43	.9787	-.1615	.0890	.8582
14	2.50	-42.43	.9742	-.1632	.0971	.8556
15	3.00	-42.41	.9731	-.1650	.1059	.8526
16	3.50	-42.42	.9807	-.1663	.0924	.8589
17	4.00	-42.44	.9736	-.1651	.0968	.8567

TABLE VII. CENTER BLADE DATA ($i = 5.3^\circ$)

X/C	Y/C	C_{p1}	C_{p2}	Mach	C_{p1}

PRESSURE SIDE CENTER BLADE					
.0007	.0054	.6655	.4573	.1785	.0796
.0150	.0019	.6417	.4140	.1849	.0824
.0319	.0066	.5240	.1999	.2136	.0951
.0479	.0112	.5287	.2083	.2126	.0946
.0858	.0215	.4871	.1327	.2219	.0968
.1218	.0303	.4818	.1230	.2231	.0993
.1956	.0452	.4700	.1017	.2257	.1004
.2695	.0576	.4757	.1120	.2244	.0999
.3433	.0663	.4764	.1133	.2243	.0998
.4192	.0716	.4892	.1366	.2215	.0986
.4930	.0736	.4871	.1327	.2219	.0988
.5669	.0727	.4771	.1146	.2241	.0997
.6407	.0678	.4956	.1482	.2201	.0979
.7146	.0601	.4889	.1359	.2216	.0986
.7884	.0487	.4949	.1469	.2202	.0980
.8283	.0411	.4672	.0965	.2263	.1007
.8683	.0327	.4537	.0719	.2292	.1020
.9082	.0230	.4245	.0189	.2354	.1047
.9481	.0123	.3815	-.0594	.2443	.1086
.9880	.0006	.2778	-.2482	.2646	.1175
SUCTION SIDE CENTER BLADE					
.0150	.0227	-1.3641	-3.2355	.4973	.2171
.0319	.0310	-.7756	-2.1647	.4249	.1867
.0479	.0389	-.4980	-1.6597	.3878	.1709
.0858	.0563	-.3541	-1.3979	.3675	.1622
.1218	.0710	-.3431	-1.3778	.3659	.1615
.1956	.0970	-.2923	-1.2853	.3585	.1583
.2695	.1170	-.2500	-1.2084	.3522	.1556
.3433	.1309	-.2038	-1.1243	.3453	.1526
.4192	.1399	-.1540	-1.0338	.3377	.1493
.4930	.1432	-.1054	-.9090	.3270	.1447
.5669	.1412	-.0584	-.8599	.3227	.1428
.6407	.1339	.0102	-.7351	.3116	.1380
.7146	.1209	.0631	-.6387	.3028	.1342
.7884	.1021	.1456	-.4897	.2886	.1280
.8283	.0895	.1911	-.4059	.2805	.1245
.8683	.0755	.2326	-.3303	.2730	.1212
.9082	.0593	.2636	-.2740	.2673	.1187
.9481	.0407	.2842	-.2365	.2634	.1170
.9880	.0206	.2948	-.2171	.2613	.1161

TABLE VIII. ADJACENT BLADES DATA ($i = 5.3^\circ$)

X/C	Y/C	Cp1	Cp2	Mach	Angle

PRESSURE SIDE LEFT BLADE					
.1218	.0303	.4014	-.0231	.2402	.1068
.4192	.0716	.4619	.0868	.2275	.1012
.8283	.0411	.4491	.0635	.2302	.1024
SUCTION SIDE LEFT BLADE					
.1218	.0710	-.3438	-1.3791	.3660	.1615
.4192	.1400	-.1505	-1.0273	.3371	.1491
.8283	.0895	.1975	-.3943	.2794	.1240
PRESSURE SIDE RIGHT BLADE					
.1218	.0303	.4640	.0907	.2270	.1010
.4192	.0716	.4658	.0939	.2266	.1008
.8283	.0411	.4597	.0829	.2279	.1014
SUCTION SIDE RIGHT BLADE					
.1218	.0710	-.3285	-1.3513	.3638	.1606
.4192	.1400	-.1519	-1.0299	.3374	.1492
.8283	.0895	.1911	-.4059	.2805	.1245

TABLE IX. BLADE PERFORMANCE DATA

	<u>Present Results</u>	<u>From Ref. 7</u>
β_1	42.42	42.42
i	5.3	5.3
β_2	1.85	0.4
δ	10.44	9.0
D	0.455	0.46
$\bar{\omega}$	0.037	0.041
$\frac{\omega \cos^3 \beta_2}{2 \sigma \cos^2 \beta_1}$	0.020	0.023
$\frac{\omega \cos \beta_2}{2 \sigma} (x 10^2)$	1.09	1.242
AVDR	1.015	1.065
$C_{PSTATIC}$	0.413	0.351
C_{xM}	-1.385	-1.380
C_{yM}	-0.669	-0.566
C_{xB}	-1.330	-1.476
C_{yB}	-0.572	-0.645
$\bar{Q}_1$ (in. H_2O)	25	22
$\bar{X}$	.14	.12

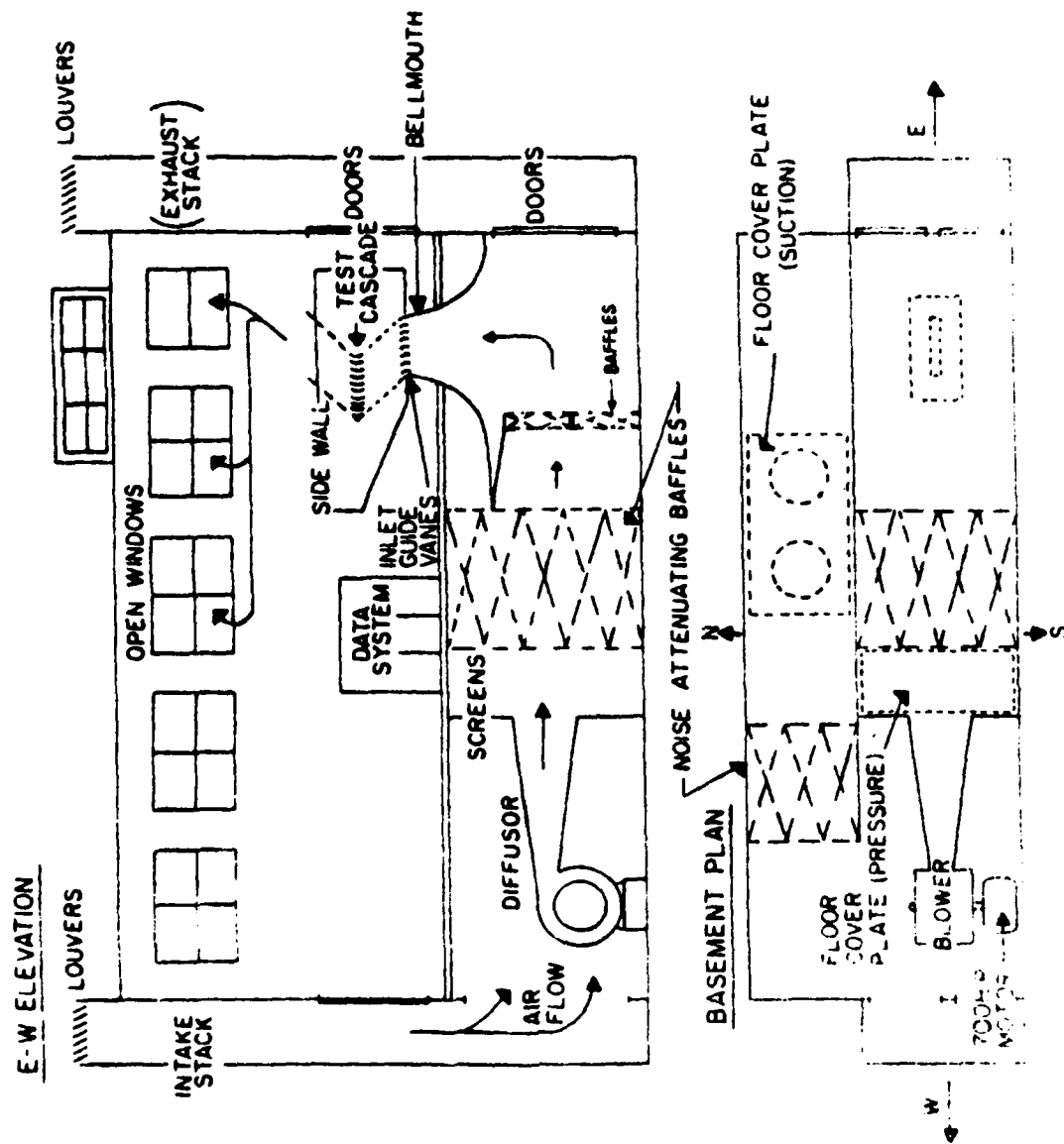


Fig. 1. Subsonic Cascade Facility

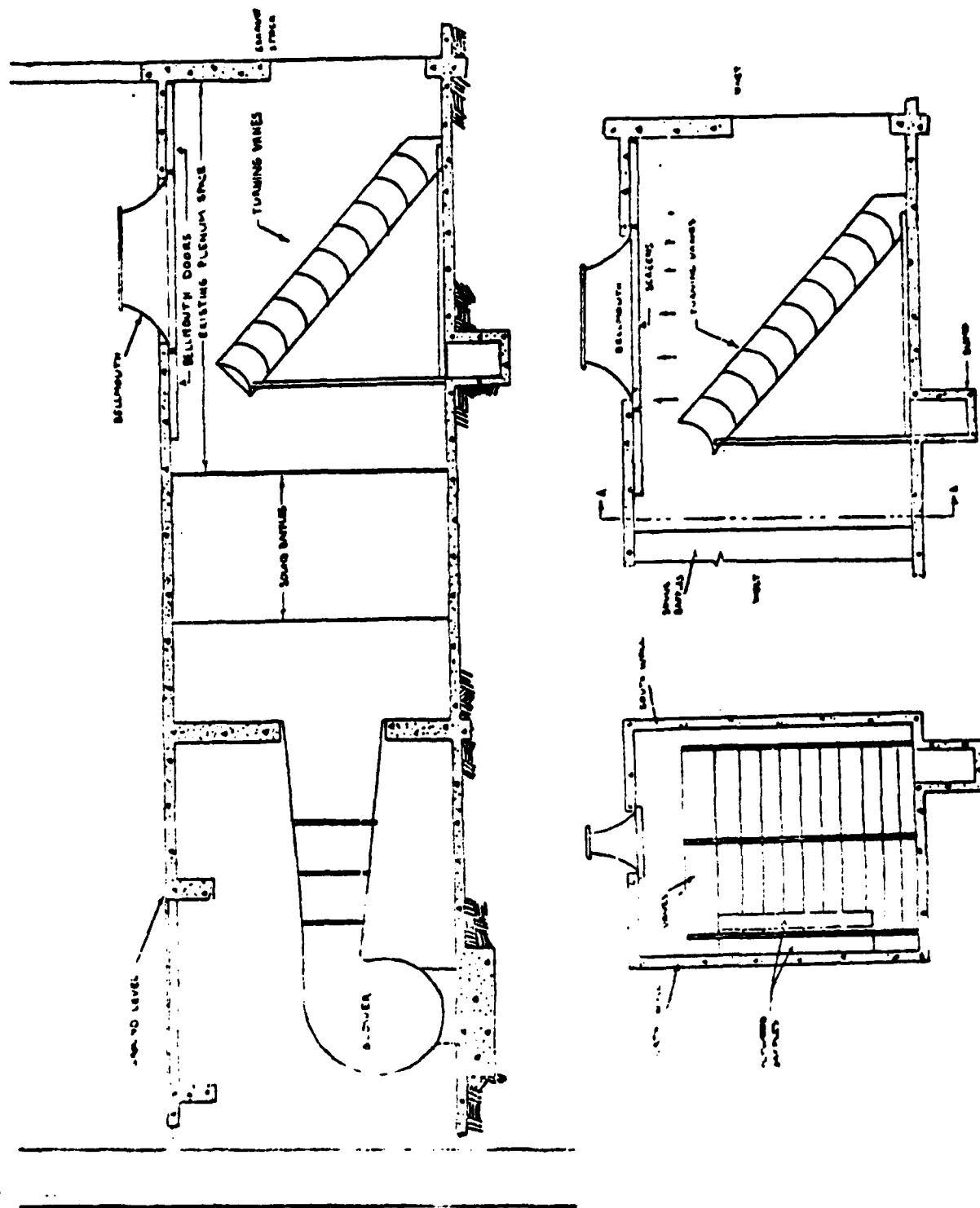


Fig. 2. Plenum Chamber as Modified by Bartocci

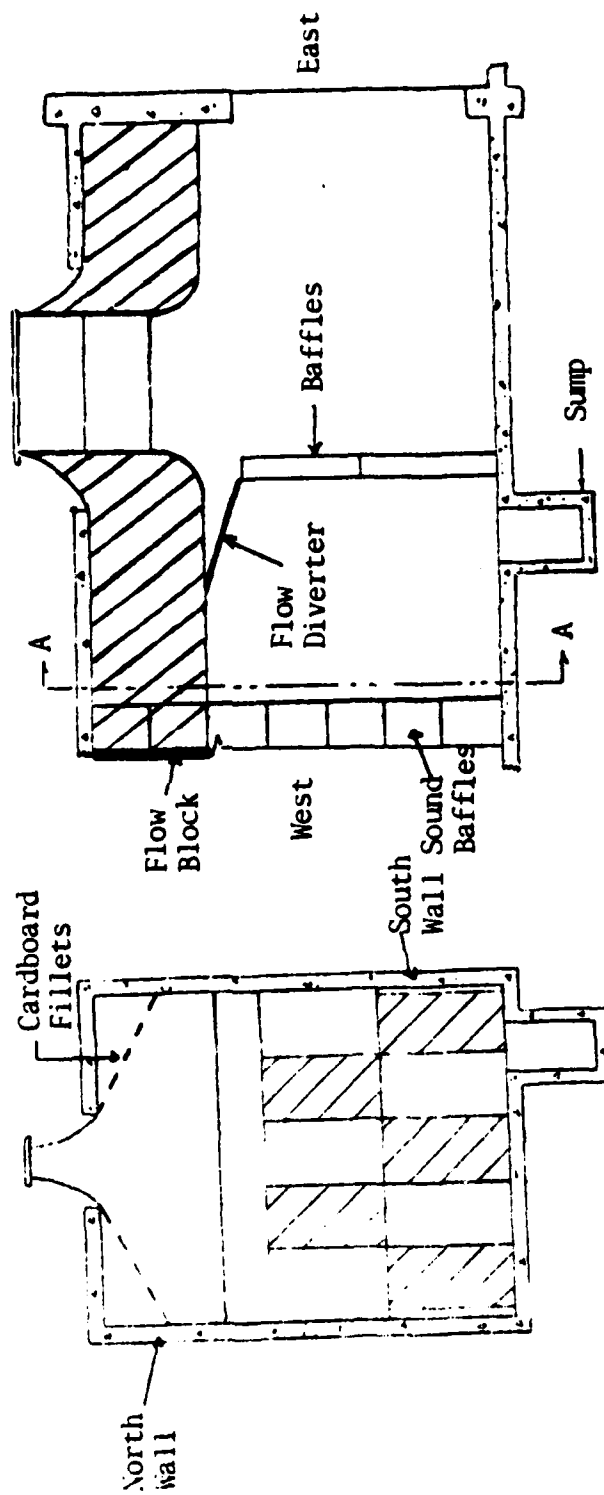


Fig. 3. Plenum Chamber as Modified by Moebius

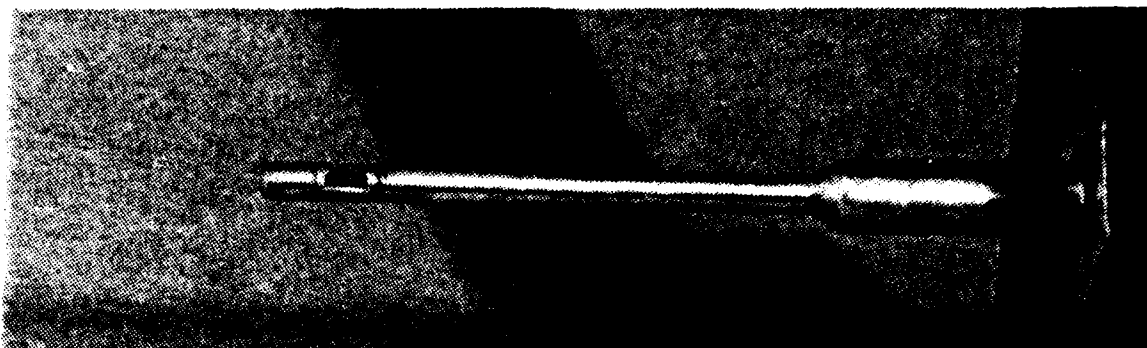


Fig. 4. Lower Plane Survey Probe

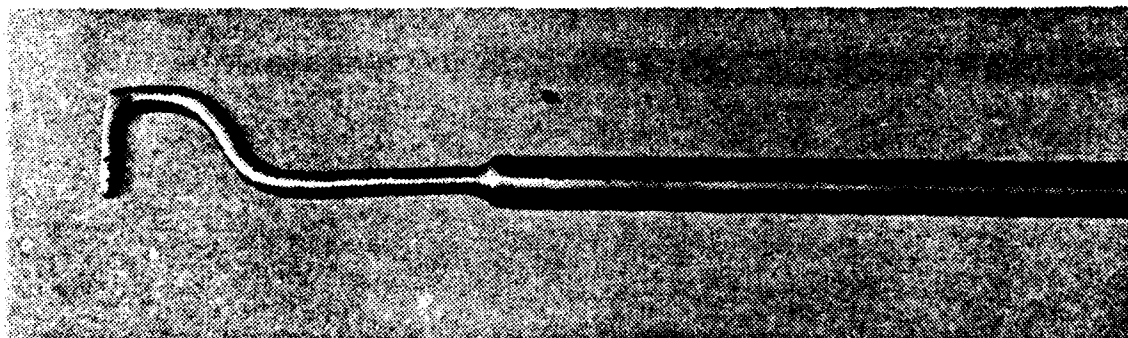


Fig. 5. Upper Plane Survey Probe

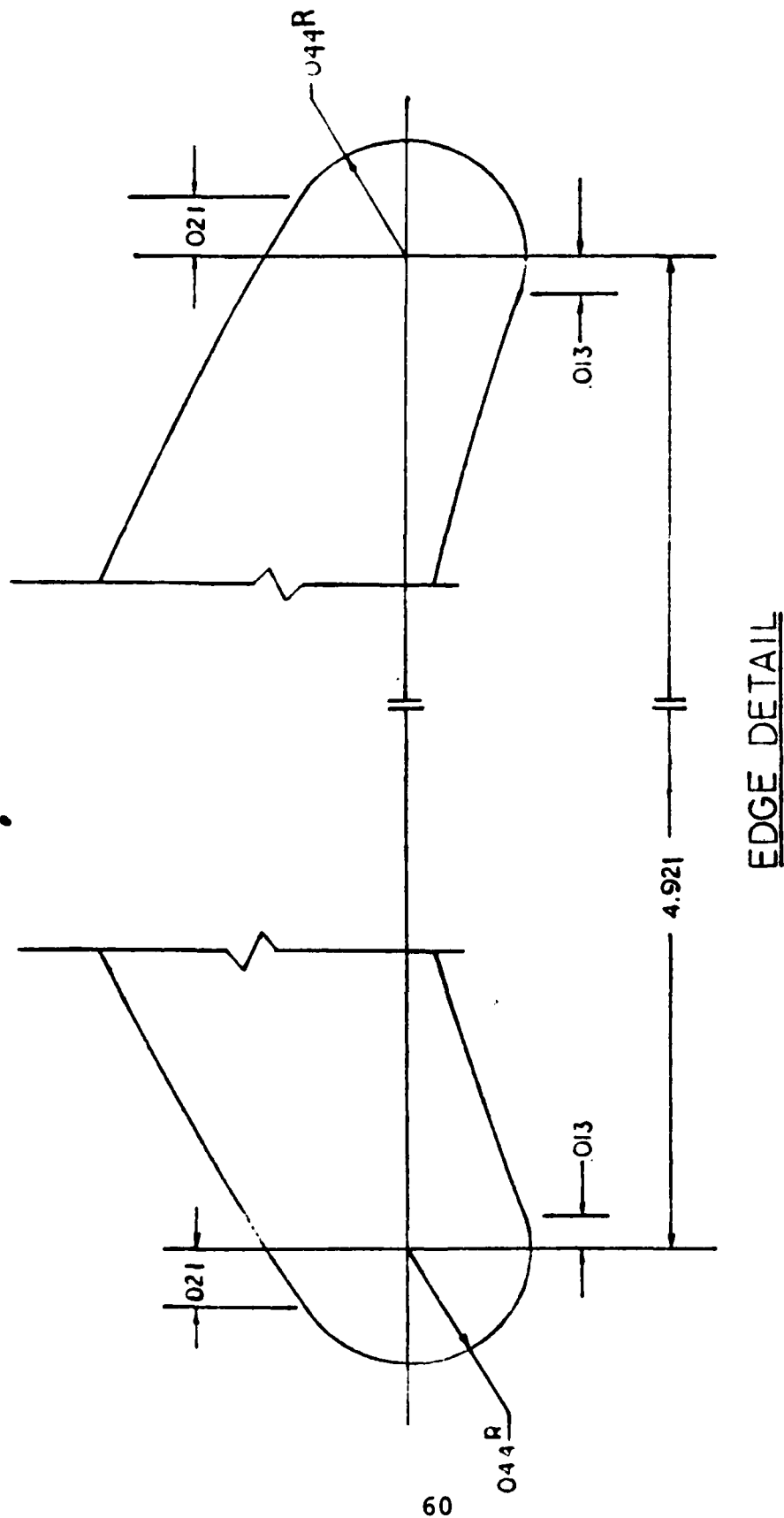


Fig. 6. Blade Edge Detail

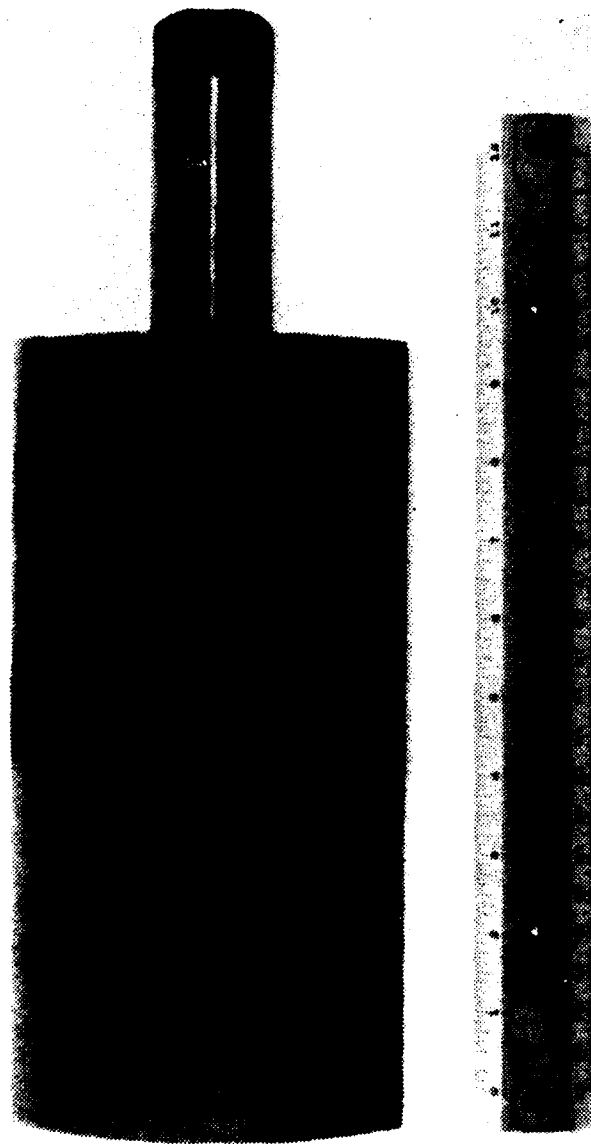


Fig. 7. Photograph of Instrumented Blade

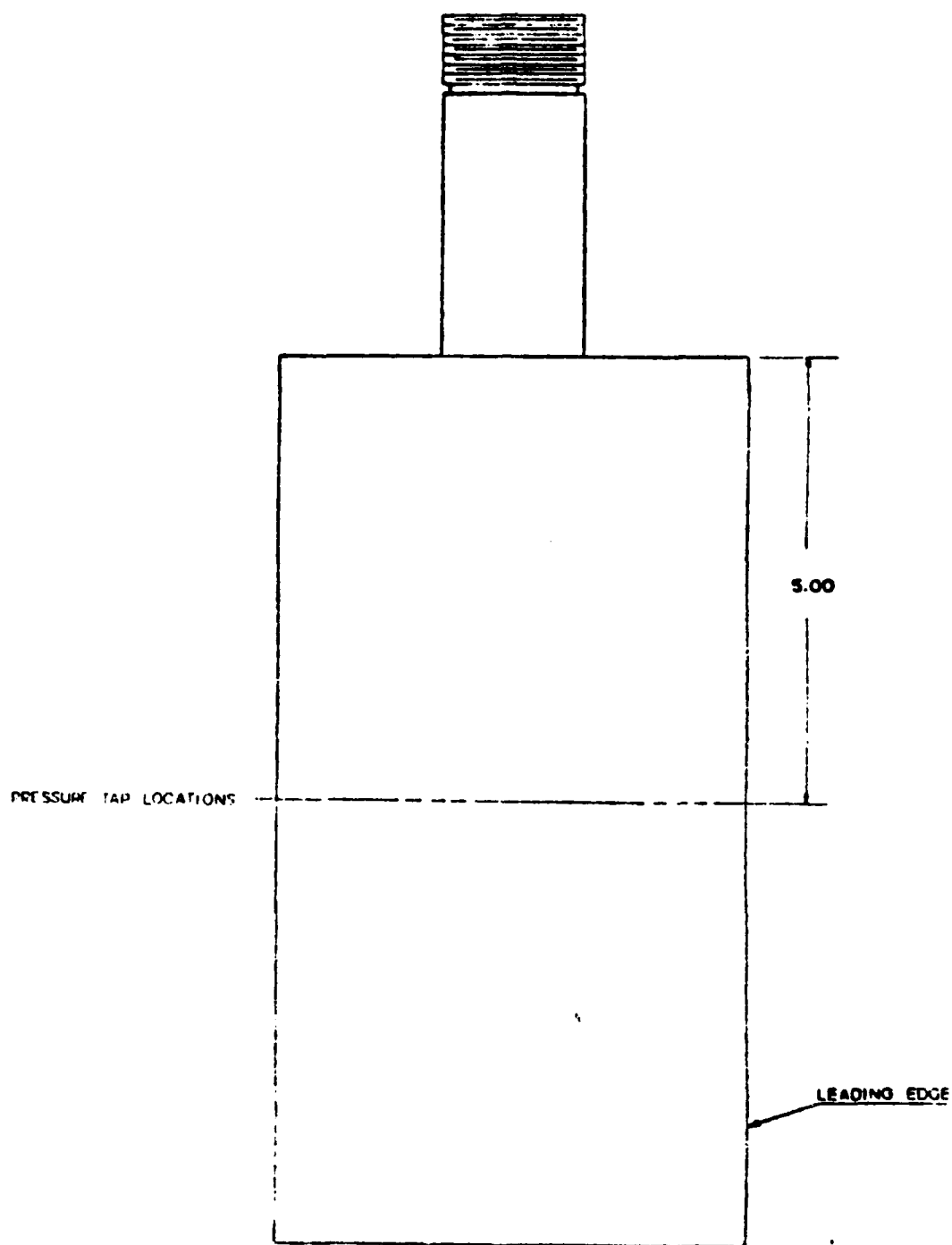


Fig. 8. Instrumented Blade

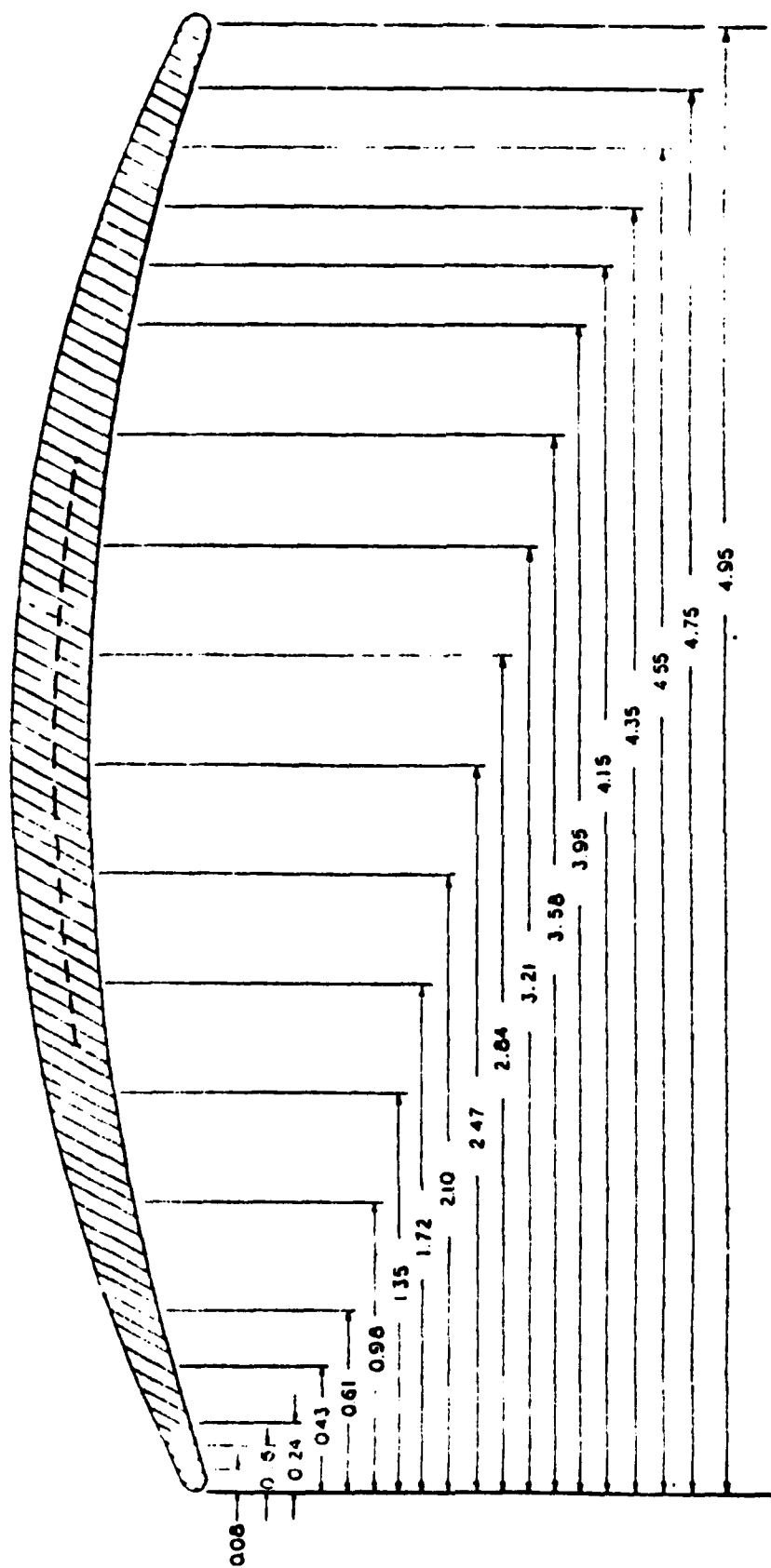


Fig. 9. Instrumented Blade Tap Locations

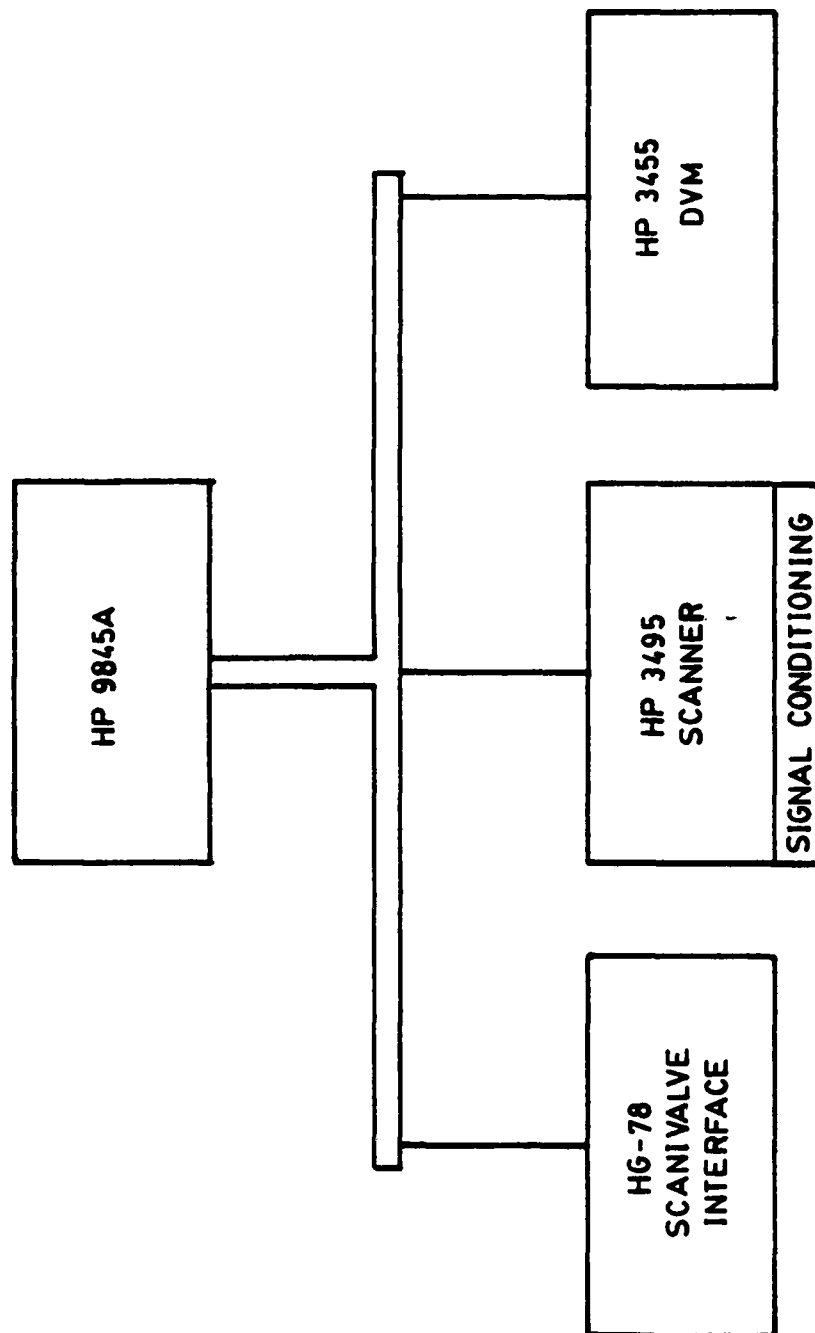


Fig. 10. Data Acquisition System

P(plenum)-P_t Q_{ref}
 POINTS 1 TO 50 LOWER PLANE 35 DEG

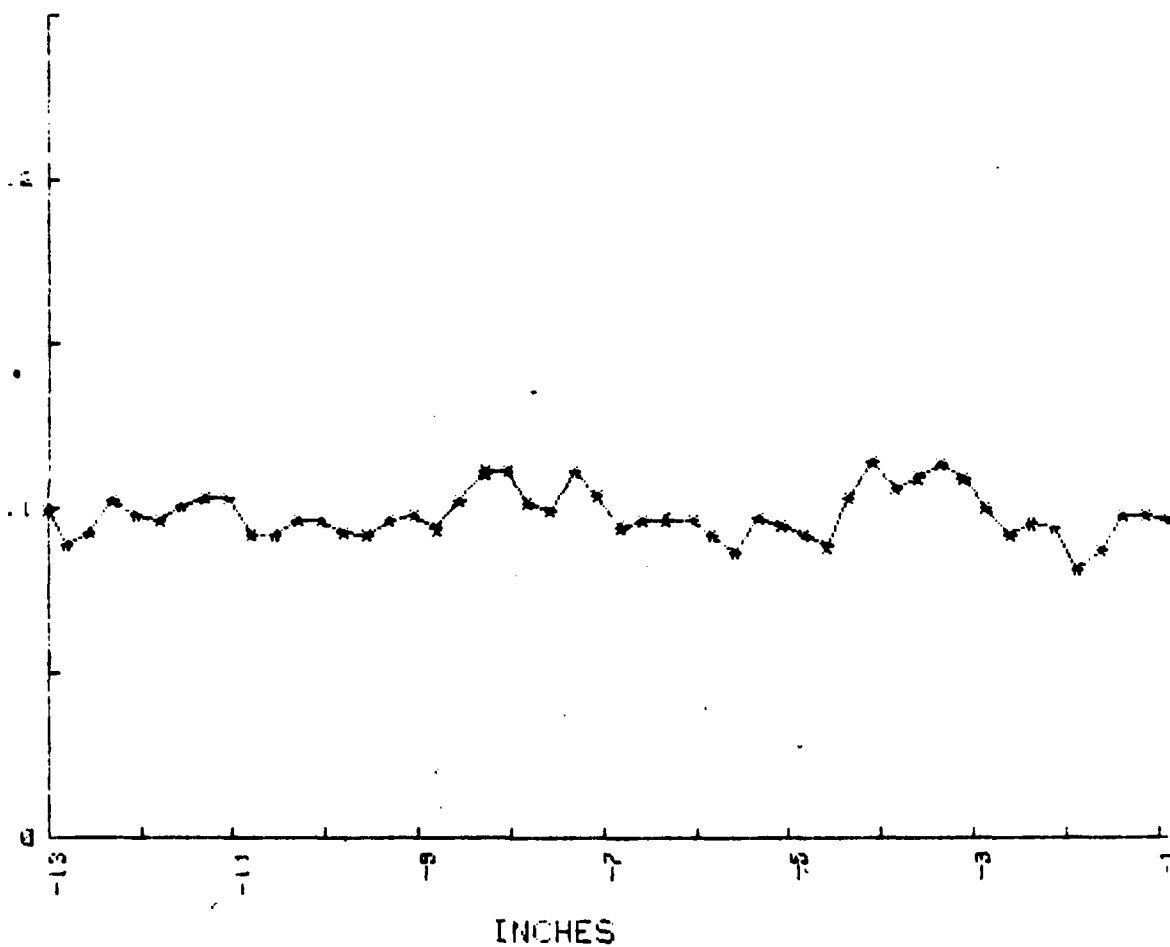


Fig. 11. Probe survey Data at Midspan of Lower Plane
 End Walls at 35°, Points 1 to 50
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P_{\text{plenum}} - P_t - Q_{\text{ref}}$
 POINTS 51 TO 100 LOWER PLANE 35 DEG

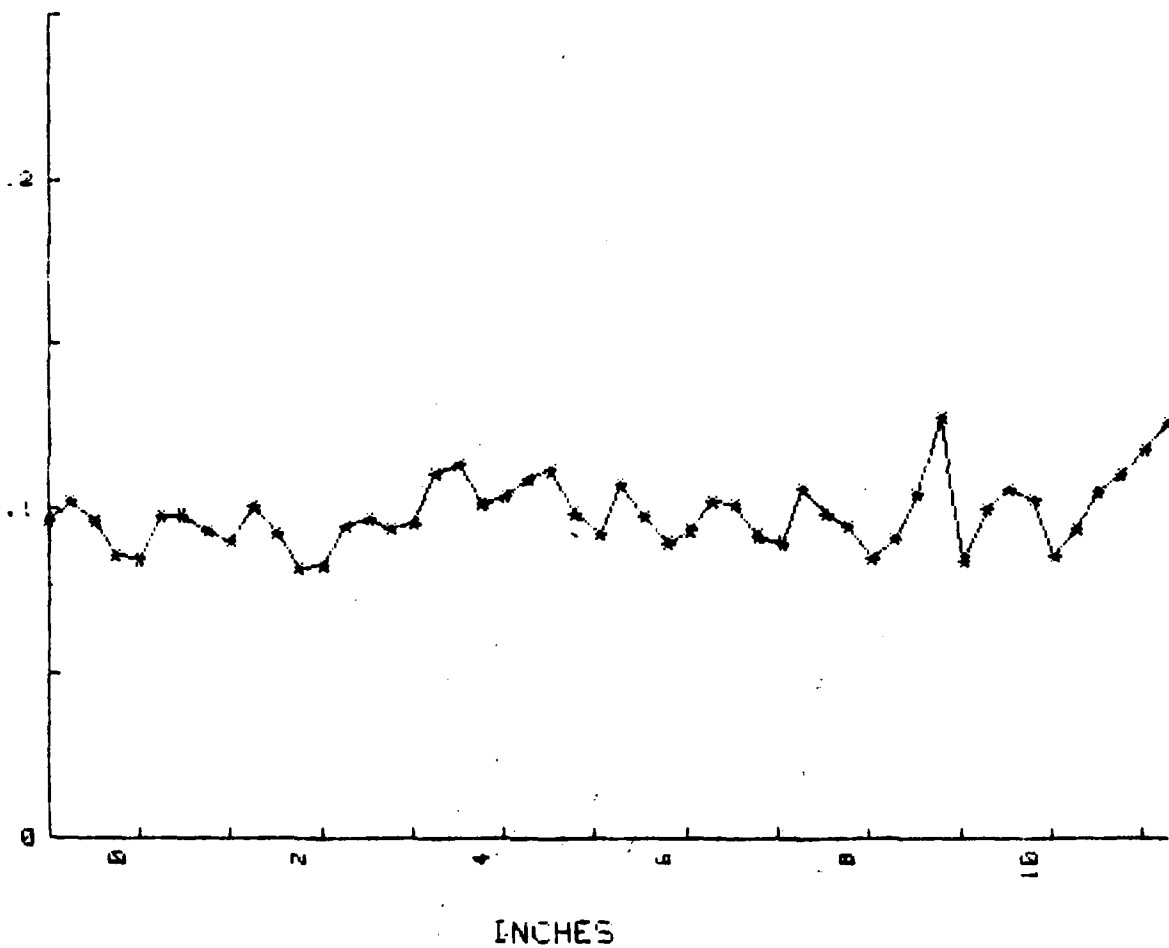


Fig. 12. Probe Survey Data at Midspan of Lower Plane
 End Walls at 35°, Points 51 to 100
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
POINTS 1 TO 50 UPPER PLANE 35 DEG

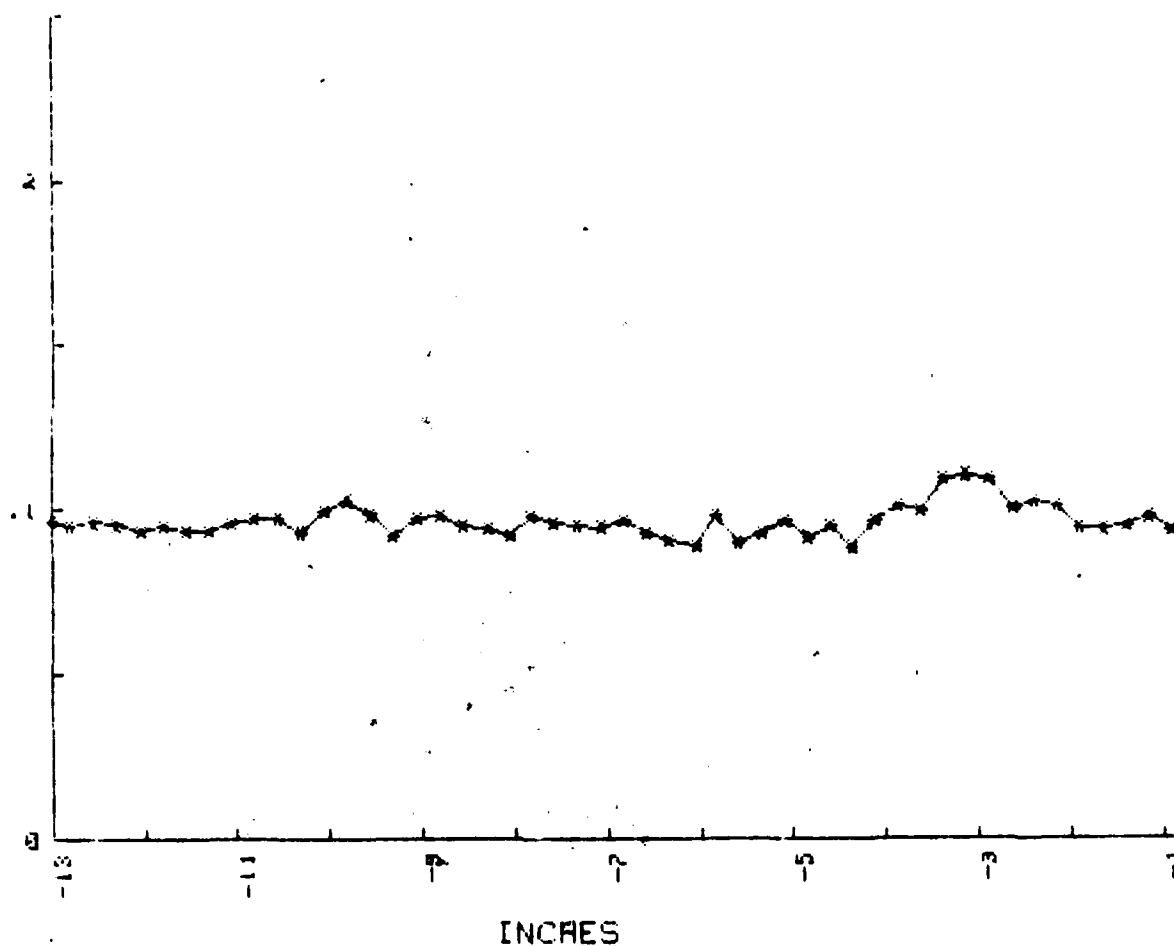


Fig. 13. Probe Survey Data at Midspan of Upper Plane
End Walls at 35°, Points 1 to 50
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
POINTS 51 TO 100 UPPER PLANE 35 DEG

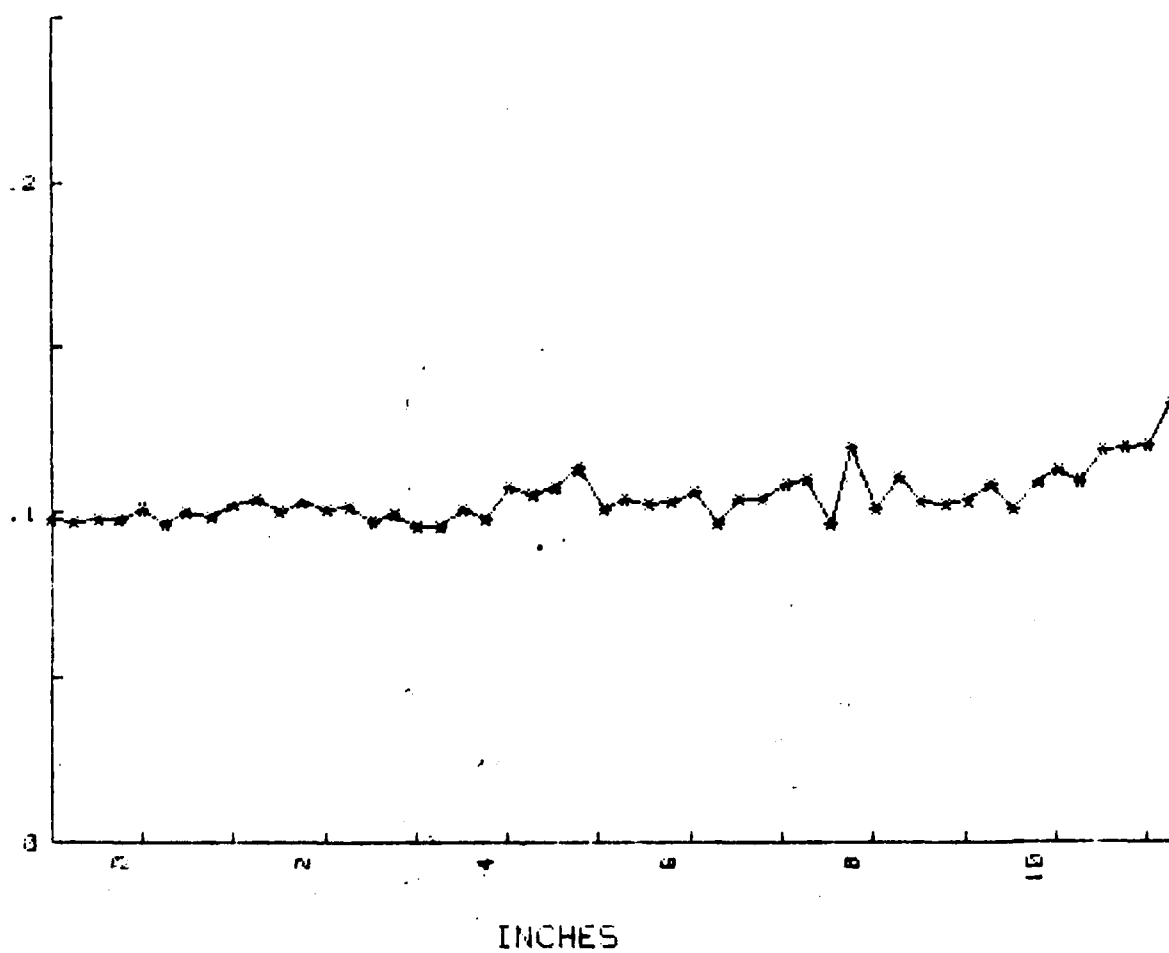


Fig. 14. Probe Survey Data at Midspan of Upper Plane
End Walls at 35°, Points 51 to 100
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$(P_{\text{plenum}} - P_t) / Q_{\text{ref}}$
 POINTS 1 TO 50 LOWER PLANE 30 DEG

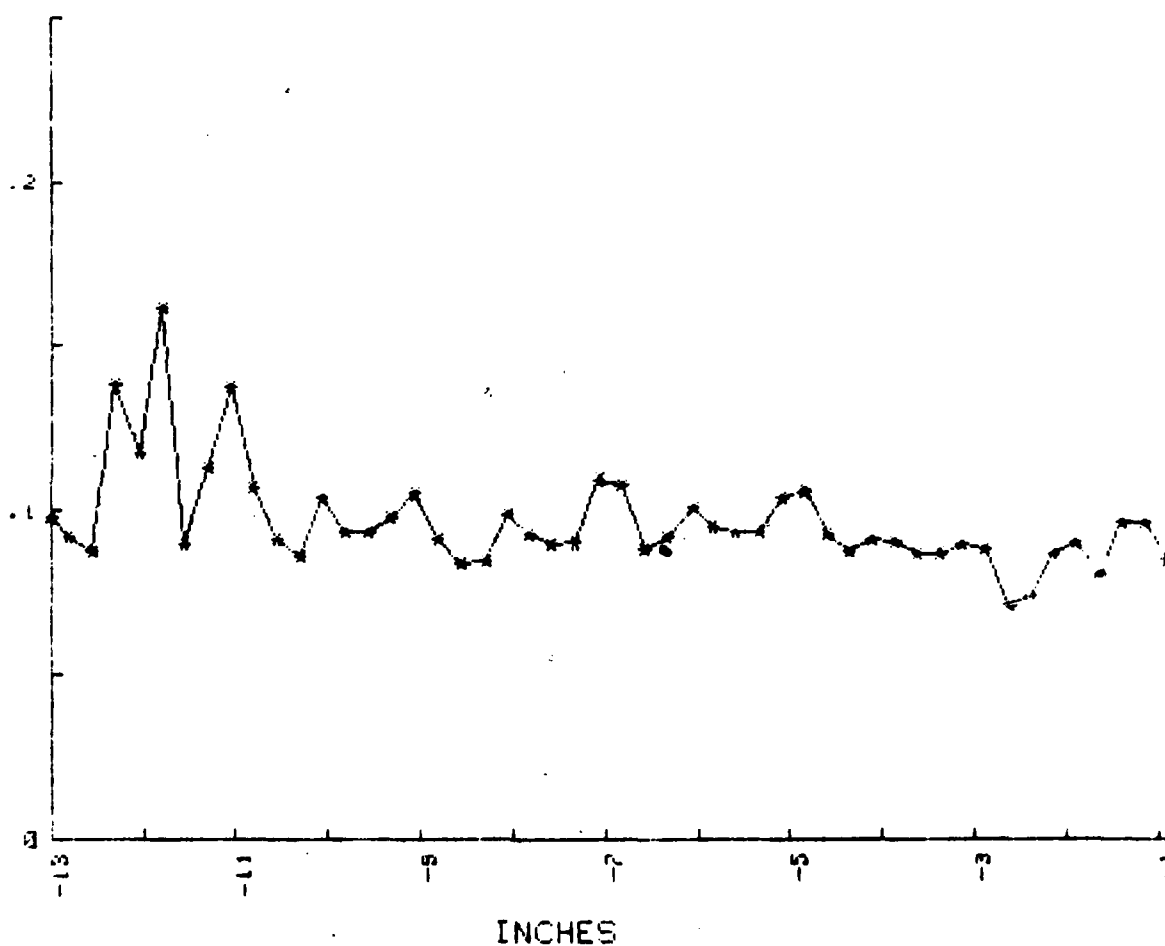


Fig. 15. Probe Survey Data at Midspan of Lower Plane
 End Walls at 30°, Points 1 to 50
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
POINTS 51 TO 100 LOWER PLANE 30 DEG

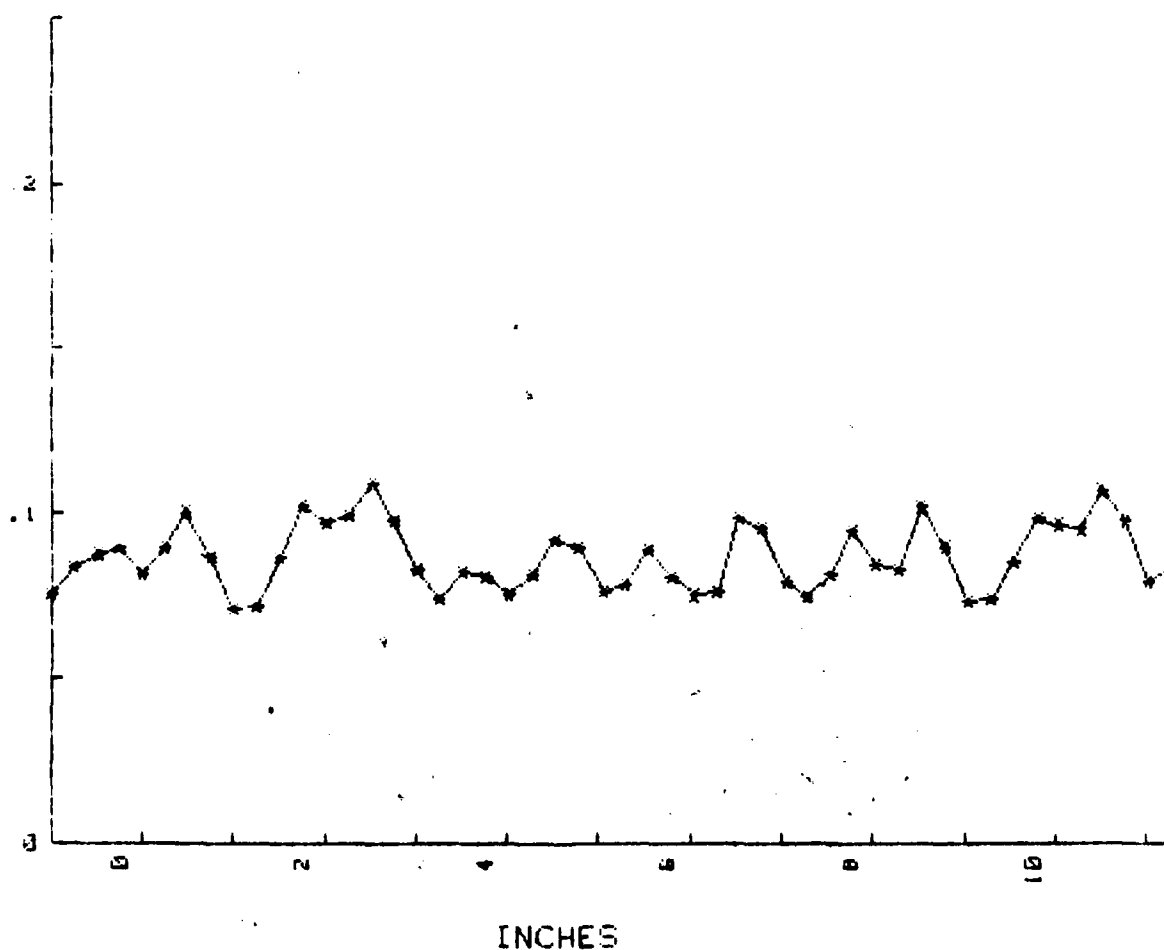


Fig. 16. Probe Survey Data at Midspan of Lower Plane
End Walls at 30°, Points 51 to 100
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$(P_{\text{plenum}} - P_t)/Q_{\text{ref}}$
 POINTS 1 TO 50 UPPER PLANE 30 DEG

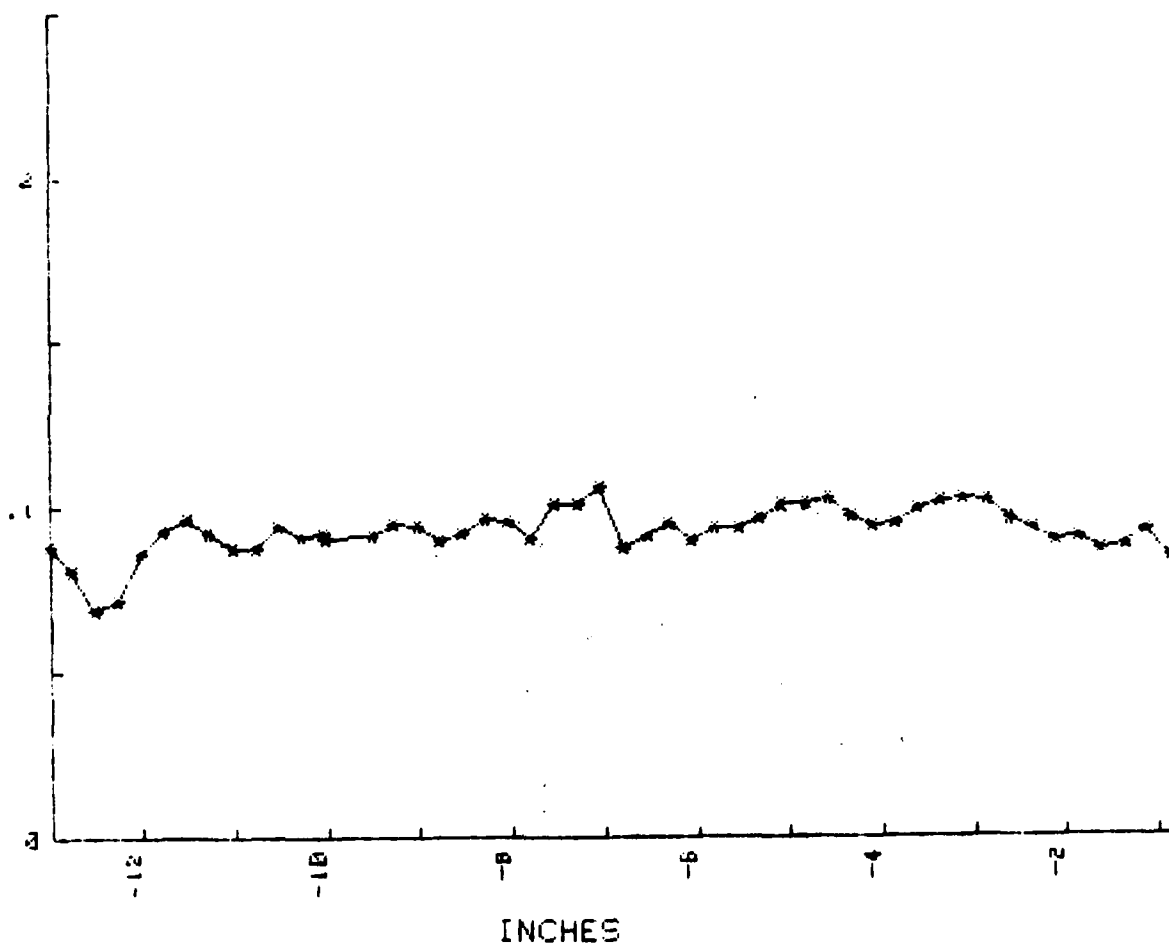


Fig. 17. Probe Survey Data at Midspan of Upper Plane
 End Walls at 30°, Points 1 to 50
 $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$

$P_{\text{plenum}} - P_t - Q_{\text{ref}}$
 POINTS 51 TO 100 UPPER PLANE 30 DEG

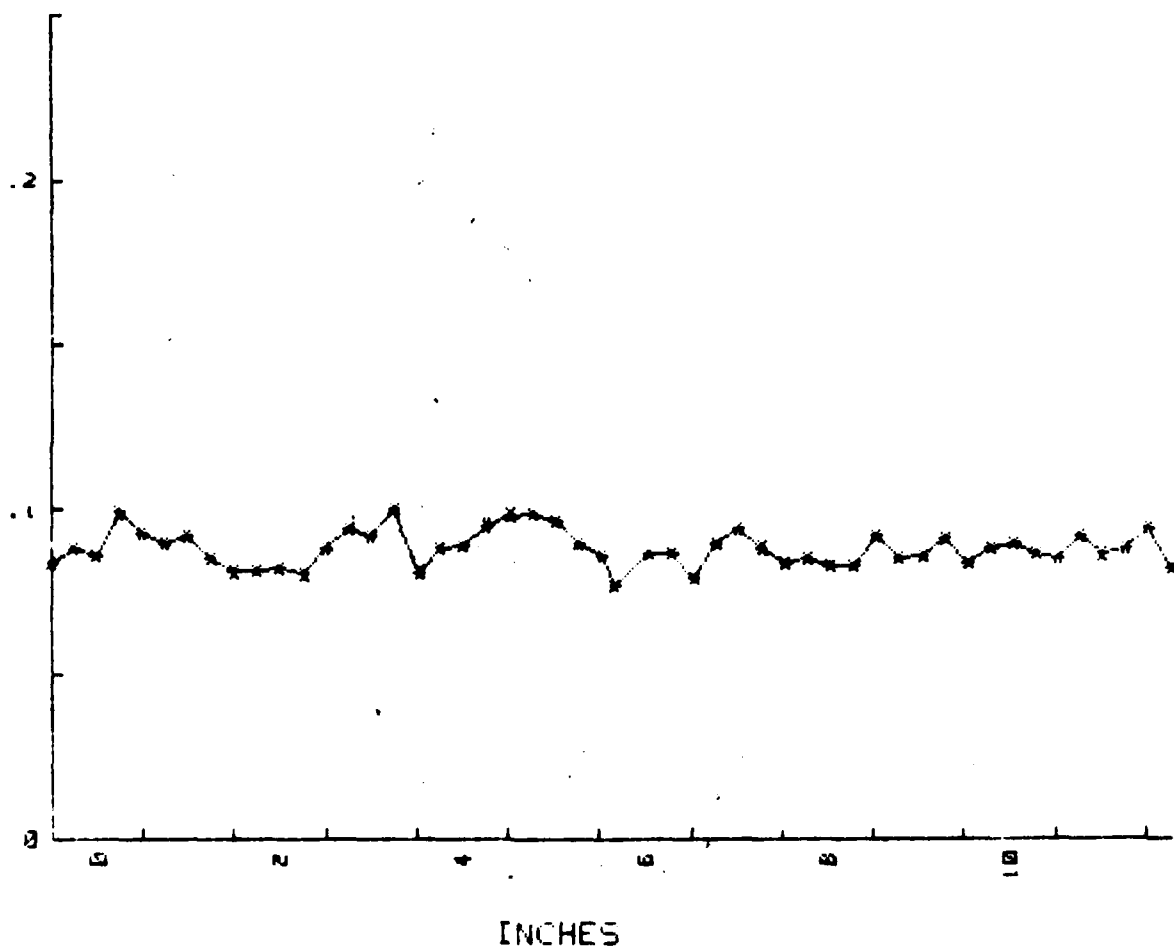


Fig. 18. Probe Survey Data at Midspan of Upper Plane
 End Walls at 30°, Points 51 to 100
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$(P_{\text{Plenum}} - P_t)/Q_{\text{ref}}$
POINTS 1 TO 50 LOWER PLANE 50 DEG

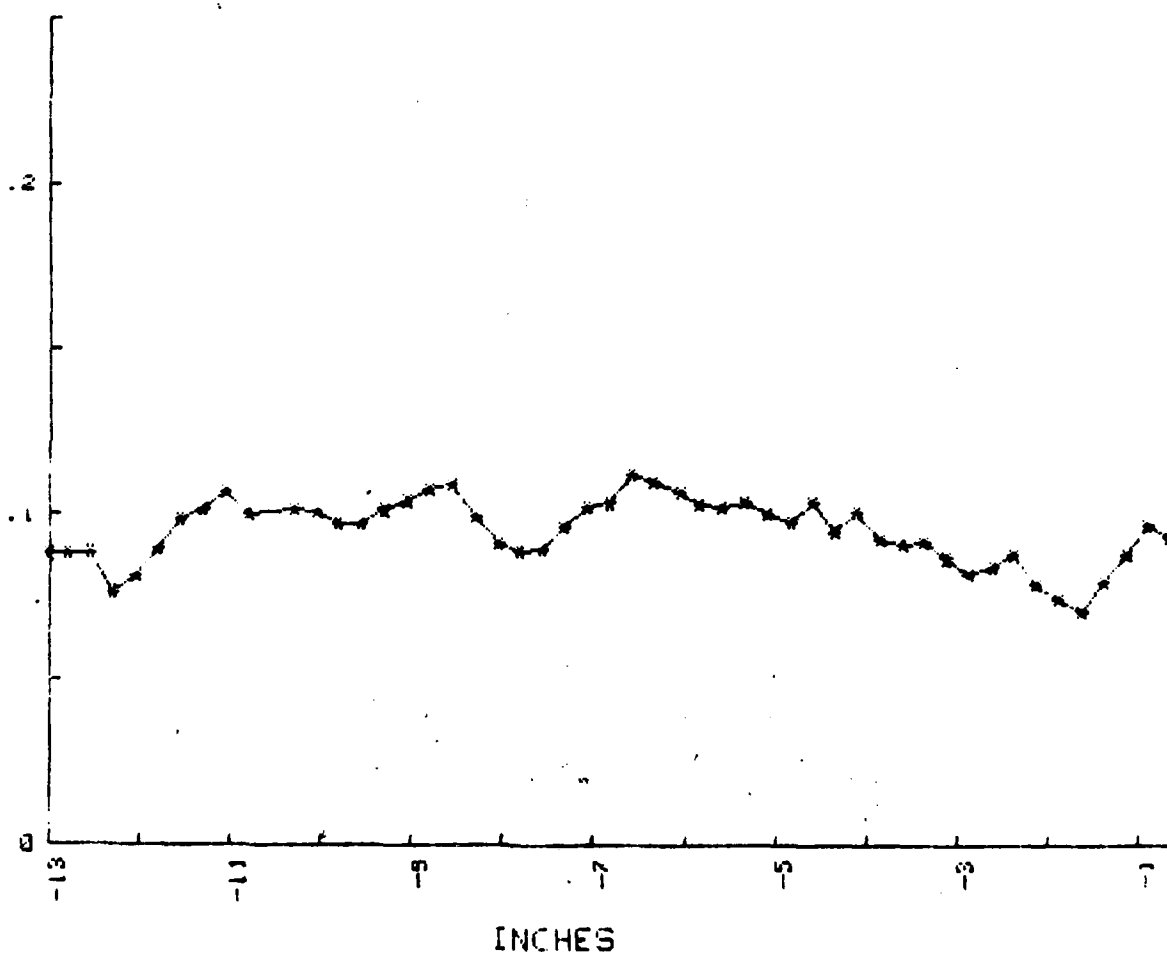


Fig. 19. Probe Survey Data at Midspan of Lower Plane
End Walls at 50°, Points 1 to 50
 $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$

$(P_{\text{plenum}} - P_t)/Q_{\text{ref}}$
 POINTS 51 TO 100 LOWER PLANE 50 DEG

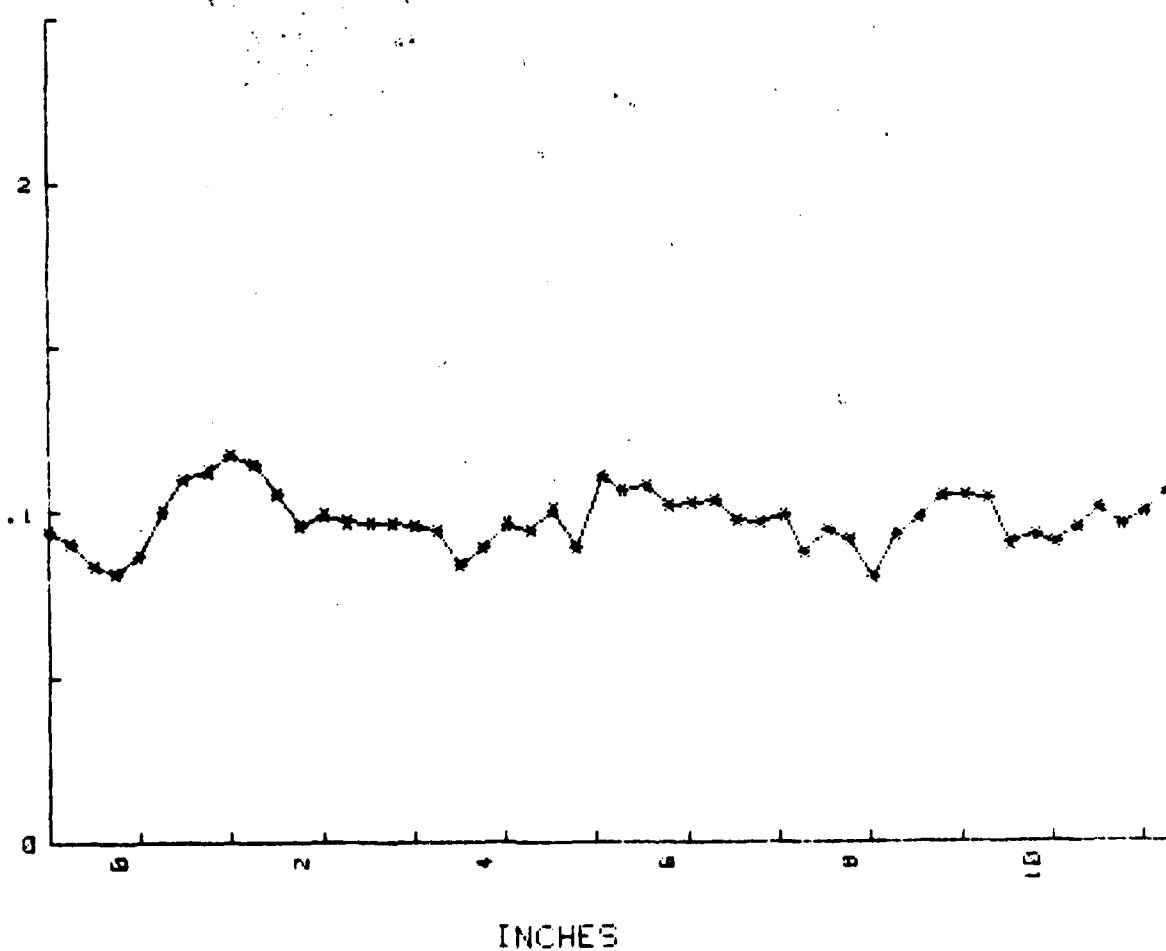


Fig. 20. Probe Survey Data at Midspan of Lower Plane
 End Walls at 50°, Points 51 to 100
 $(P_{\text{PLENUM}} - P_t)/Q_{\text{ref}}$

$P(\text{Plenum}) - P_t / Q_{\text{ref}}$
 50 POINTS UPPER PLANE 50 DEG

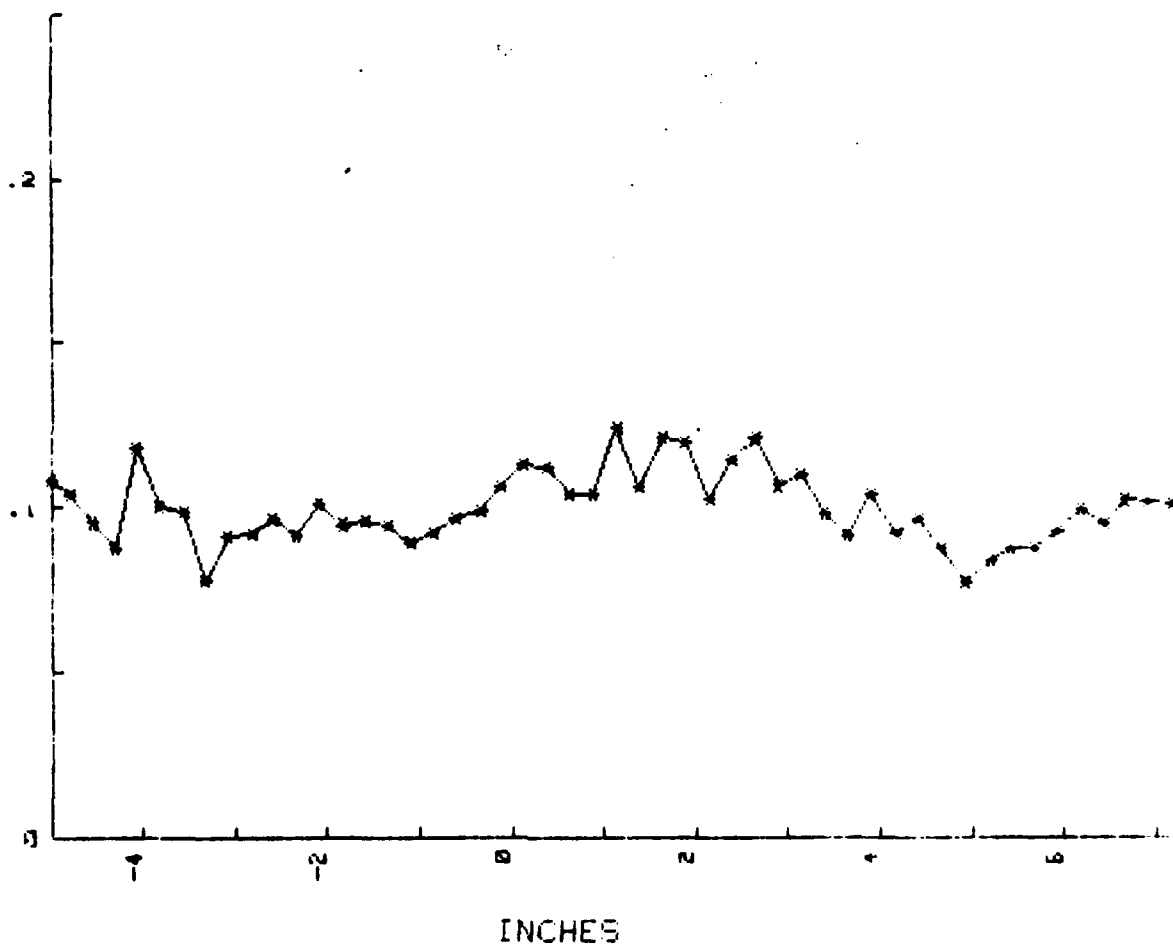


Fig. 21. Probe Survey Data at Midspan of Upper Plane
 End Walls at 50°
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{Plenum}) - P_t / Q_{\text{ref}}$
 POINTS 1 TO 50 LOWER PLANE 50 DEG
 * FIRST RUN + SECOND RUN

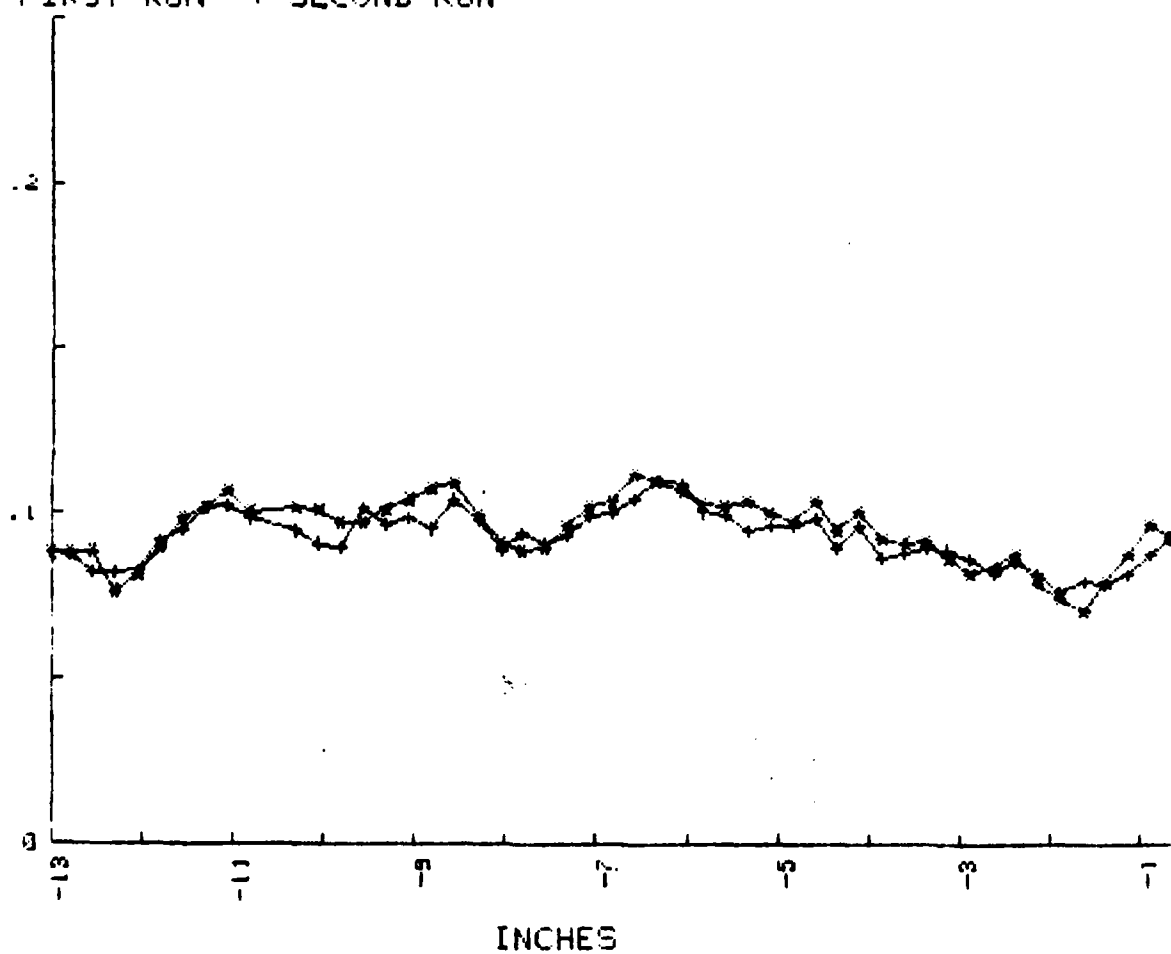


Fig. 22. Probe Survey Data at Midspan at Lower Plane
 End Walls at 50°, Two Runs, Points 1 to 50
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{Plenum}) - P_t / Q_{\text{ref}}$
 POINTS 51 TO 100 LOWER PLANE 50 DEG
 * FIRST RUN + SECOND RUN

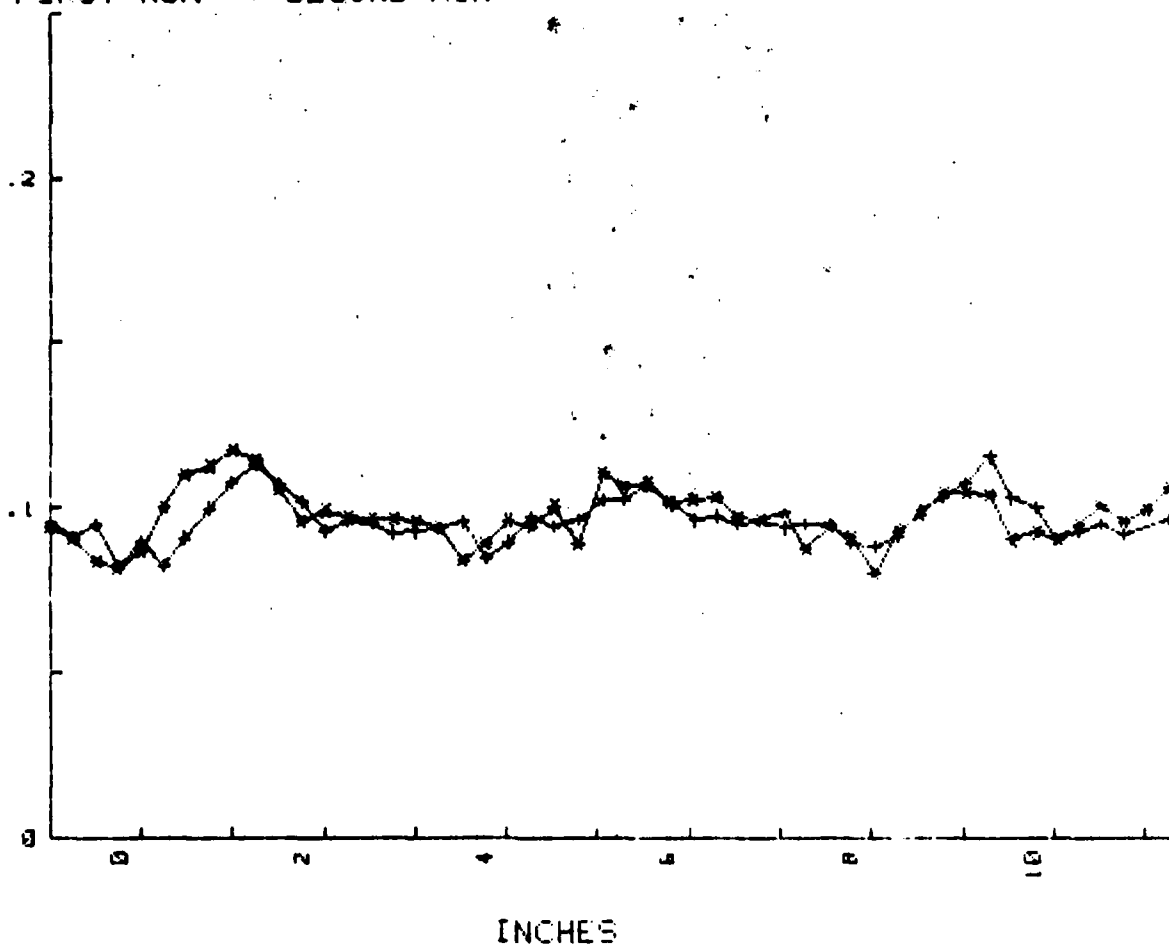


Fig. 23. Probe Survey Data at Midspan at Lower Plane
 End Walls at 50°, Two Runs, Points 51 to 100
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_t / Q_{\text{ref}}$

* FIRST RUN

+ SECOND RUN

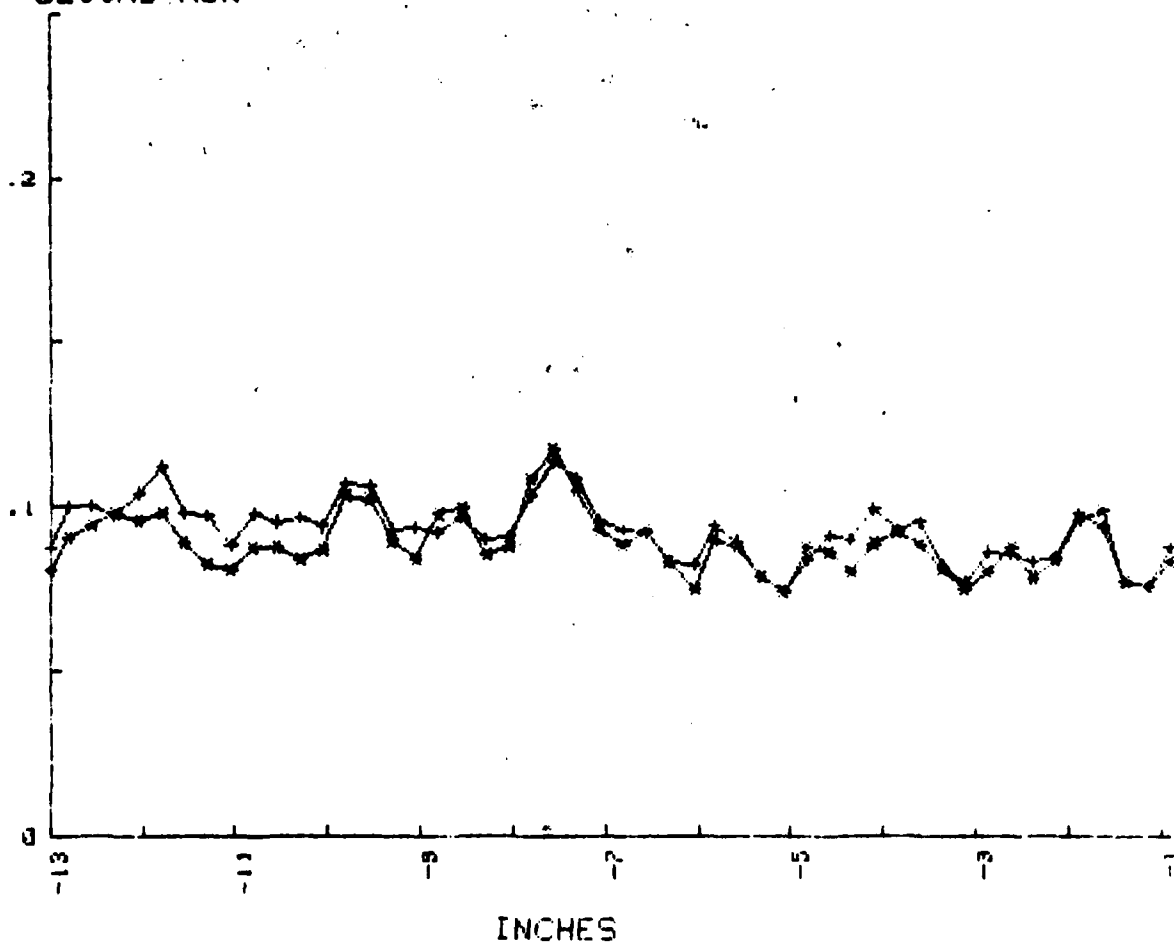


Fig. 24. Probe Survey Data at Midspan at Lower Plane
End Walls at 30°, Two Runs, Points 1 to 50
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P(\text{amb}) / Q_{\text{ref}}$
 10 INCHES LEFT OF CTR 30 DEG LOWER

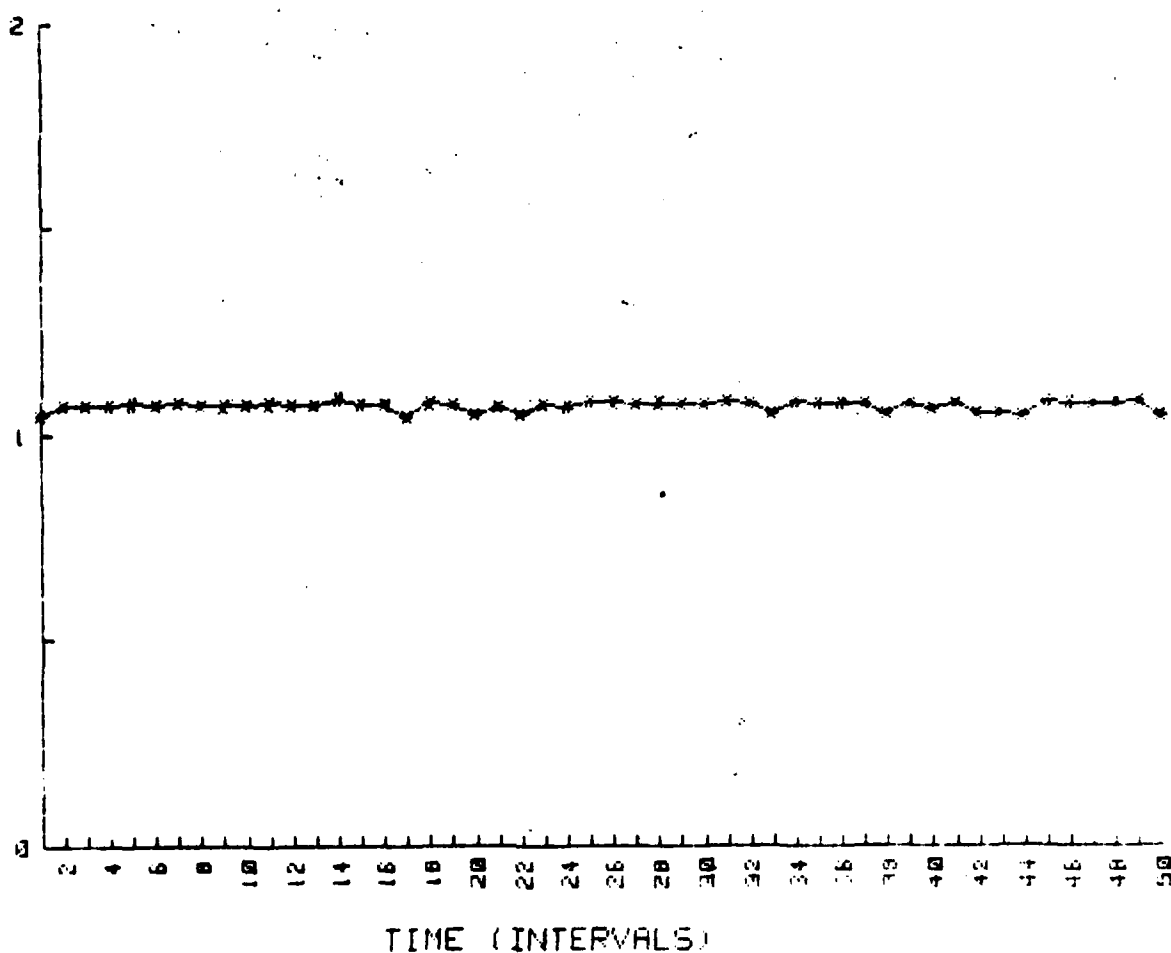


Fig. 25. Repetitive Samples with Fixed Probe Position
 (10" Left of CTR Midspan, End Walls 30°,
 Lower Plane $(P_{\text{PLENUM}} - P_{\text{AMB}}) / Q_{\text{ref}}$)

$P_t - P_{amb}) / Q_{ref}$
 10 INCHES LEFT OF CTR 30 DEG LOWER

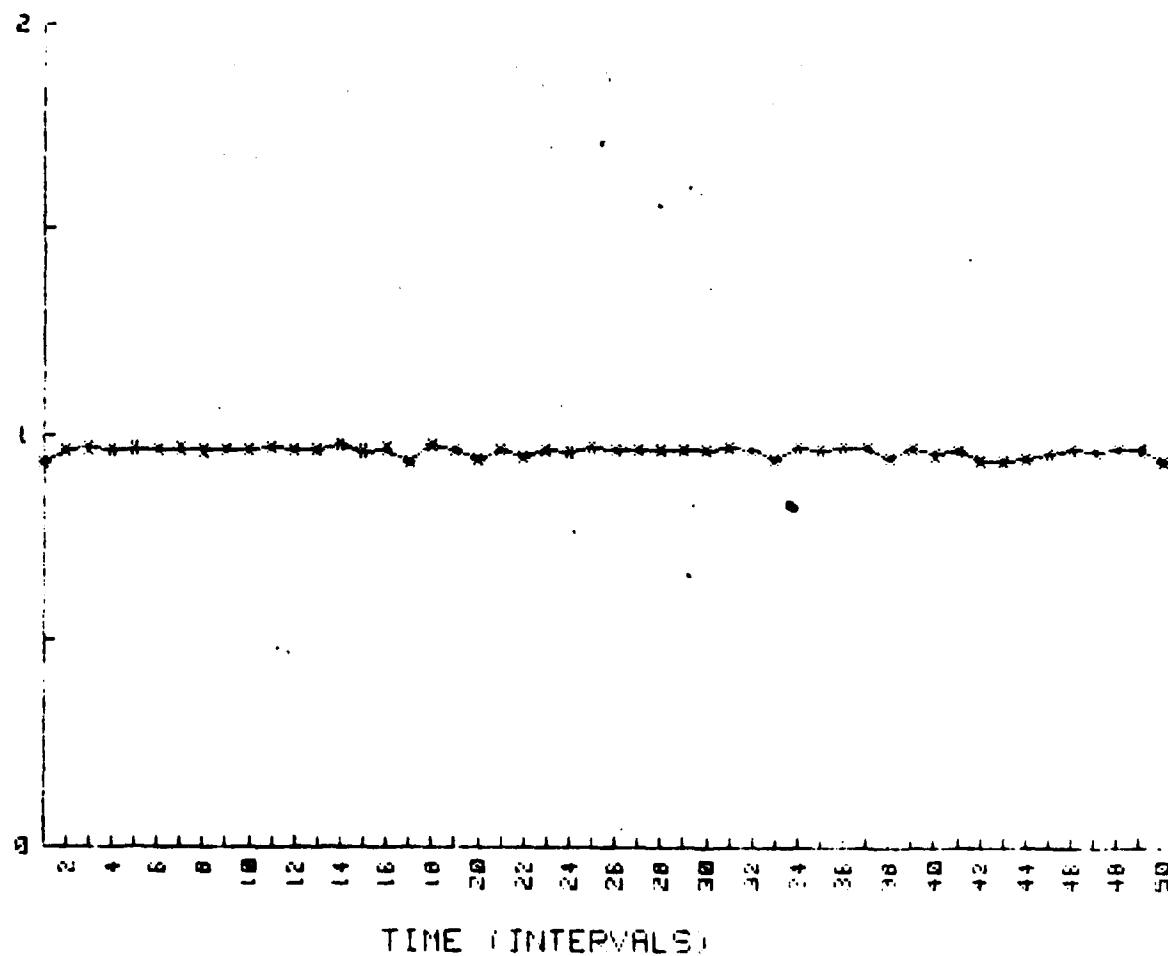


Fig. 26. Repetitive Samples with Fixed Probe Position
 (10" Left of CTR Midspan, End Walls at 30°,
 Lower Plane $(P_t - P_{AMB}) / Q_{ref}$)

$P(\text{plenum}) - P_1 / Q_{\text{ref}}$
 10 INCHES LEFT OF CTR 30 DEG LOWER

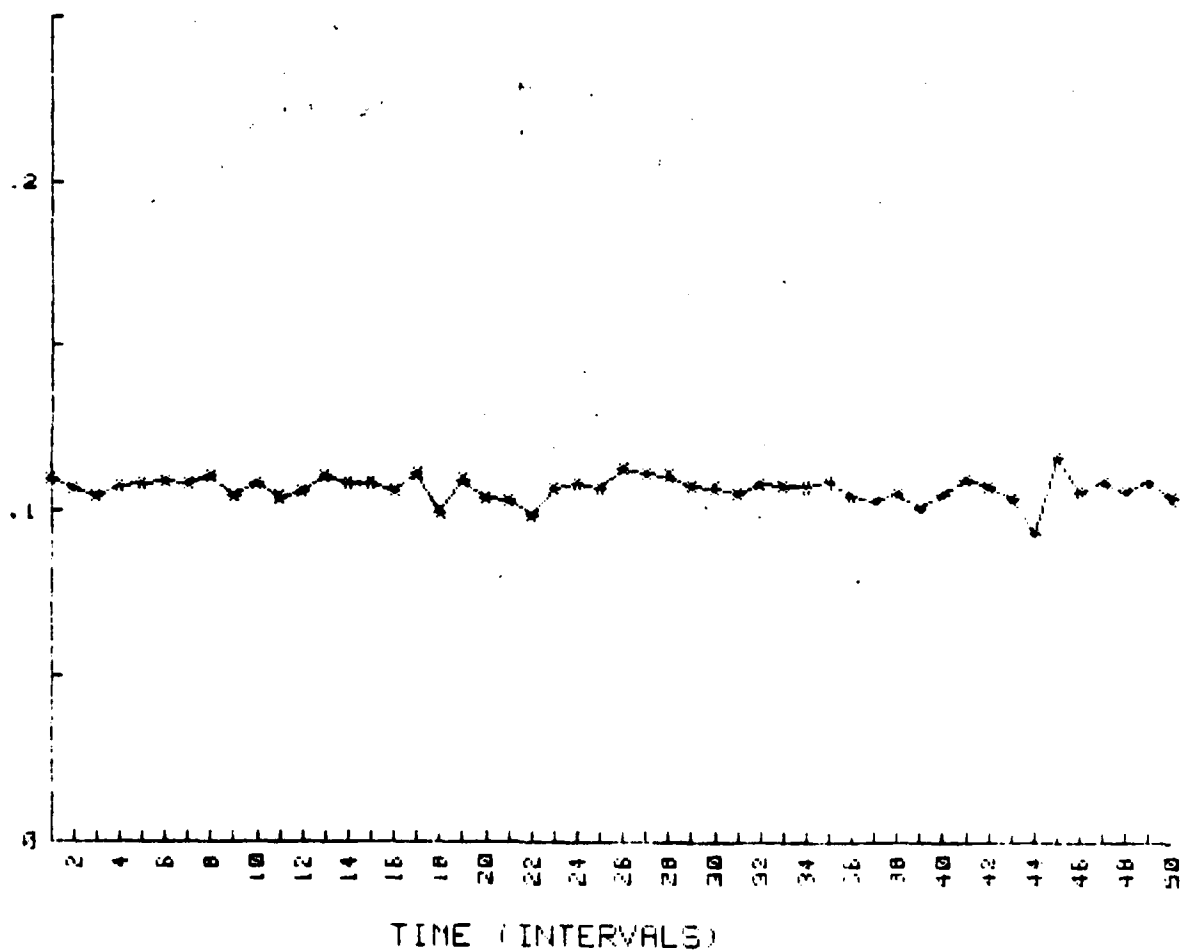


Fig. 27. Repetitive Samples with Fixed Probe Position
 (10" Left of CTR Midspan, End Walls at 30°,
 Lower Plane $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P(\text{amb}) / Q_{\text{ref}}$
 CENTER 30 DEG LOWER

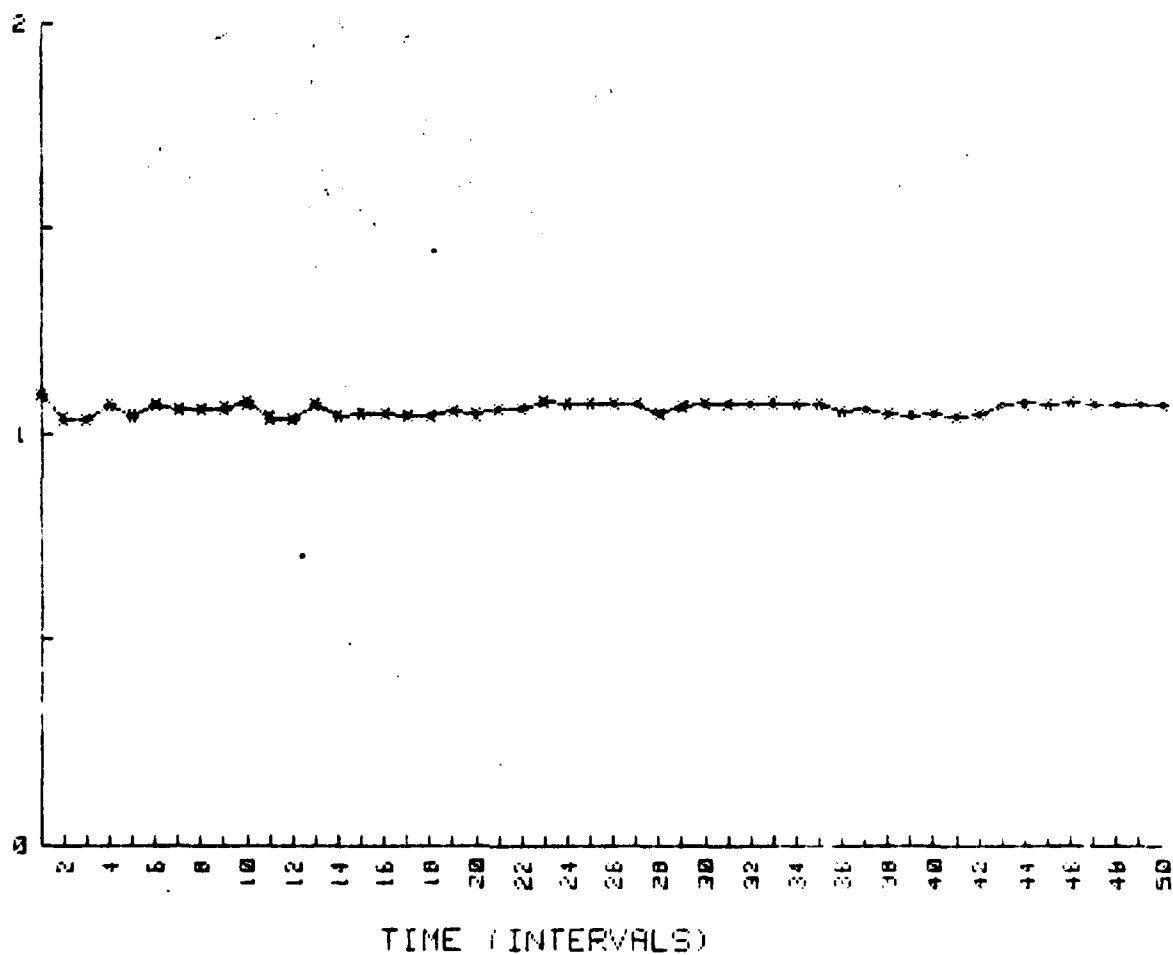


Fig. 28. Repetitive Samples with Fixed Probe Position
 (on Centerline at Midspan, End Walls at 30°,
 Lower Plane $(P_{\text{PLENUM}} - P_{\text{AMB}}) / Q_{\text{ref}}$)

$P_t - P_{AMB} / Q_{ref}$
 CENTER 30 DEG LOWER

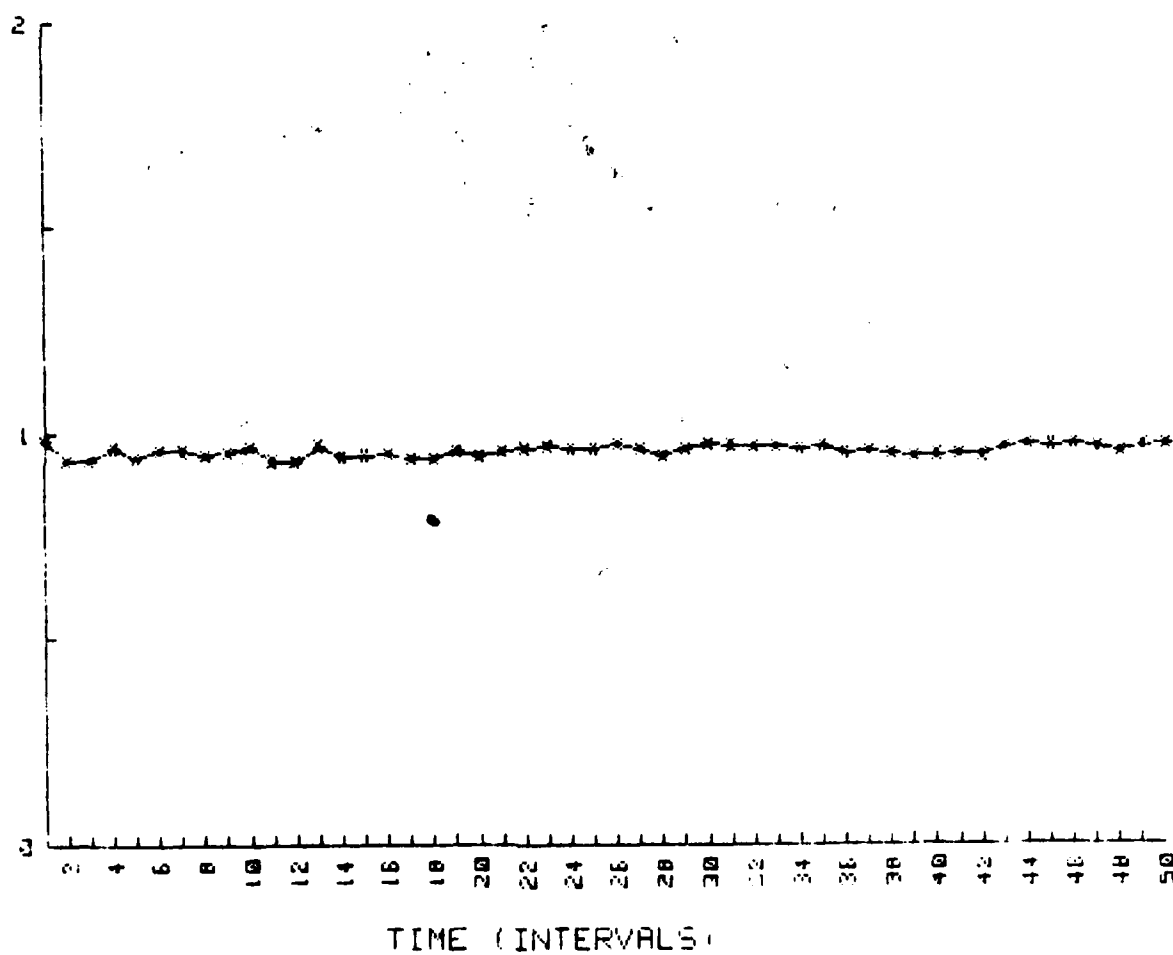


Fig. 29. Repetitive Samples with Fixed Probe Position
 (on Centerline at Midspan, End Walls at 30°,
 Lower Plane $(P_t - P_{AMB}) / Q_{ref}$)

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
 CENTER 30 DEG LOWER

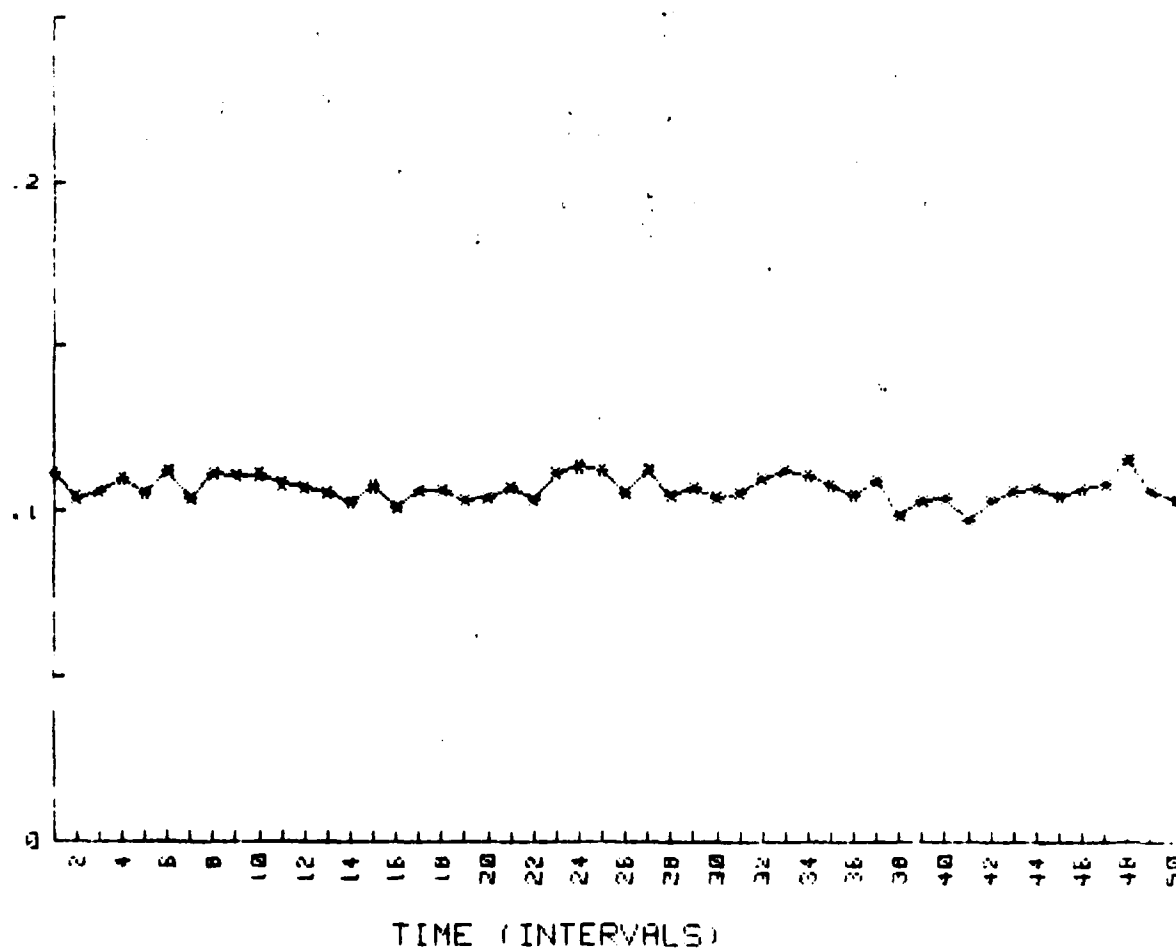


Fig. 30. Repetitive Samples with Fixed Probe Position
 (on Centerline at Midspan, End Walls at 30°,
 Lower Plane $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$)

$P(\text{plenum}) - P(\text{amb}) / Q_{\text{ref}}$
 10 INCHES RT OF CTR 30 DEG LOWER

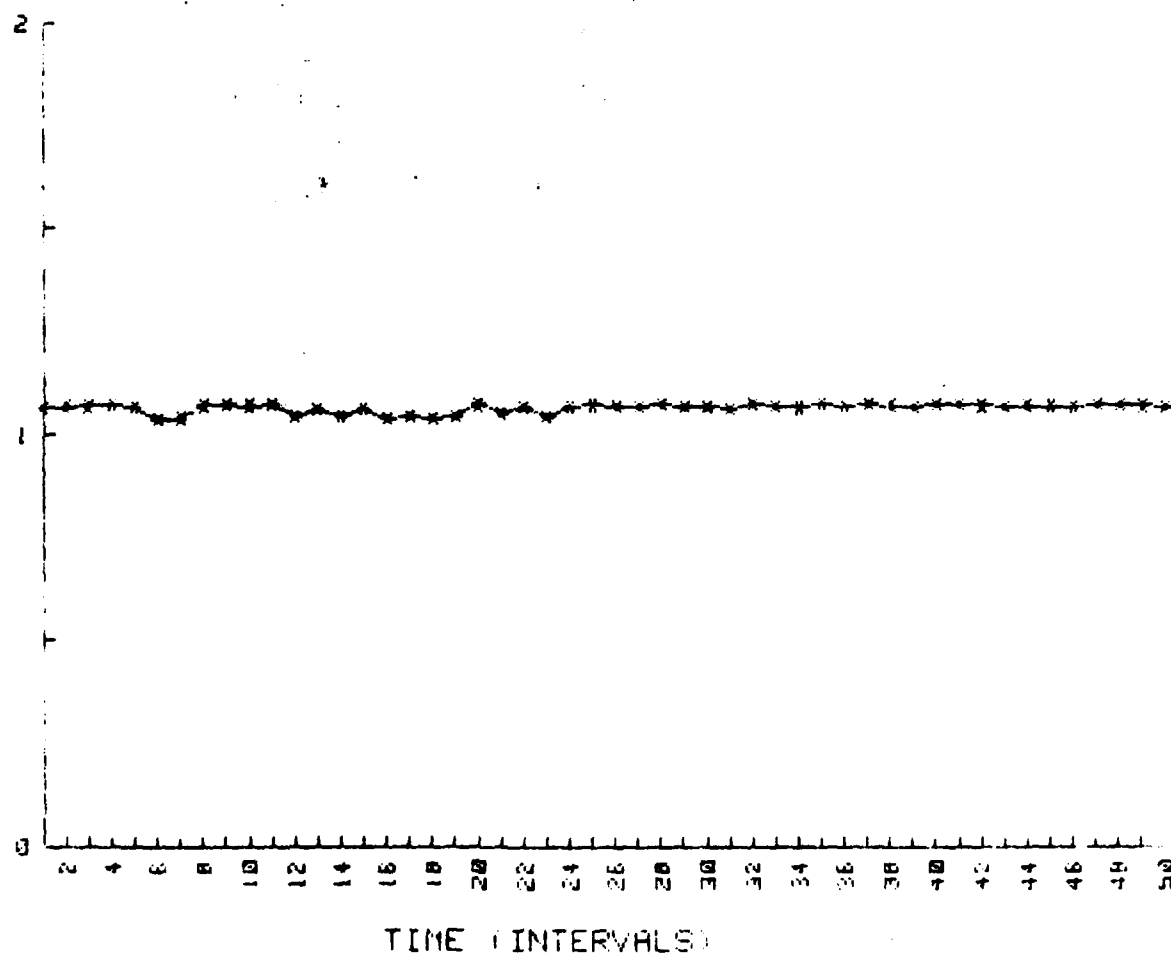


Fig. 31. Repetitive Samples with Fixed Probe Position
 (10" Right of CTR Midspan, End Walls at 30°,
 Lower Plane $(P_{\text{PLENUM}} - P_{\text{AMB}}) / Q_{\text{ref}}$)

$P_t - P_{amb} / Q_{ref}$
 10 INCHES RT OF CTR 30 DEG LOWER

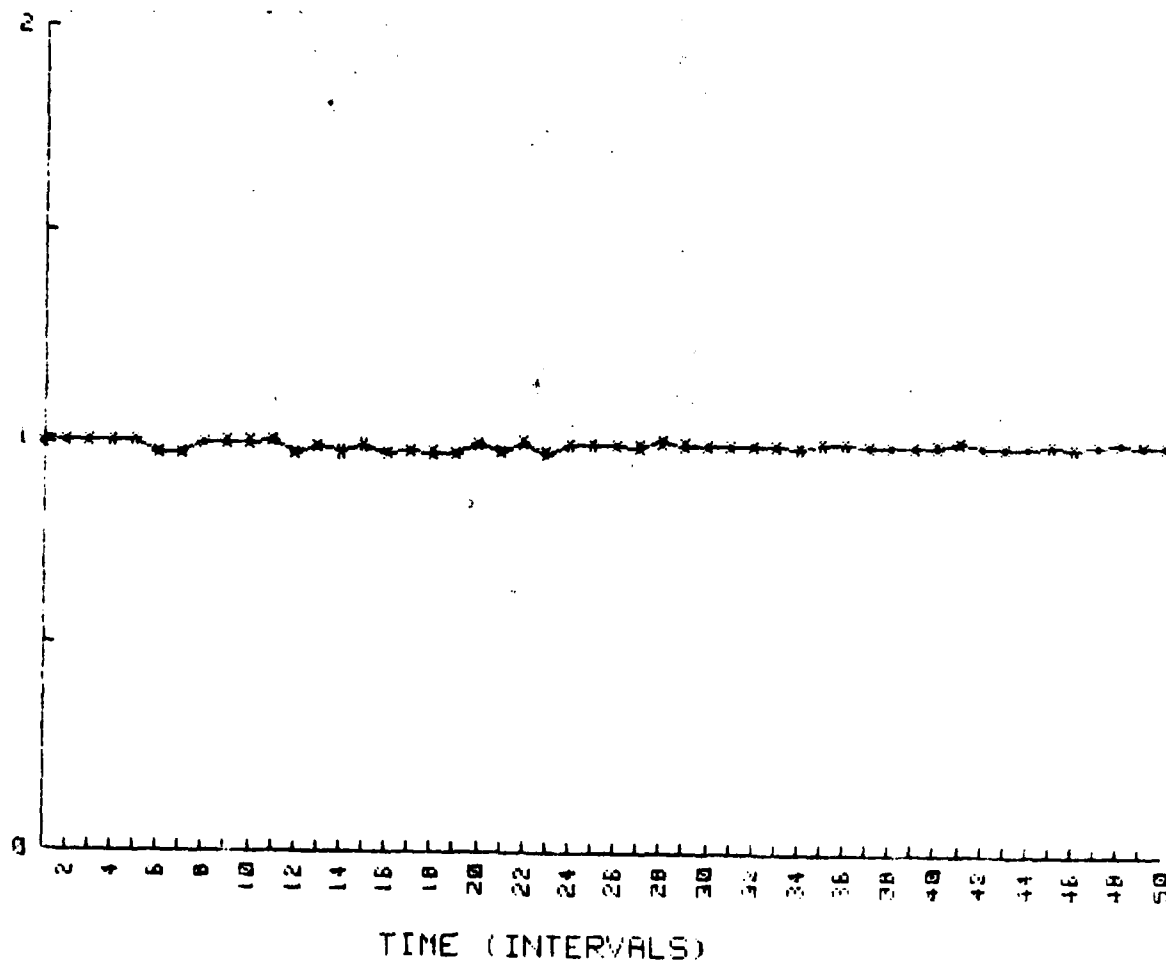


Fig. 32. Repetitive Samples with Fixed Probe Position
 (10" Right of CTR Midspan, End Walls at 30°,
 Lower Plane $(P_t - P_{AMB}) / Q_{ref}$)

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
 10 INCHES RT OF CTR 30 DEG LOWER

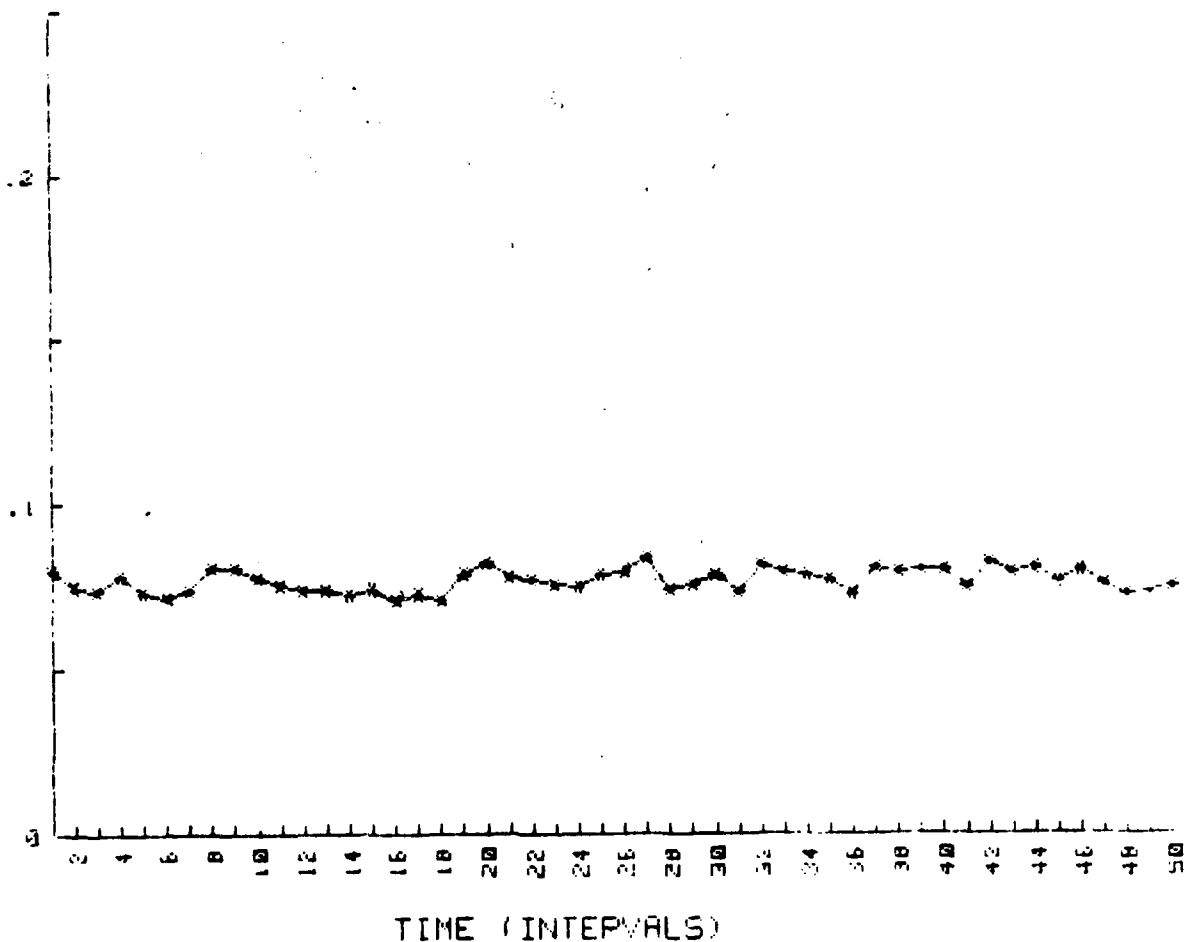


Fig. 33. Repetitive Samples with Fixed Probe Position
 (10" Right of CTR Midspan, End Walls at 30°,
 Lower Plane ($P_{\text{PLENUM}} - P_t$) / Q_{ref})

$P(\text{plenum}) - P_1 / Q_{\text{ref}}$
POINTS 1 TO 50 LOWER PLANE 35 DEG TEMP SCREEN

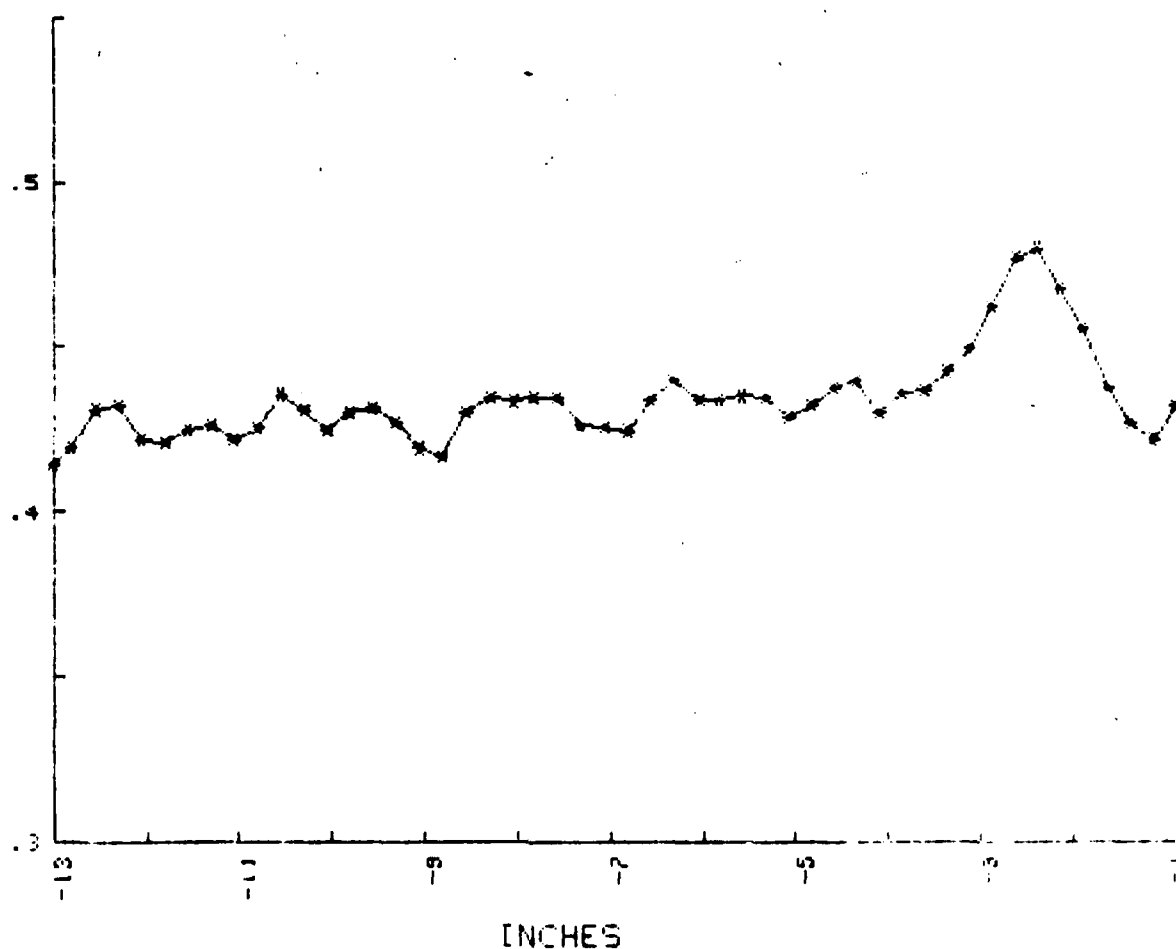


Fig. 34. Probe Survey Data at Midspan, Lower Plane
16 Mesh Screen, Walls at 35°, Points 1 to 50
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
 POINTS 51 TO 100 LOWER PLANE 35 DEG TEMP SCREEN

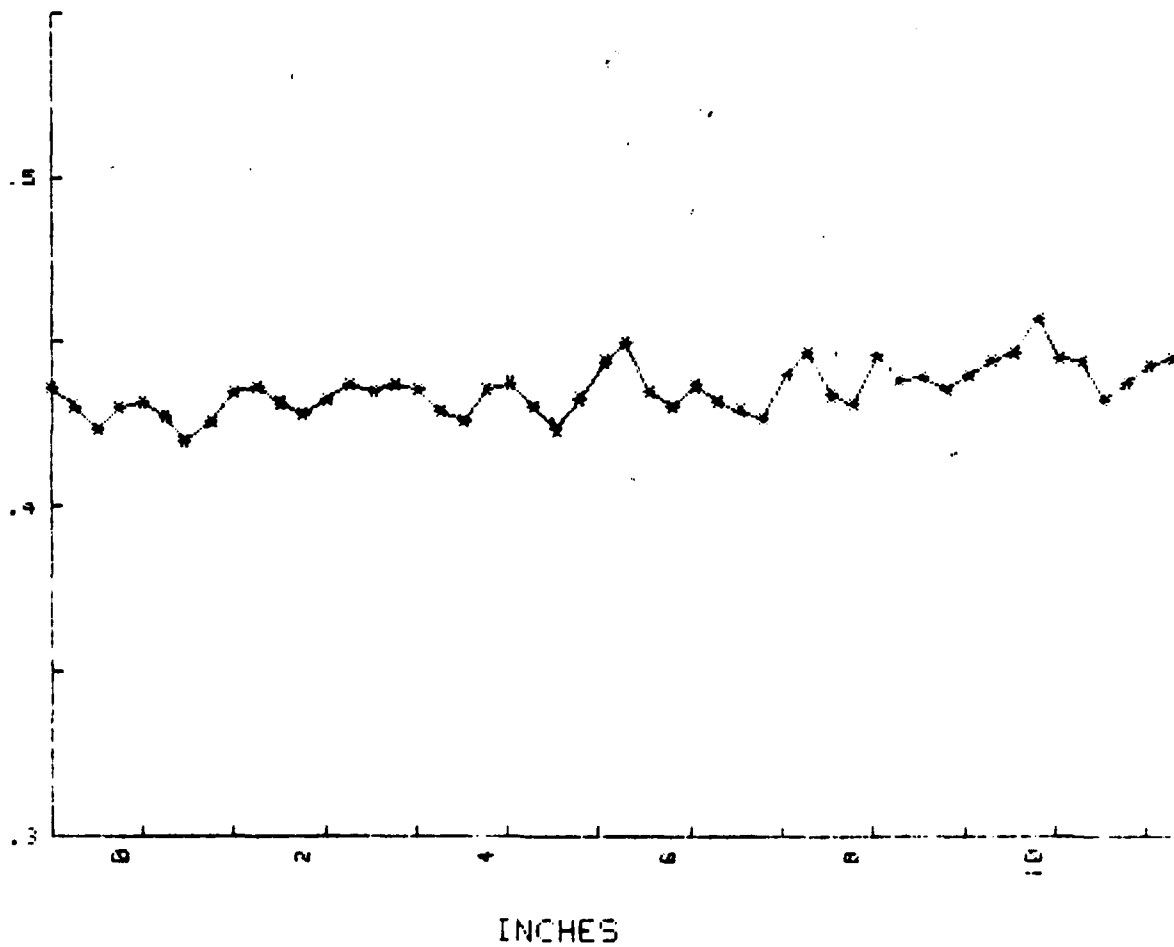


Fig. 35. Probe Survey Data at Midspan, Lower Plane
 16 Mesh Screen, Walls at 35°, Points 51 to 100
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
 POINTS 1 TO 50 UPPER PLANE 35 DEG TEMP SCREEN

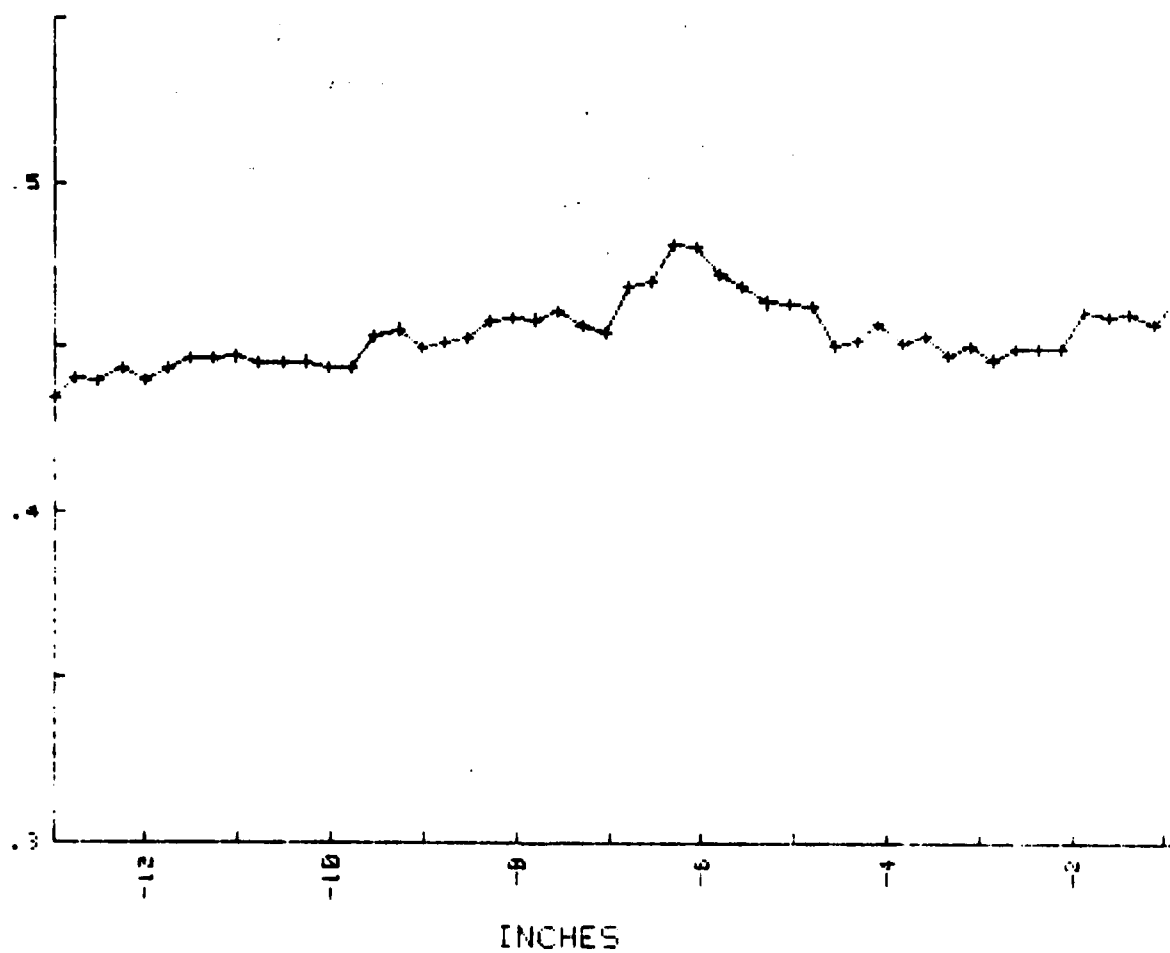


Fig. 36. Probe Survey Data at Midspan, Upper Plane
 16 Mesh Screen, Walls at 35°, Points 1 to 50
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
 POINTS 51 TO 100 UPPER PLANE 35 DEG TEMP SCREEN

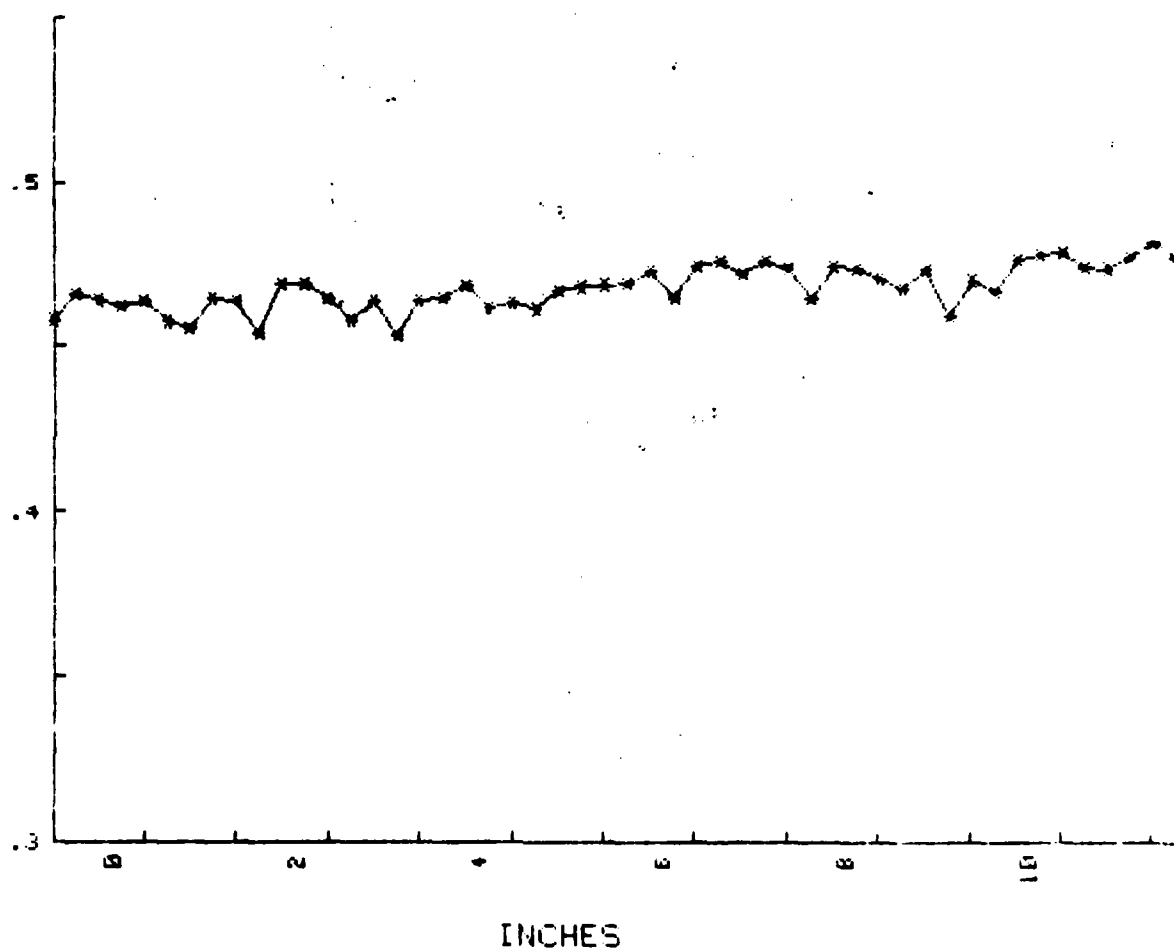


Fig. 37. Probe Survey Data at Midspan, Upper Plane
 16 Mesh Screen, Walls at 35°, Points 51 to 100
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
 10 in. LEFT OF CTR SPAN TRAVERSE
 35 DEG UPPER PLANE TEMP SCREEN

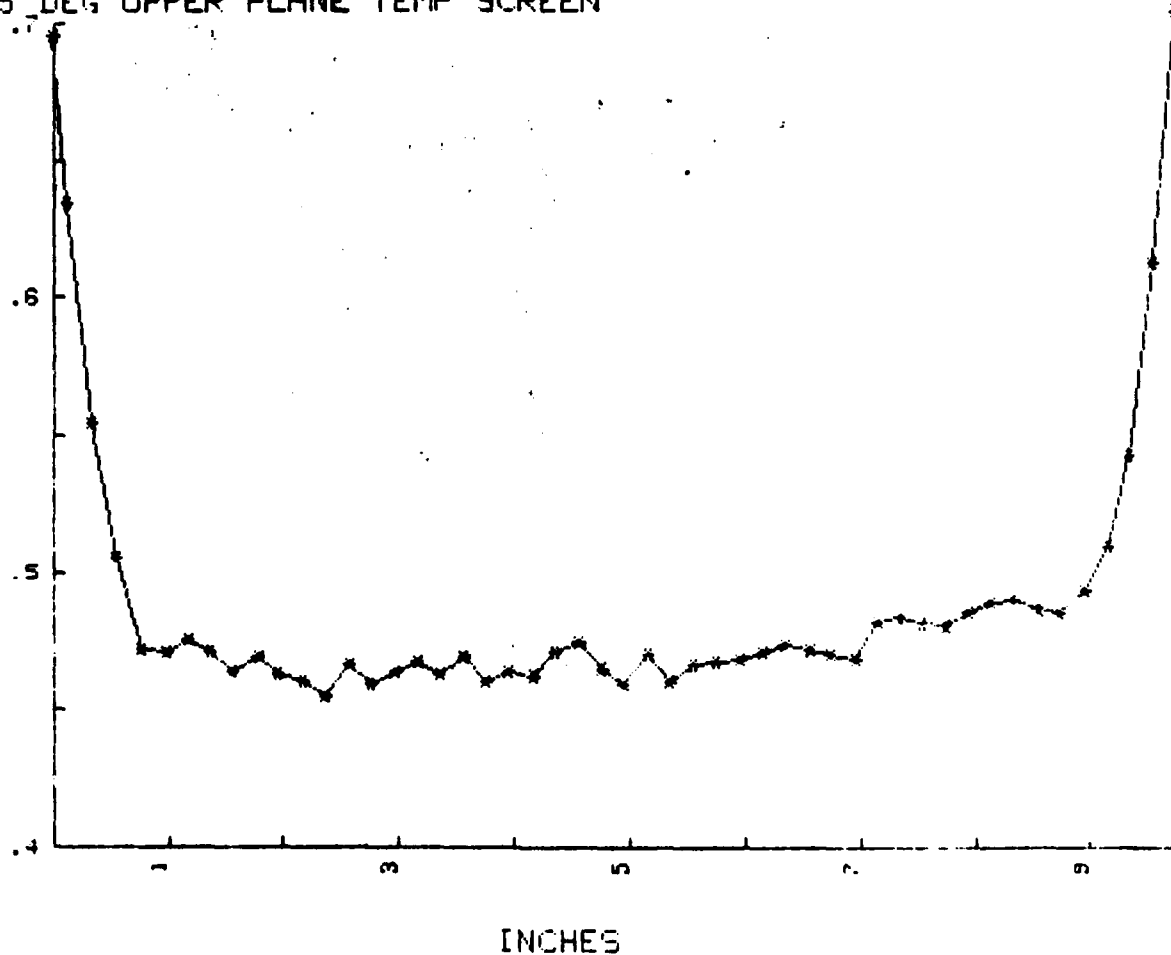


Fig. 38. Probe Survey Data Span Traverse, Lower Plane
 16 Mesh Screen, Walls at 35°, 10" Left of CTR
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
 CTR SPAN TRAVERSE
 35 DEG UPPER PLANE TEMP SCREEN

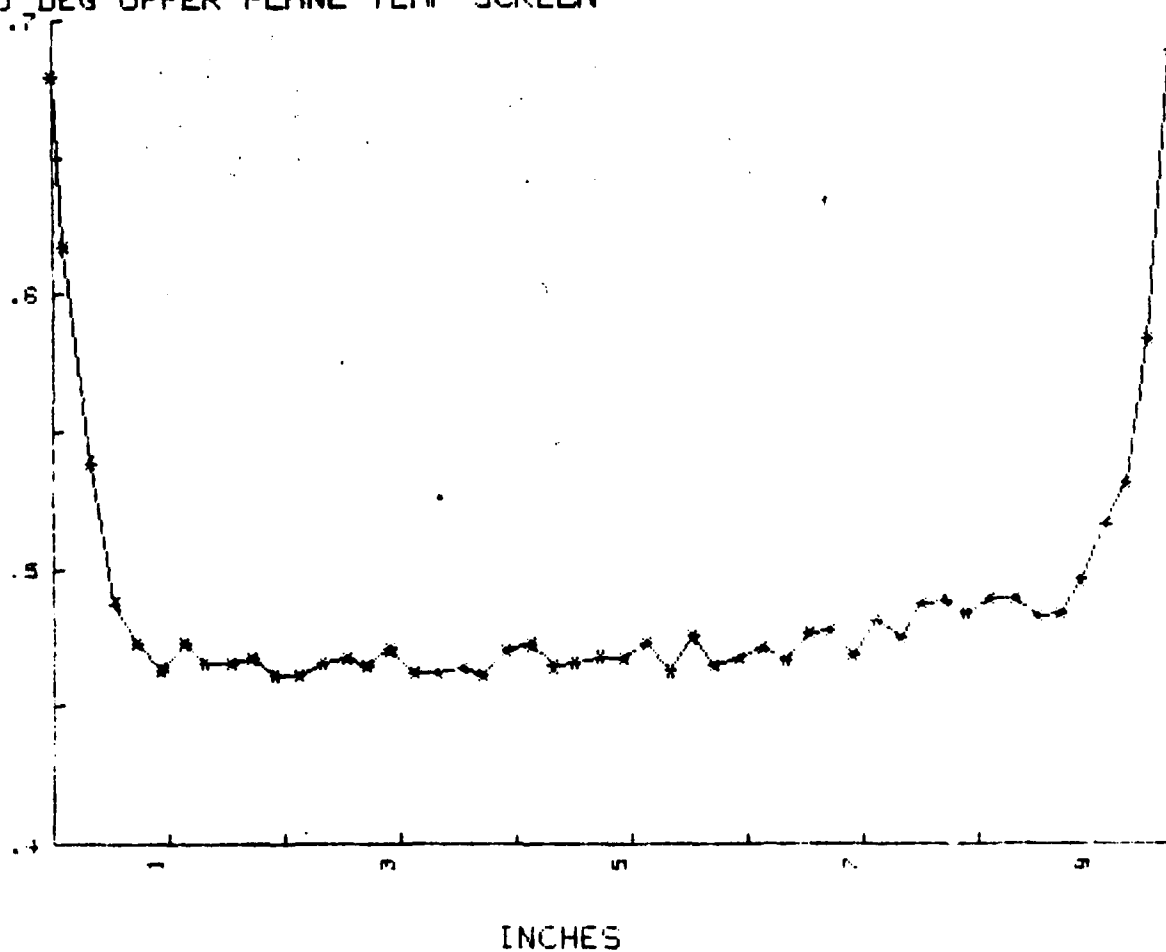


Fig. 39. Probe Survey Data Span Traverse, Lower Plane
 16 Mesh Screen, Walls at 35°, Center of Test
 Section $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
 10 in. RT OF CTR SPAN TRAVERSE
 35 DEG UPPER PLANE TEMP SCREEN

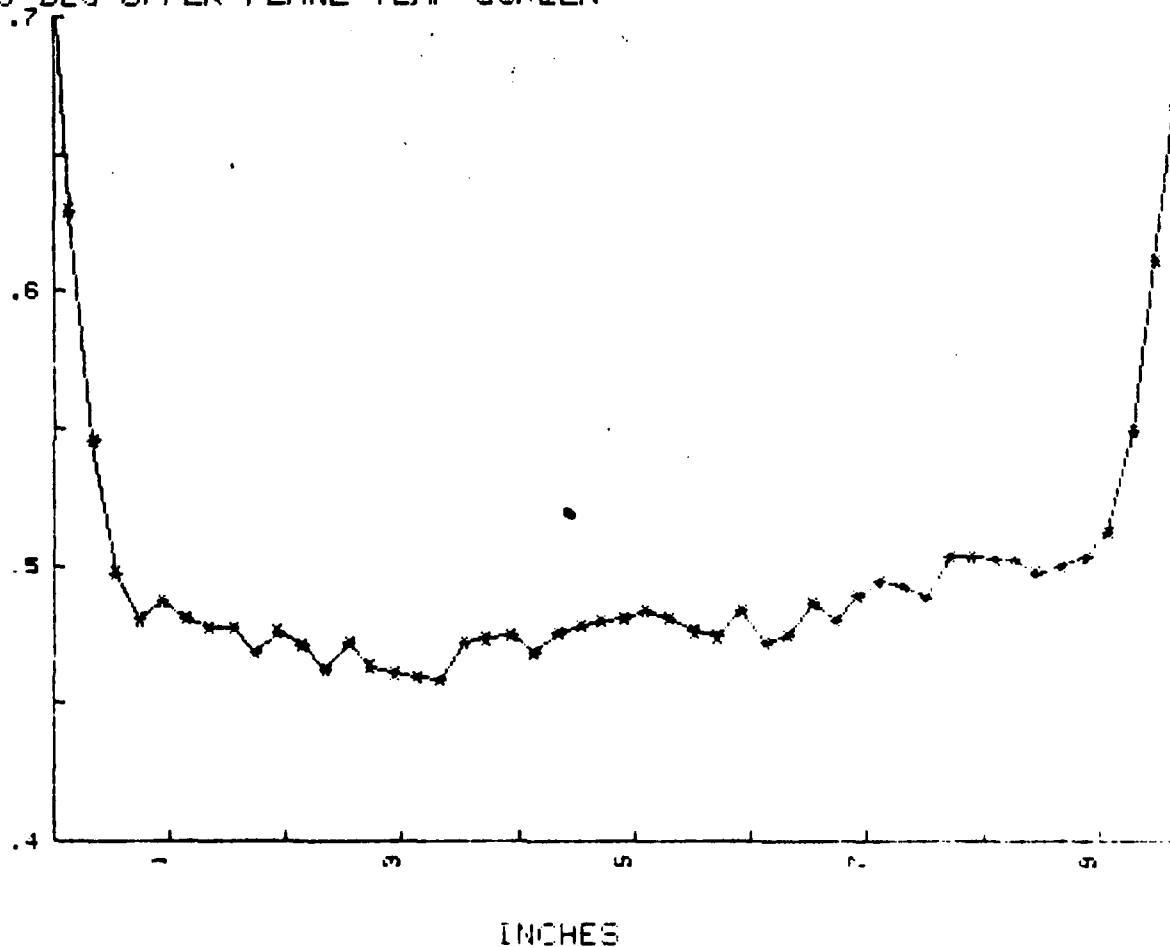


Fig. 40. Probe Survey Data Span Traverse, Lower Plane
 16 Mesh Screen, Walls at 35°, Center of Test
 Section $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_1 / Q_{\text{ref}}$
 35 DEG LOWER PLANE 2 SCREENS
 POINTS 1 TO 50

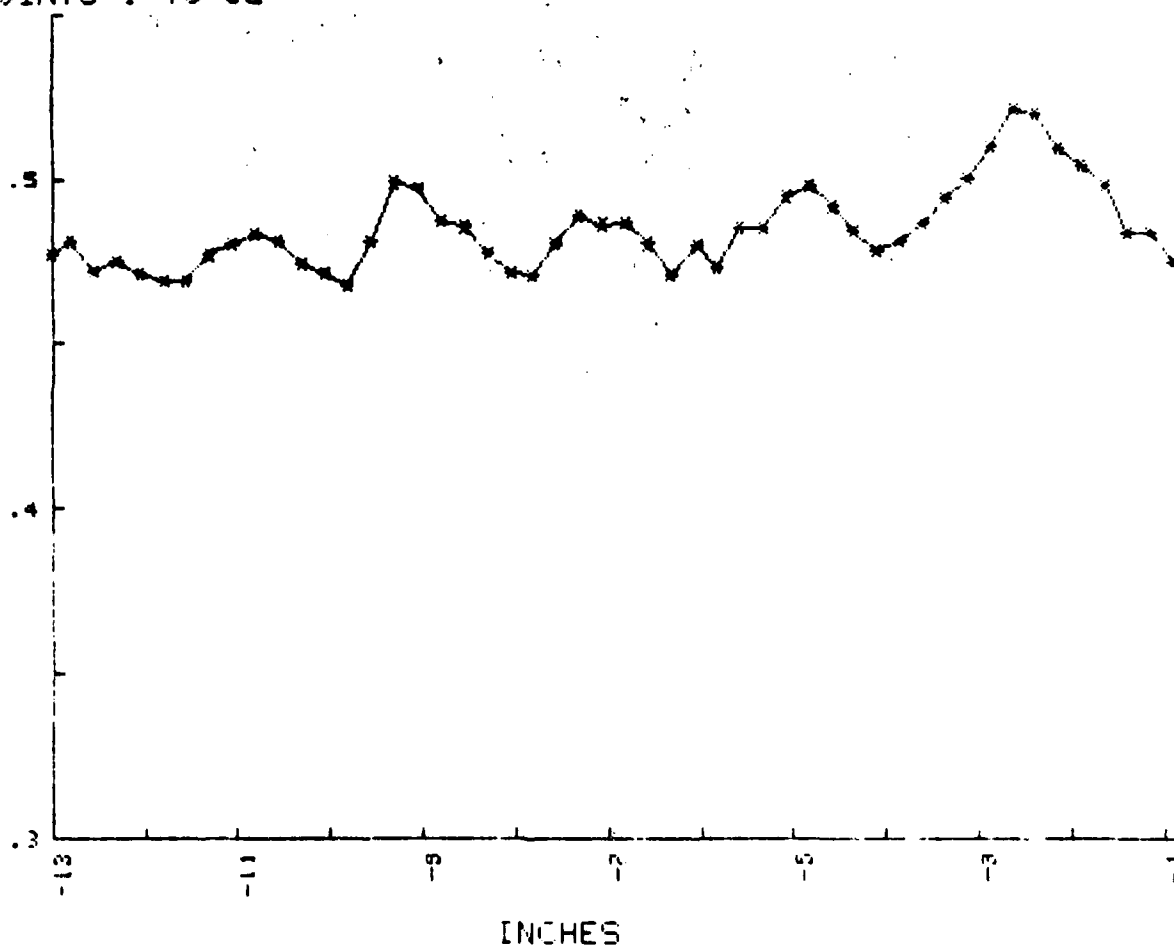


Fig. 41. Probe Survey Data at Midspan, Lower Plane
 16 Mesh and 2 Mesh, Walls at 35°, Points 1 to 50
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

OD-A120 456

PRELIMINARY MEASUREMENTS AND CODE CALCULATIONS OF FLOW
THROUGH A CASCADE OF DCA BLADING AT A SOLIDITY OF 167
(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA W D MOLLOY

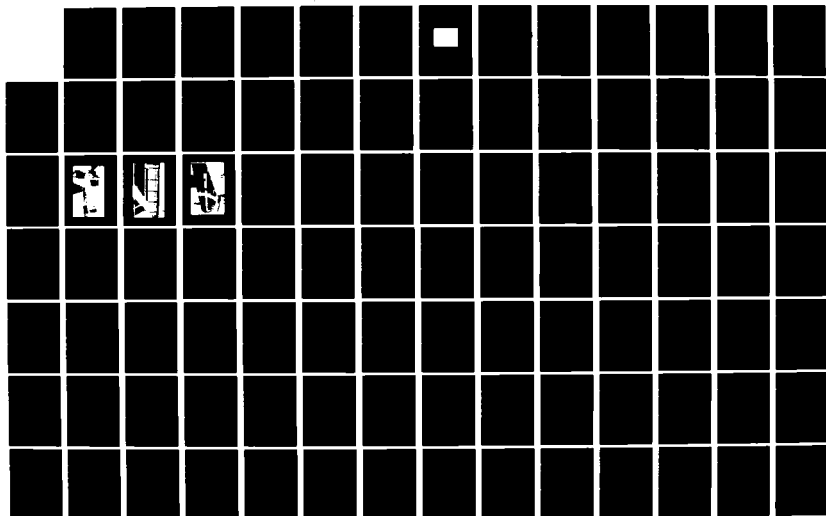
2/3

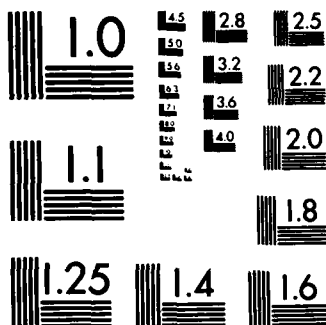
UNCLASSIFIED

JUN 82

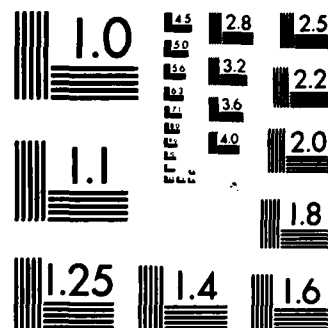
F/G 21/5

NL

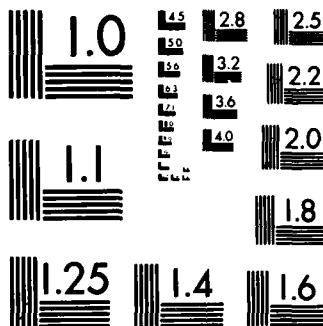




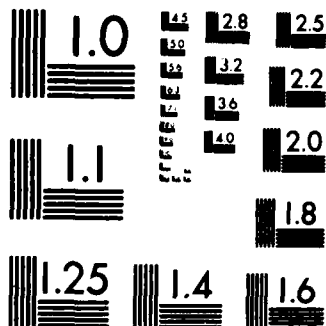
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



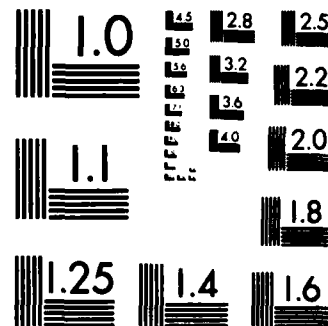
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
 35 DEG LOWER PLANE 2 SCREENS
 POINTS 51 TO 100

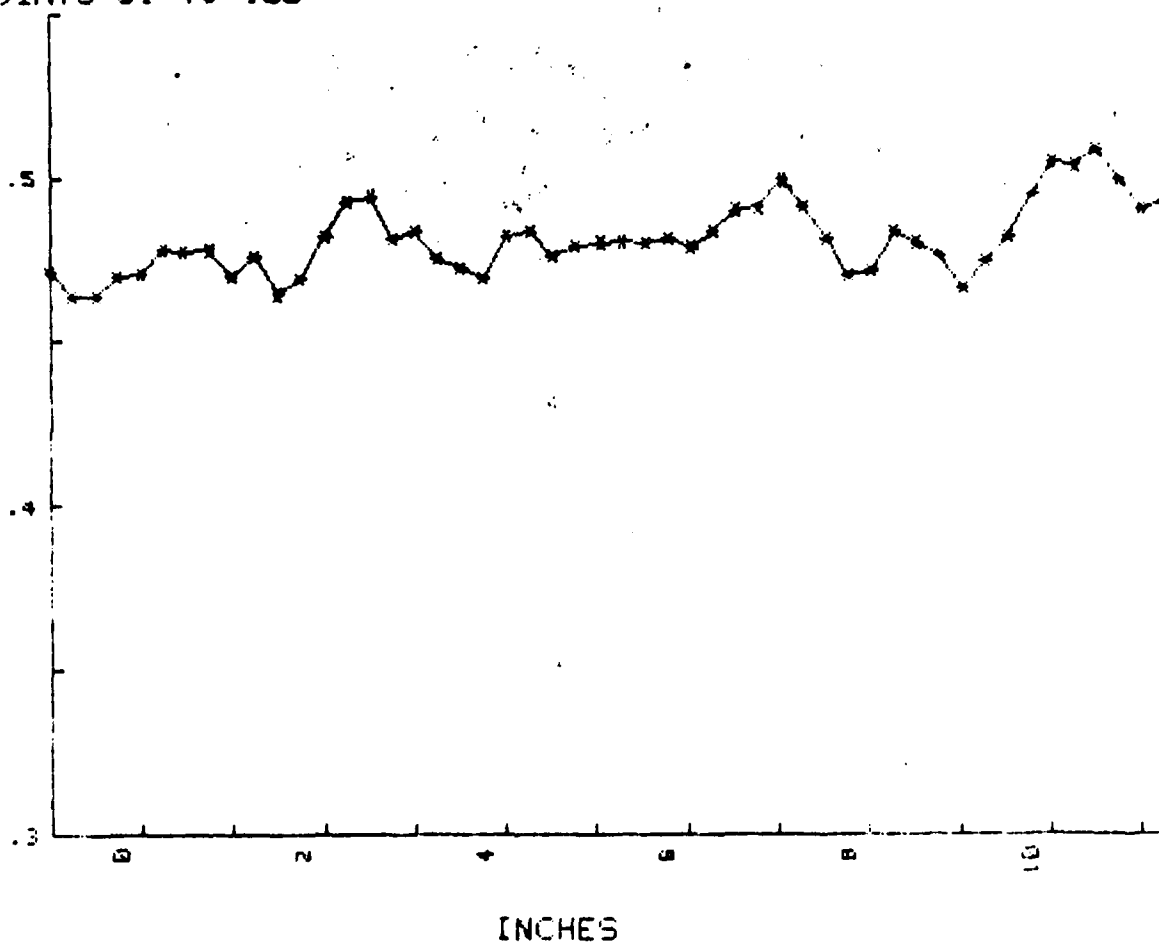


Fig. 42. Probe Survey Data at Midspan, Lower Plane
 16 Mesh and 2 Mesh, Walls at 35°, Points 51 to 100
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
 PT 1 TO 50 LOWER PLANE 35 DEG
 4 MESH .041 WIRE SCREEN

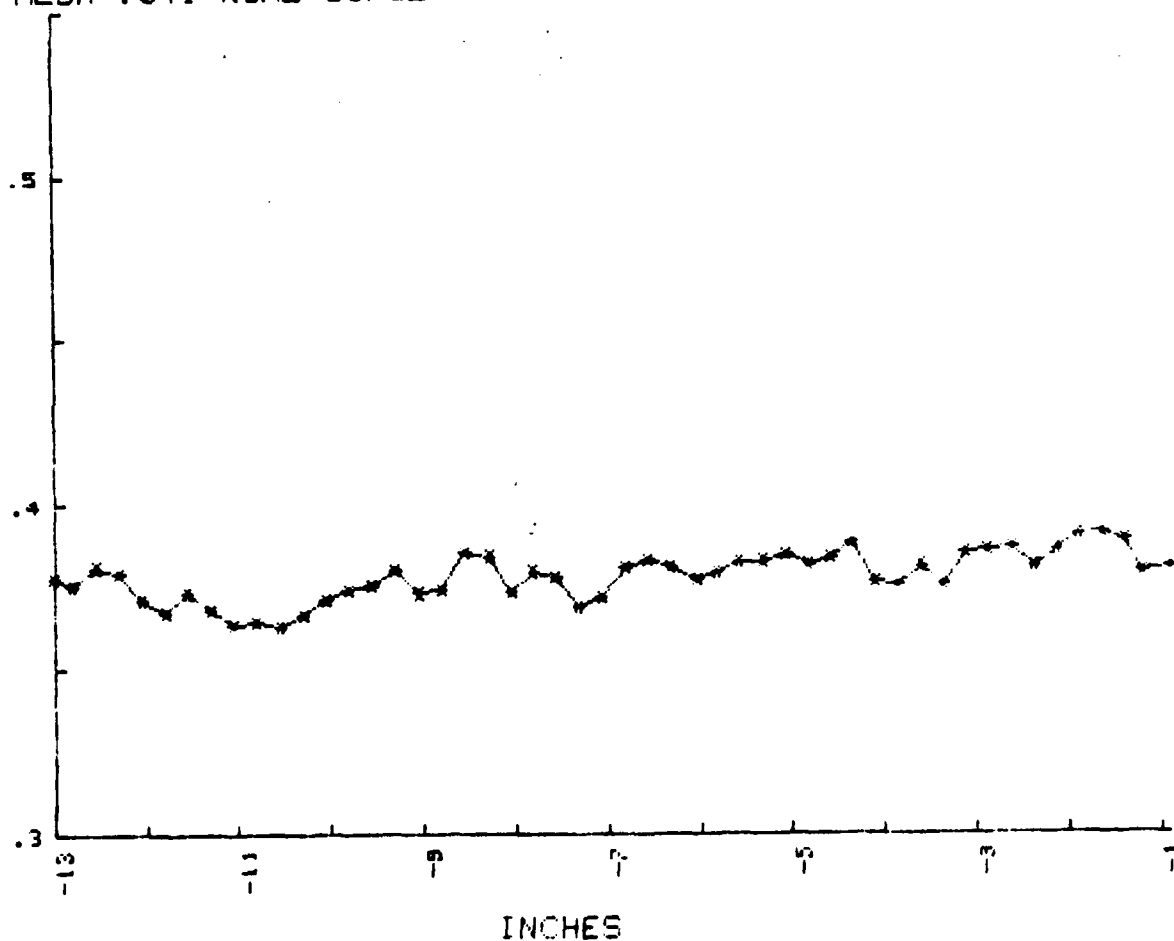


Fig. 43. Probe Survey Data at Midspan, Lower Plane
 4 Mesh Screen, Walls at 35°, Points 1 to 50
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
 PT 51 TO 100 LOWER PLANE 35 DEG
 4 MESH .041 WIRE SCREEN

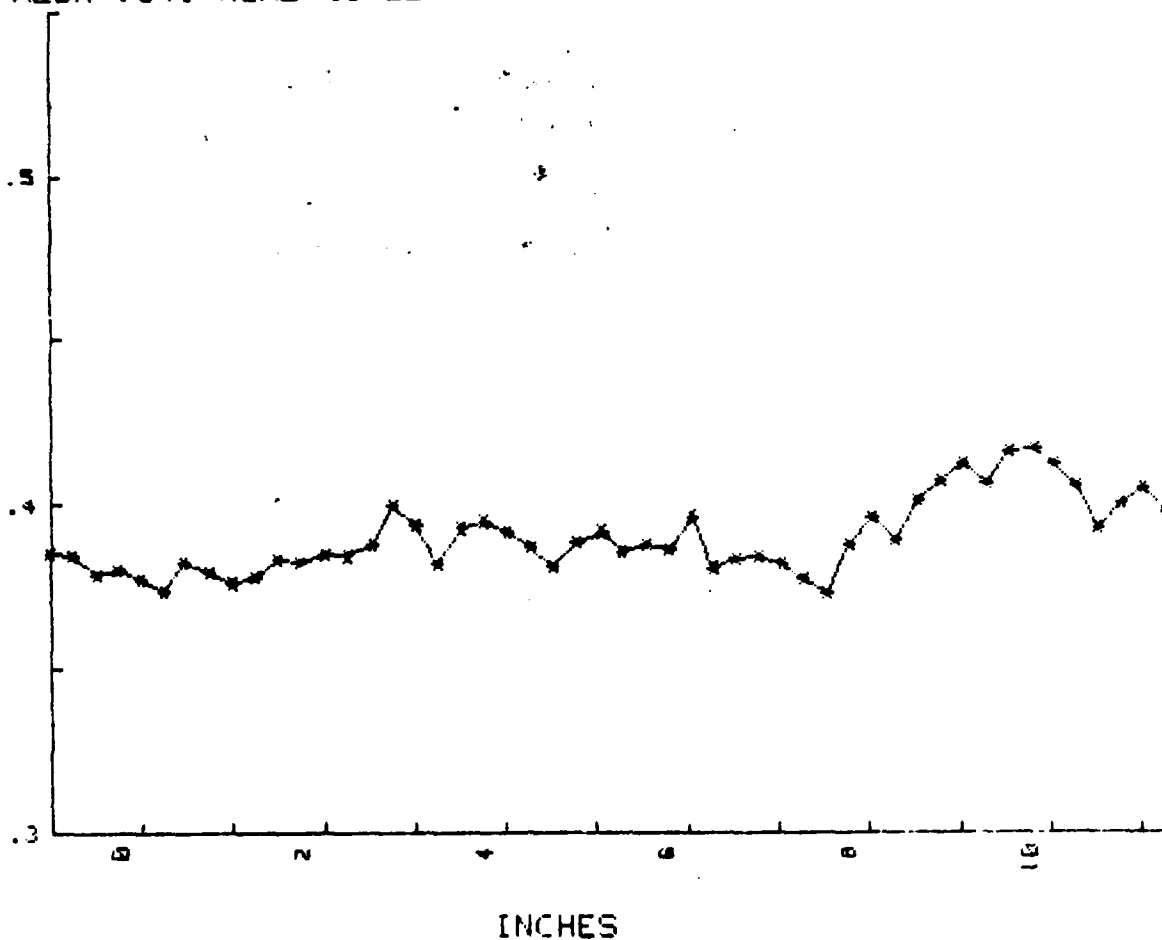


Fig. 44. Probe Survey Data at Midspan, Lower Plane
 4 Mesh Screen, Walls at 35°, Points 51 to 100
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
 PT 1 TO 50 LOWER PLANE 35 DEG
 5 MESH .041 WIRE SCREEN

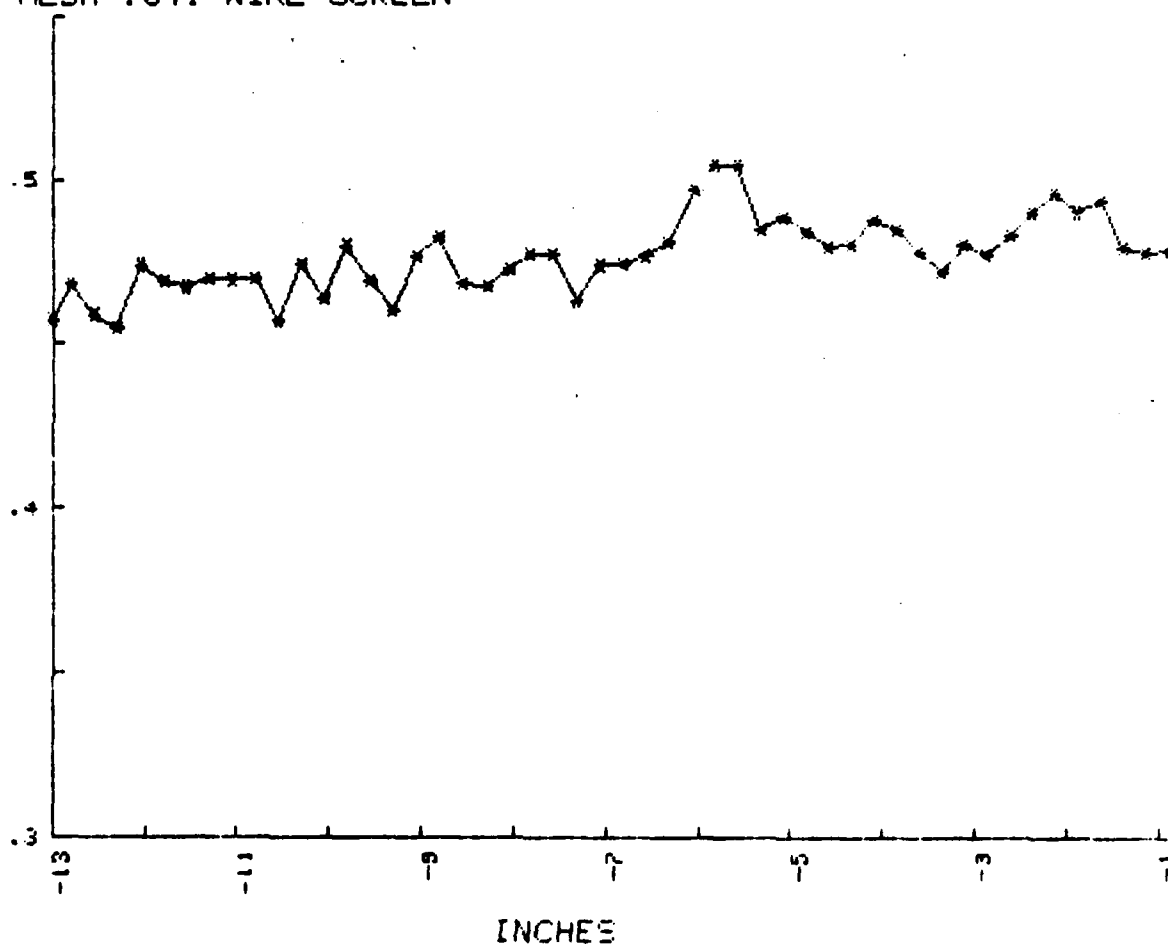


Fig. 45. Probe Survey Data at Midspan, Lower Plane
 5 Mesh Screen, Walls at 35°, Points 1 to 50
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

$P(\text{plenum}) - P_t / Q_{\text{ref}}$
 PT 51 TO 100 LOWER PLANE 35 DEG.
 5 MESH .041 WIRE SCREEN

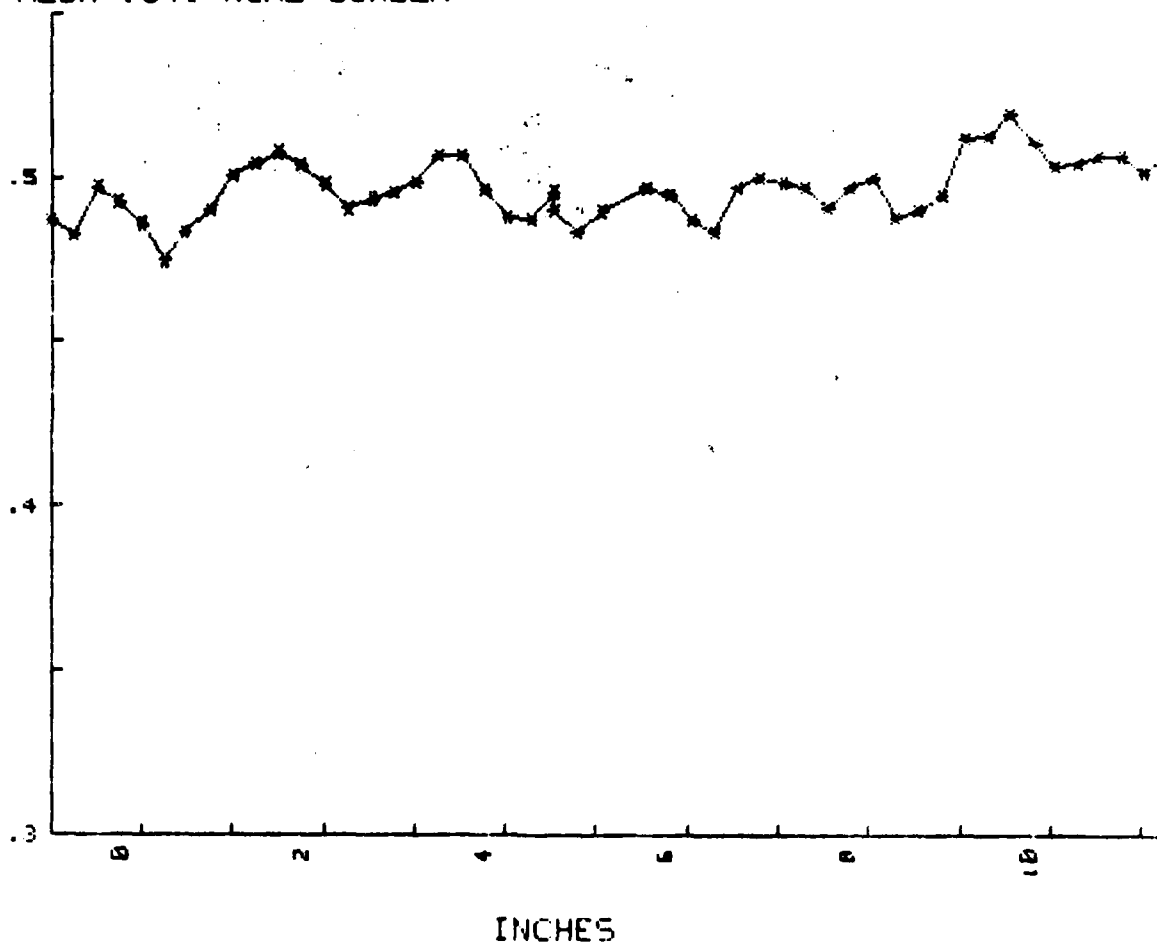


Fig. 46. Probe Survey Data at Midspan, Lower Plane
 5 Mesh Screen, Walls at 35°, Points 51 to 100
 $(P_{\text{PLENUM}} - P_t) / Q_{\text{ref}}$

Upper

Plenum

Ambient

Lower

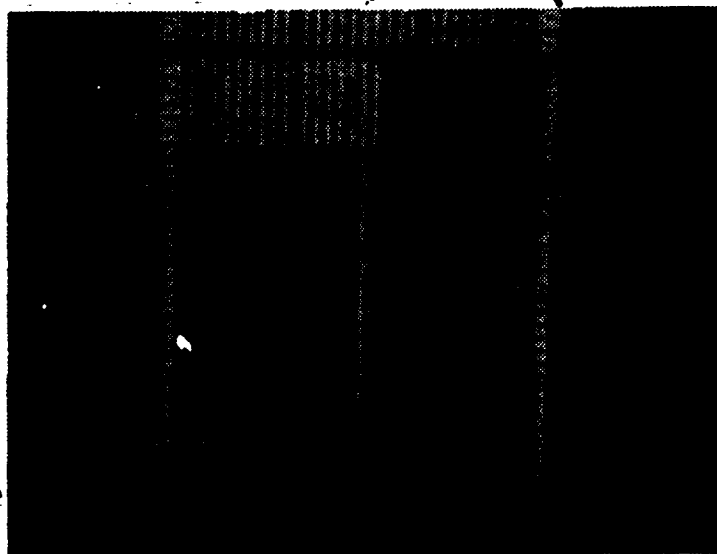


Fig. 47. Wall Static Pressure Distribution

$(P_{\text{plenum}} - P_t) / Q_1 \text{ bar}$
 LOWER PLANE MIDSPAN ($i = 5.3$)

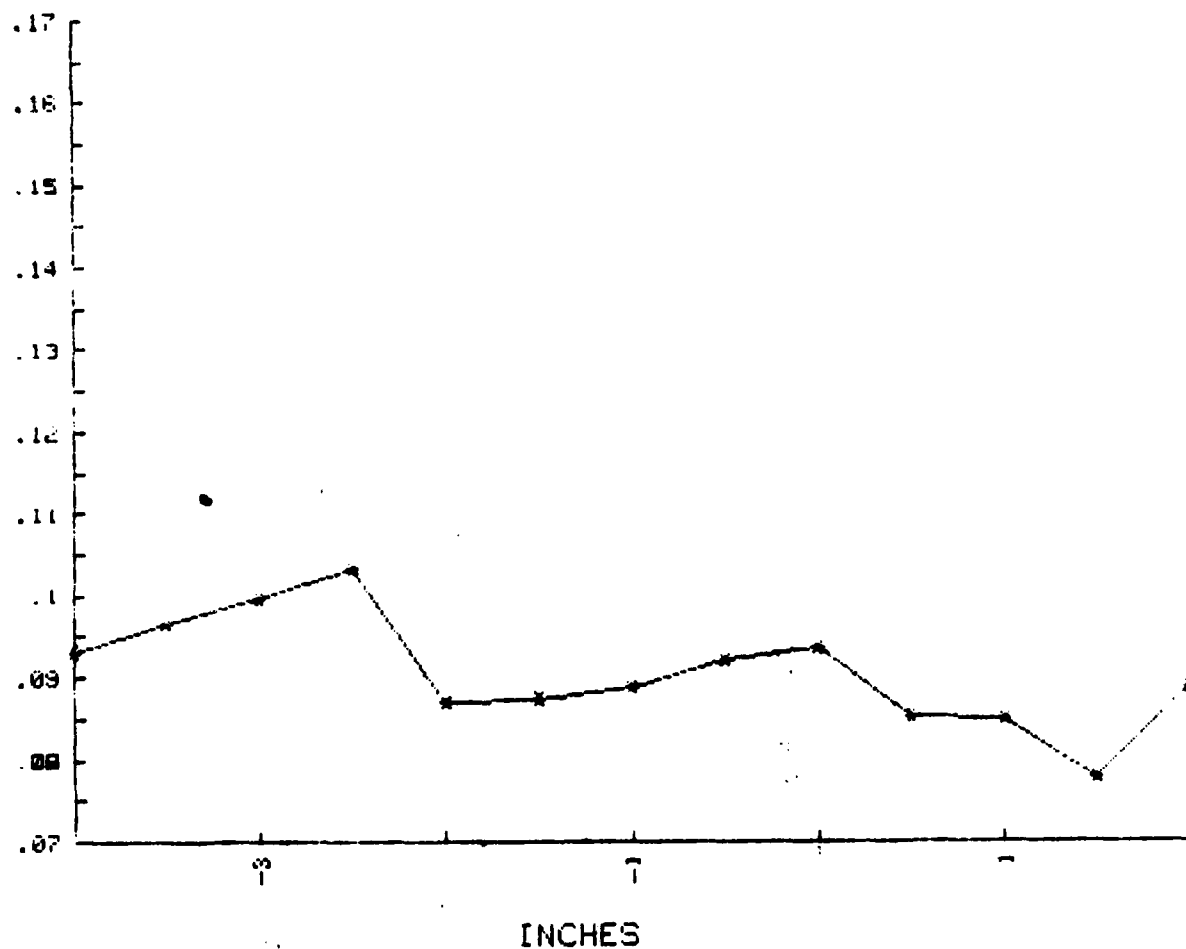
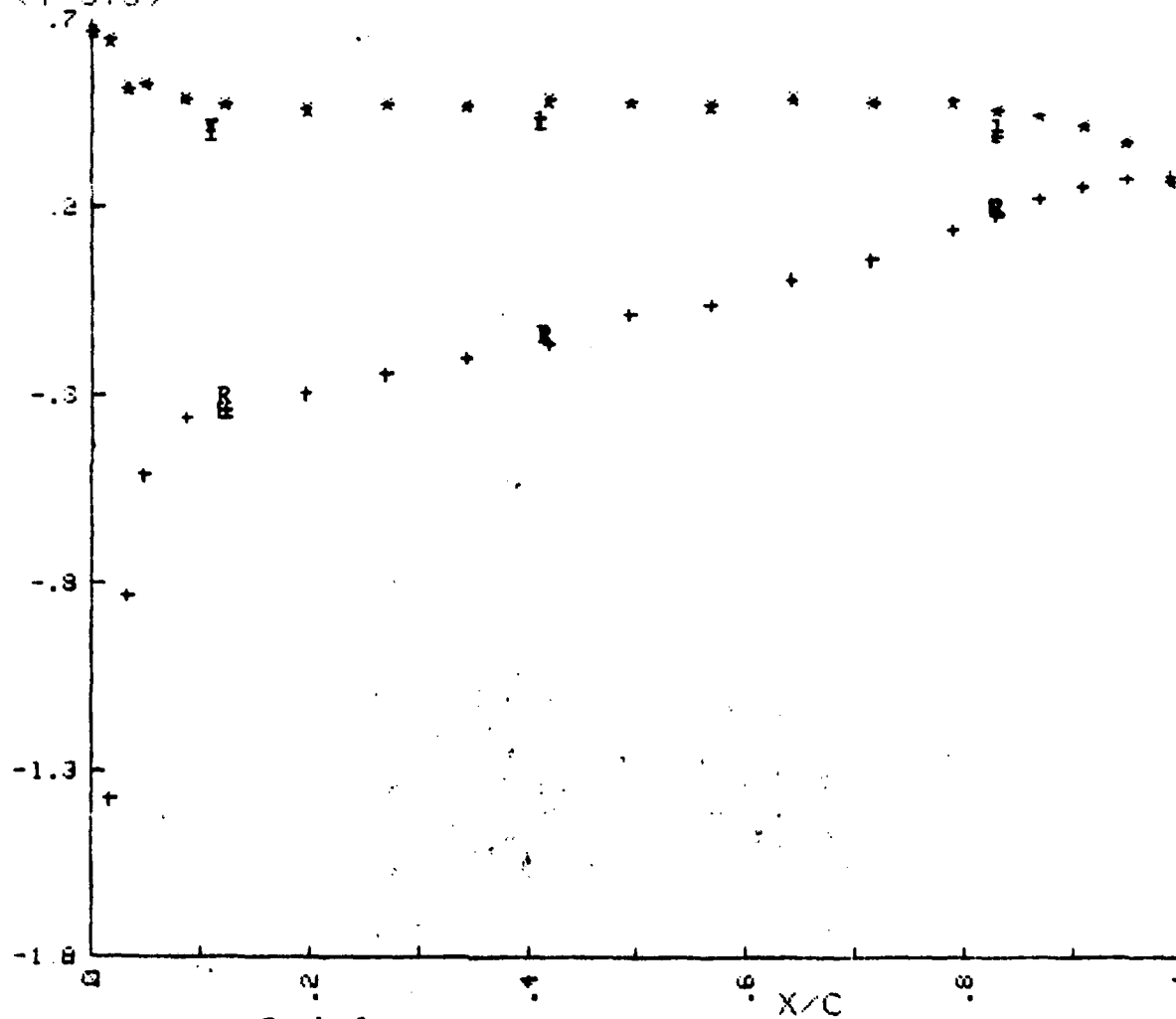


Fig. 48. Probe Survey Data at Upstream Midspan
 ($i = 5.3$, $(P_{\text{PLENUM}} - P_t) / \bar{Q}_1$, Lower Plane)

Cp1 vs X/C
 THREE CENTERMOST BLADES OVERLAYED
 (i=5.3)



Symbols:

BLADE	LEFT	CENTER	RIGHT
Pressure Side	1	*	r
Suction Side	L	+	R

Fig. 49. Blade Surface Pressure Distribution on Three Centermost Blades (i = 5.3)

$(P_{\text{plenum}} - P_t) / Q_1 \text{ bar}$
 UPPER PLANE MIDSPAN ($i = 5.3$)
 (THREE PASSAGES OVERLAYED)

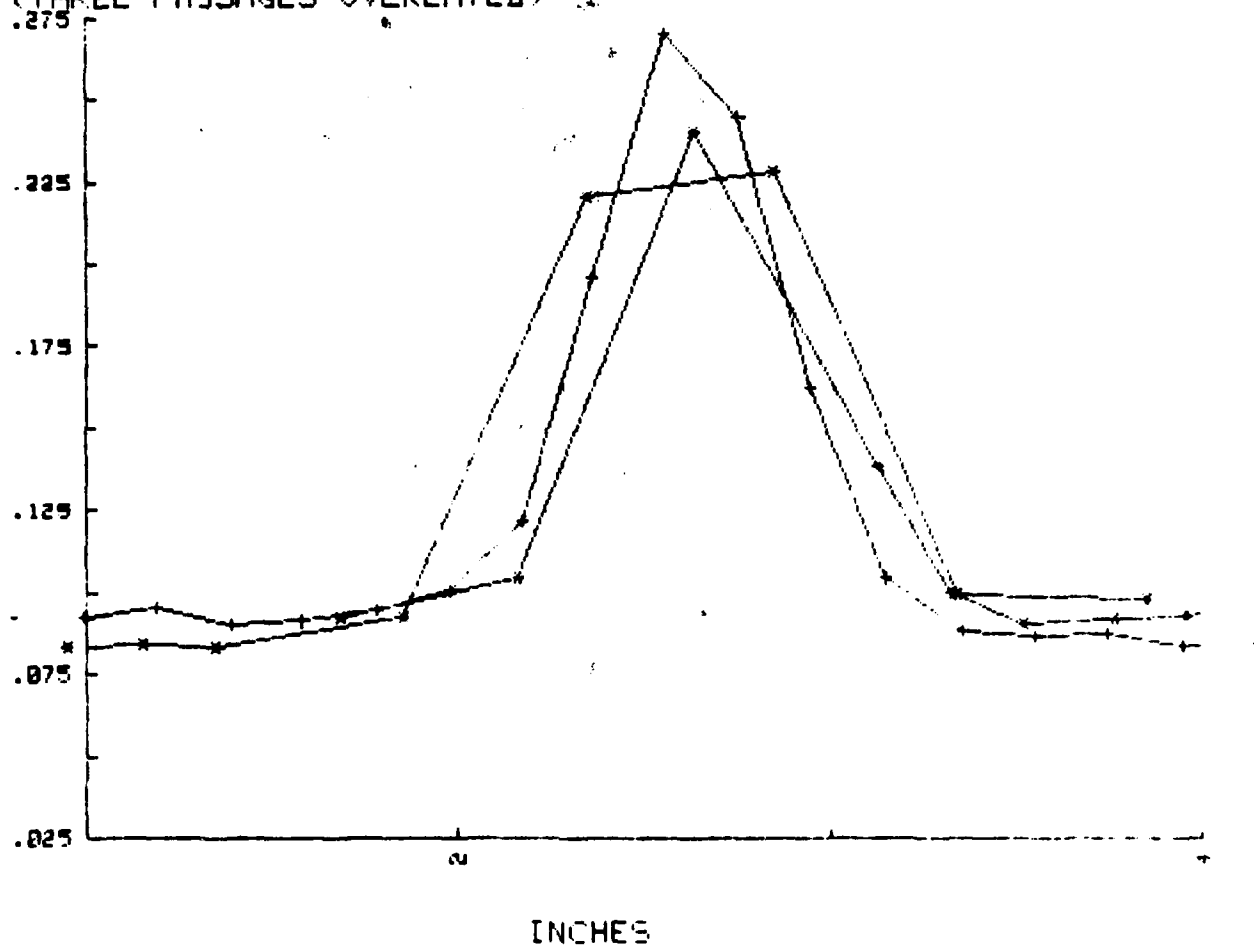


Fig. 50. Probe Survey Data at Midspan
 ($i = 5.3$, $(P_{\text{PLENUM}} - P_t) / Q_1$, Upper Plane)

$X/\bar{X}$
 UPPER PLANE MIDSPAN ($i=5.3$)
 (THREE PASSAGES OVERLAYED)

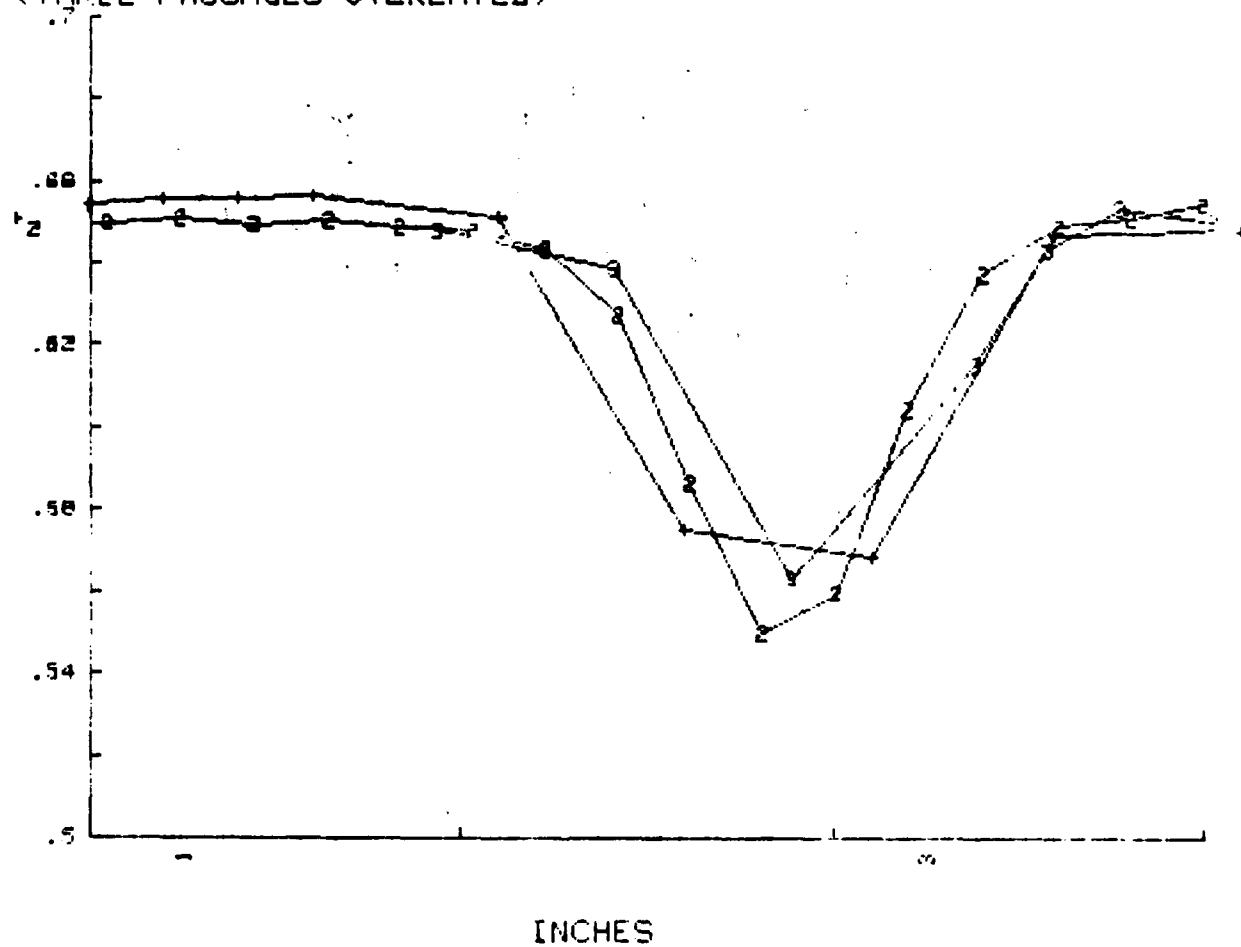


Fig. 51. Probe Survey Data at Midspan
 ($i = 5.3$, $(X/\bar{X})$, Upper Plane)

$(P_s - P_{w1}) / Q_1 \text{ bar}$
 UPPER PLANE MIDSPAN ($i = 5.3$)
 (THREE PASSAGES OVERLAYED)

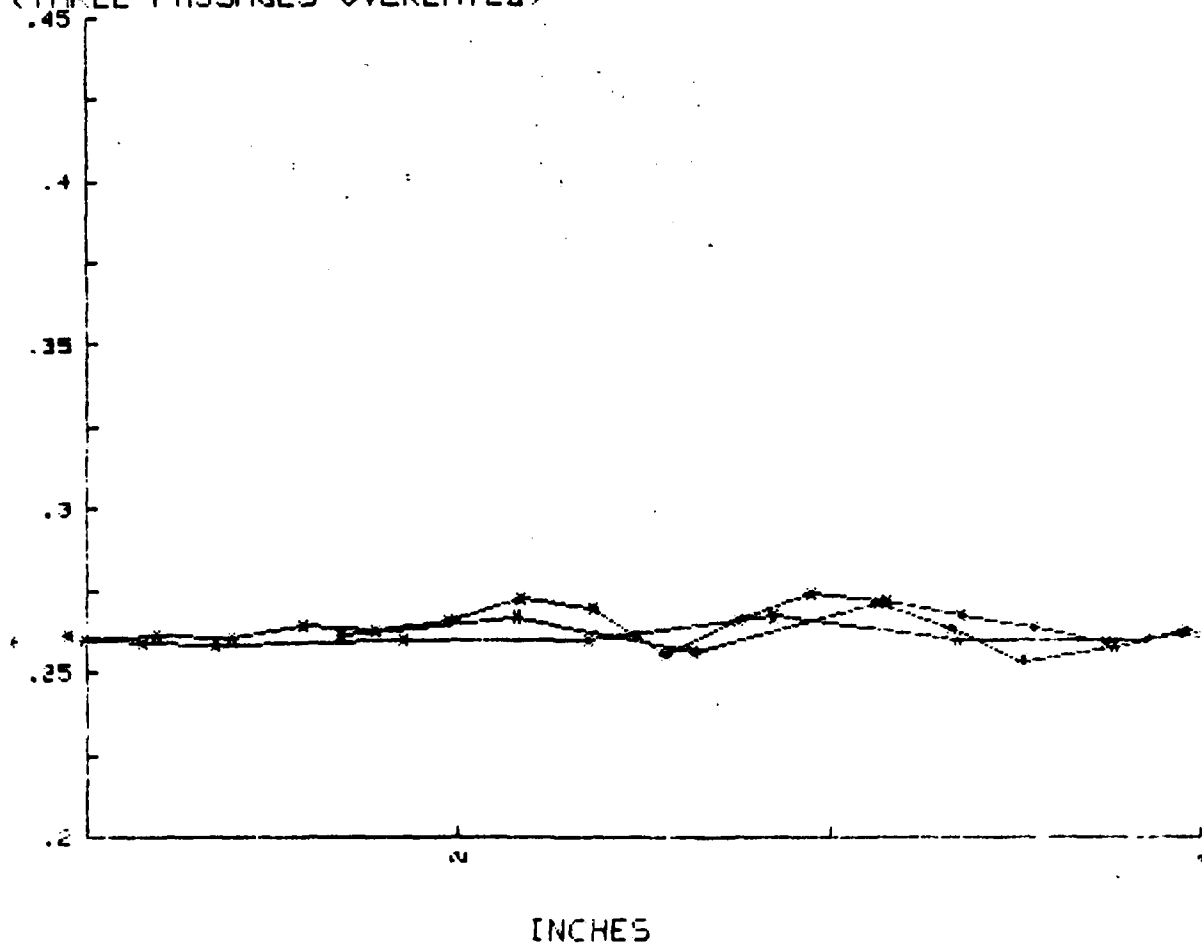


Fig. 52. Probe Survey Data at Midspan
 ($i = 5.3$, $(P_s - P_{w1}) / \bar{Q}_1$, Upper Plane)

OUTLET ANGLE
UPPER PLANE MIDSPAN (i=5.3)

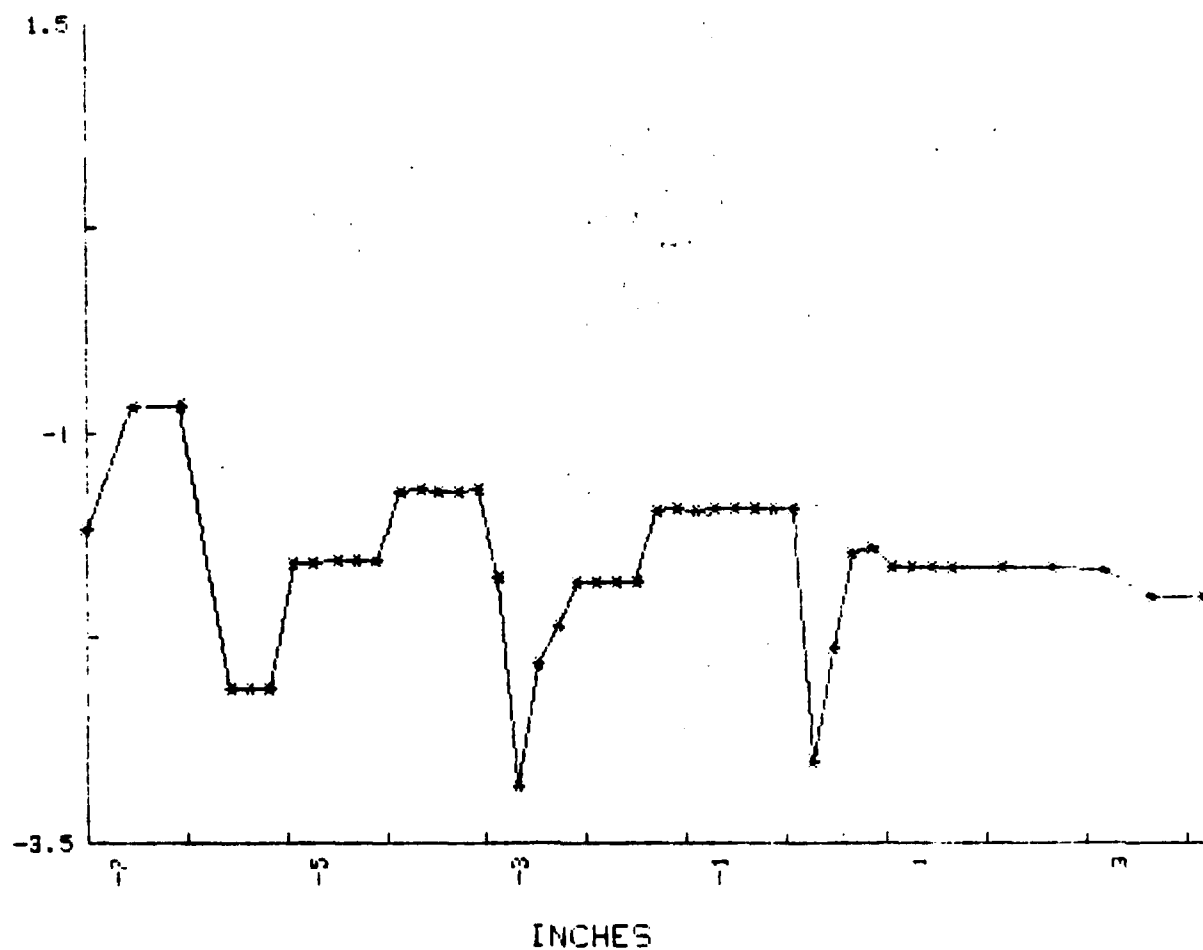


Fig. 53. Probe Survey Data at Midspan
(i = 5.3, Outlet Angle, Upper Plane)

$(P_{\text{plen}} - P_t) / Q_1 \text{ bar}$
 1 in FROM SUCTION SIDE
 CENTERMOST BLADE ($i=5.3$)

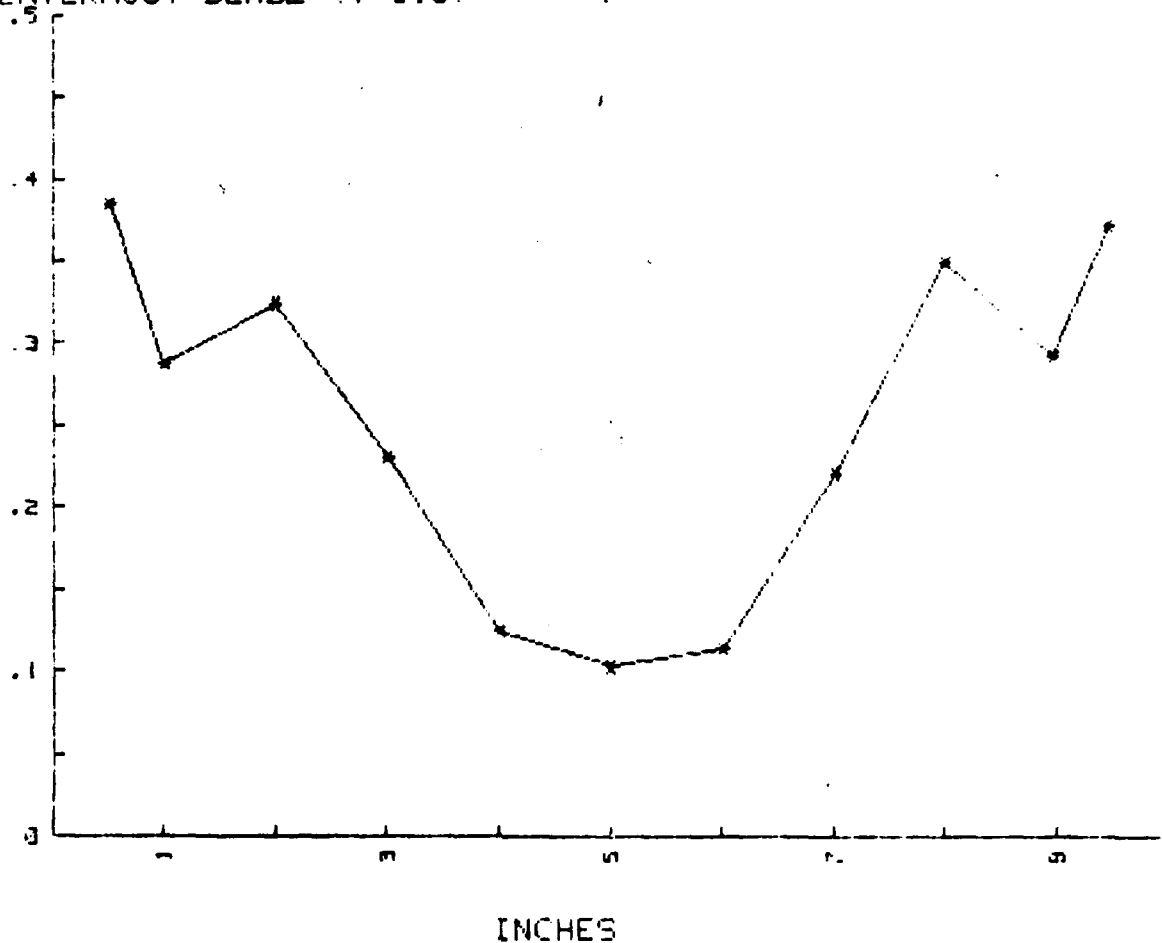


Fig. 54. Spanwise Probe Data Surveyed 1 in. from
 Suction Side of Centermost Blade
 ($i = 5.3$, $(P_{\text{PLENUM}} - P_t) / \bar{Q}_1$, Upper Plane)

$X/\bar{X}$
 1.0 in FROM SUCTION SIDE
 CENTERMOST BLADE ($i=5.3$)

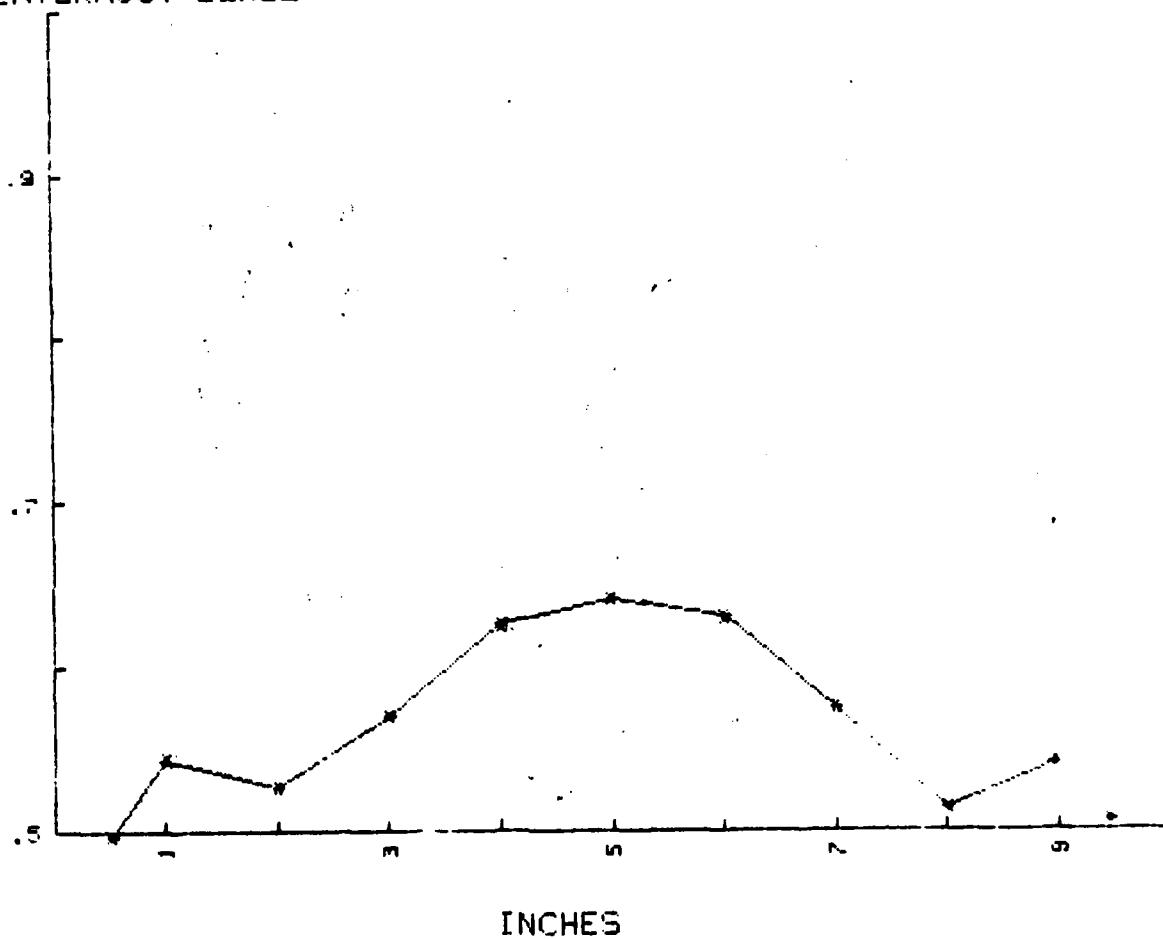


Fig. 55. Spanwise Probe Data Surveyed 1 in. from Suction Side of Centermost Blade ($i = 5.3$, $X/\bar{X}$, Upper Plane)

$(P_{\text{plen}} - P_t) / Q_1 \text{ bar}$
 1.0 in FROM PRESSURE SIDE
 CENTERMOST BLADE ($i = 5.3$)

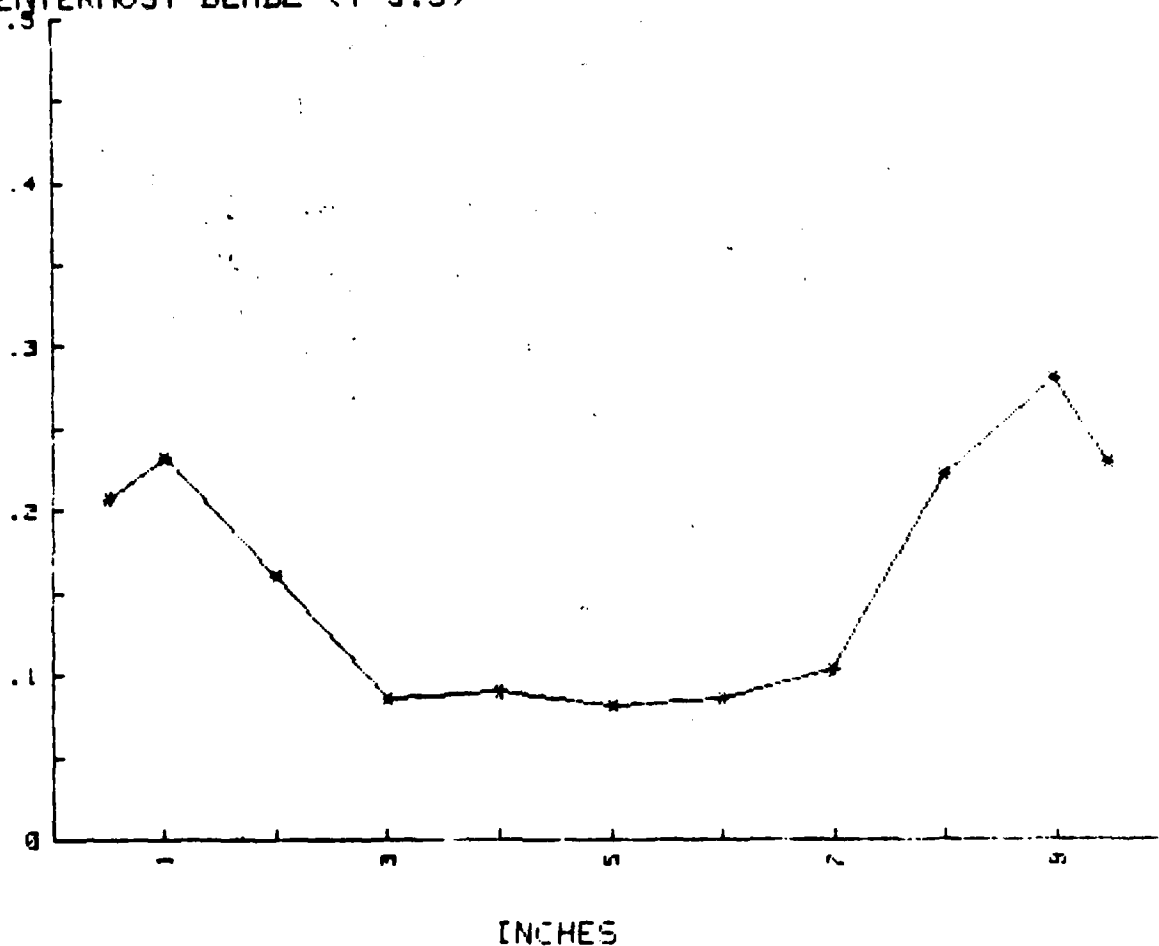


Fig. 56. Spanwise Probe Data Surveyed 1 in. from
 Pressure Side of Centermost Blade
 ($i = 5.3$, $(P_{\text{PLENUM}} - P_t) / \bar{Q}_1$, Upper Plane)

X/Xbar
 1.0 in FROM PRESSURE SIDE
 CENTERMOST BLADE (i=5.3)

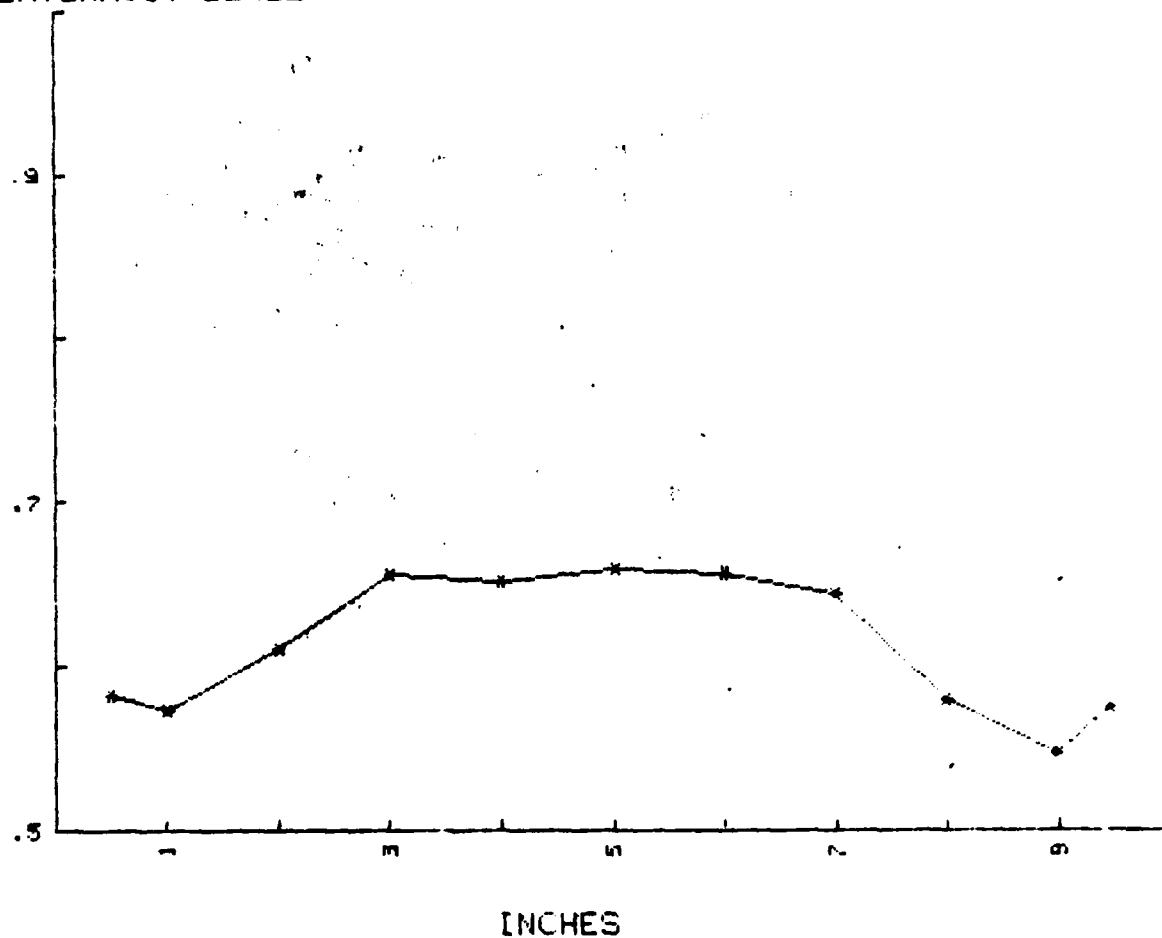


Fig. 57. Spanwise Probe Data Surveyed 1 in. from Pressure Side of Centermost Blade (i = 5.3, $X/\bar{X}$, Upper Plane)

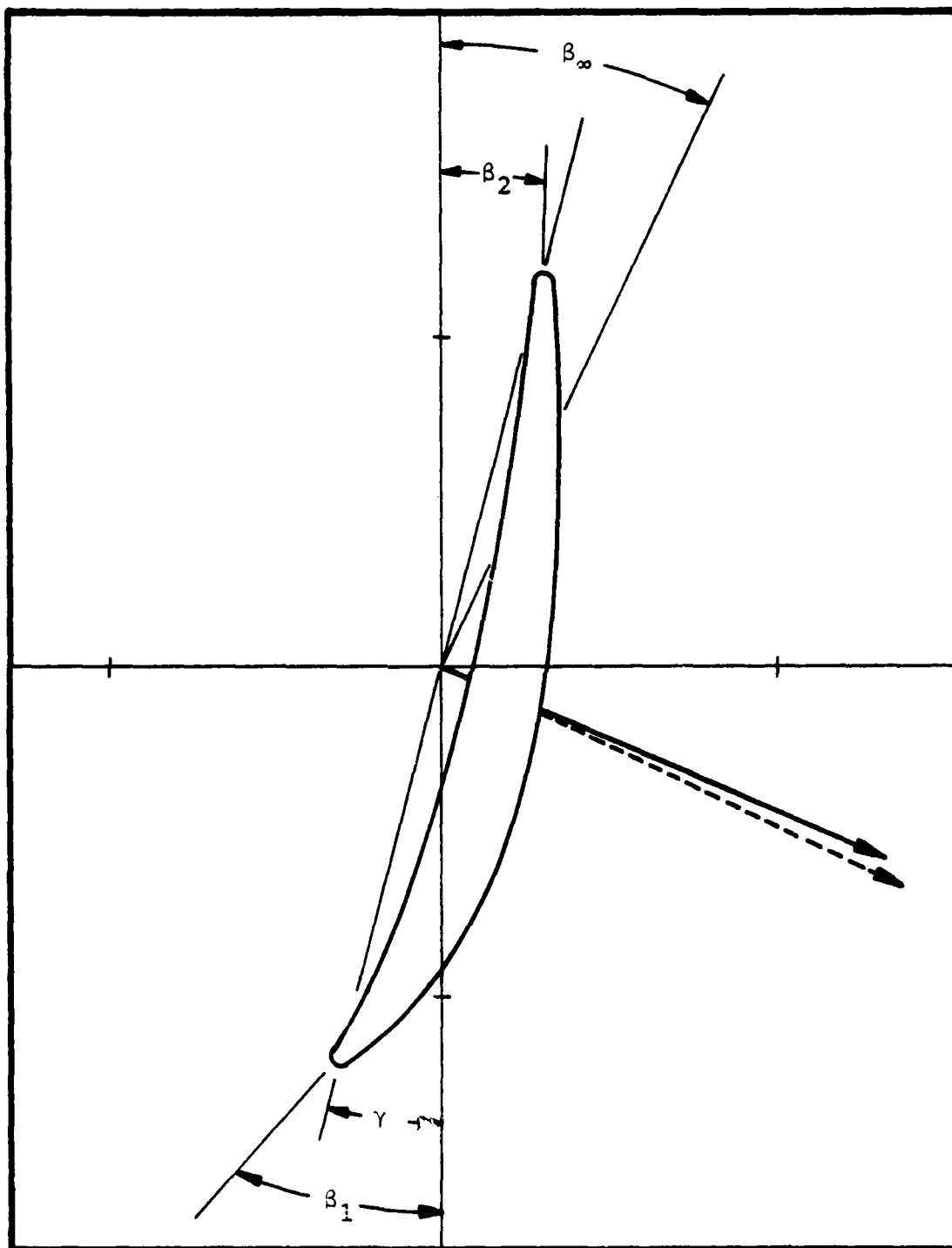


Fig. 58. Resultant Blade Force Vectors by Momentum Balance (-----) and from Surface Pressure Integration (——) $i = 5.3$

C_{p1} vs X/C ($i=5.3$)
CENTERMOST BLADE

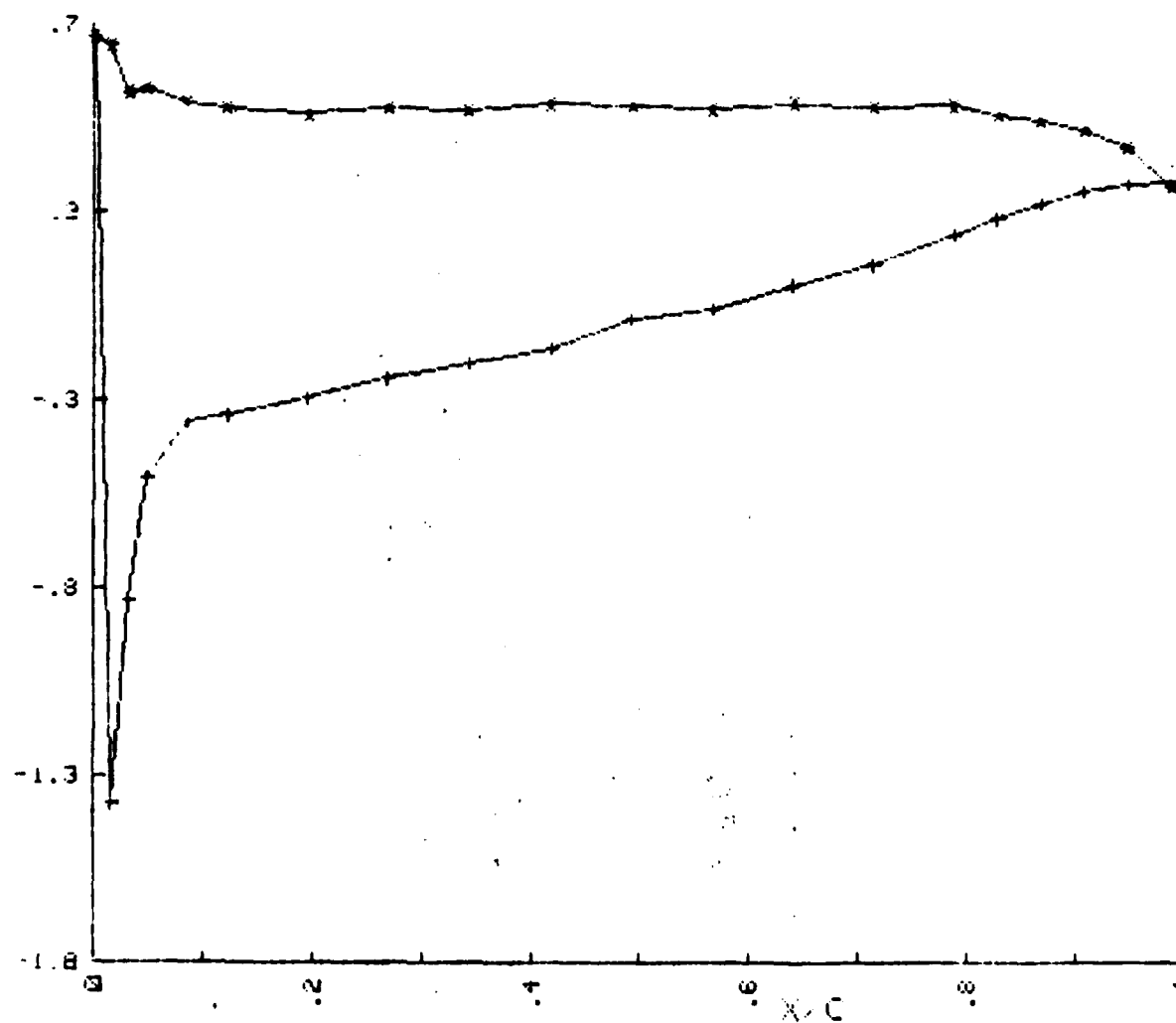


Fig. 59. Measured Blade Surface Pressure Distribution
($i = 5.3$, * = Pressure Side,
+ = Suction Side)

X_{vel} VS X/C ($i=5.3$)
CENTERMOST BLADE

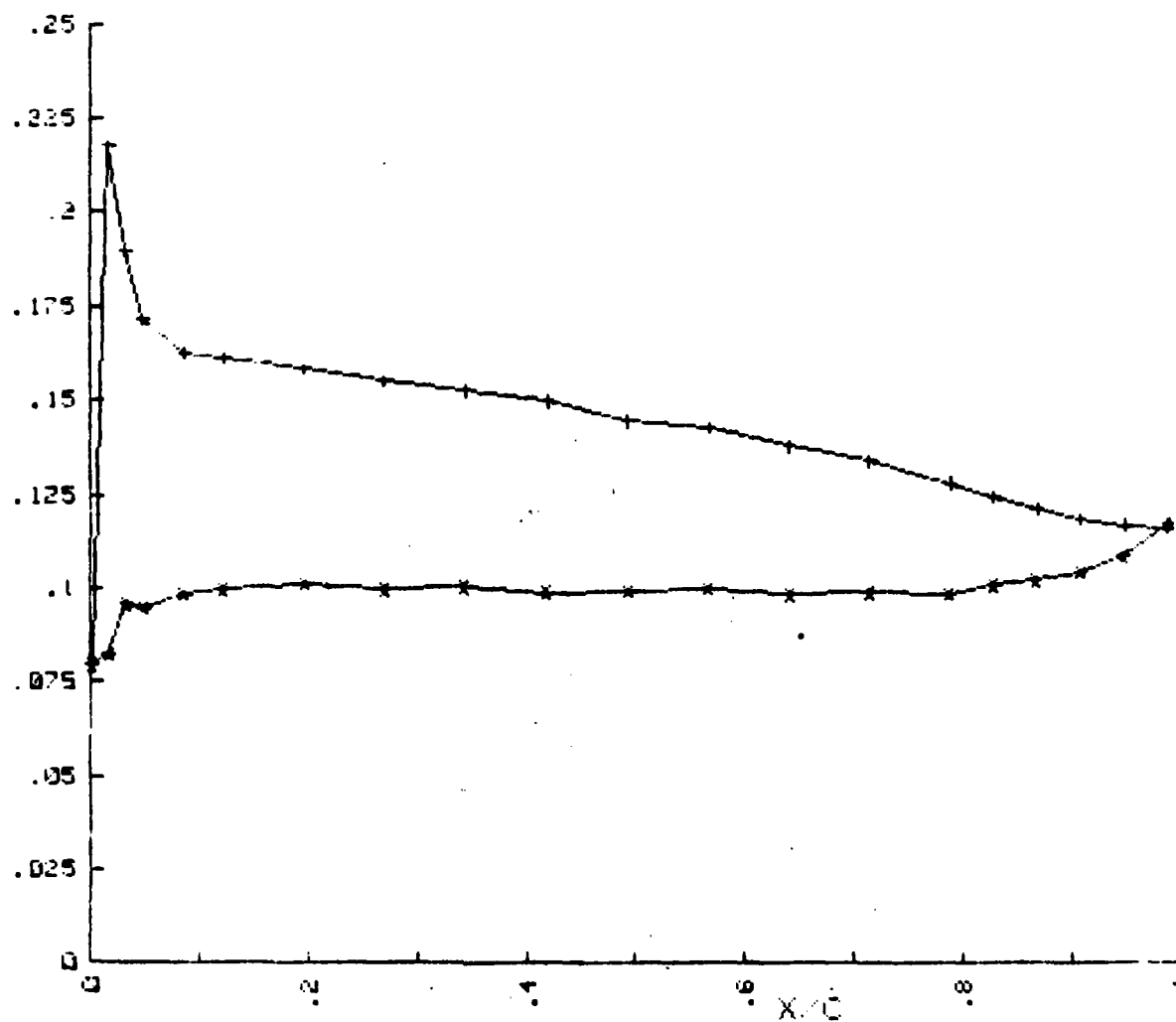


Fig. 60. Measured Blade Surface Velocity Distribution
($i = 5.3$, $*$ = Pressure Side,
 $+$ = Suction Side)

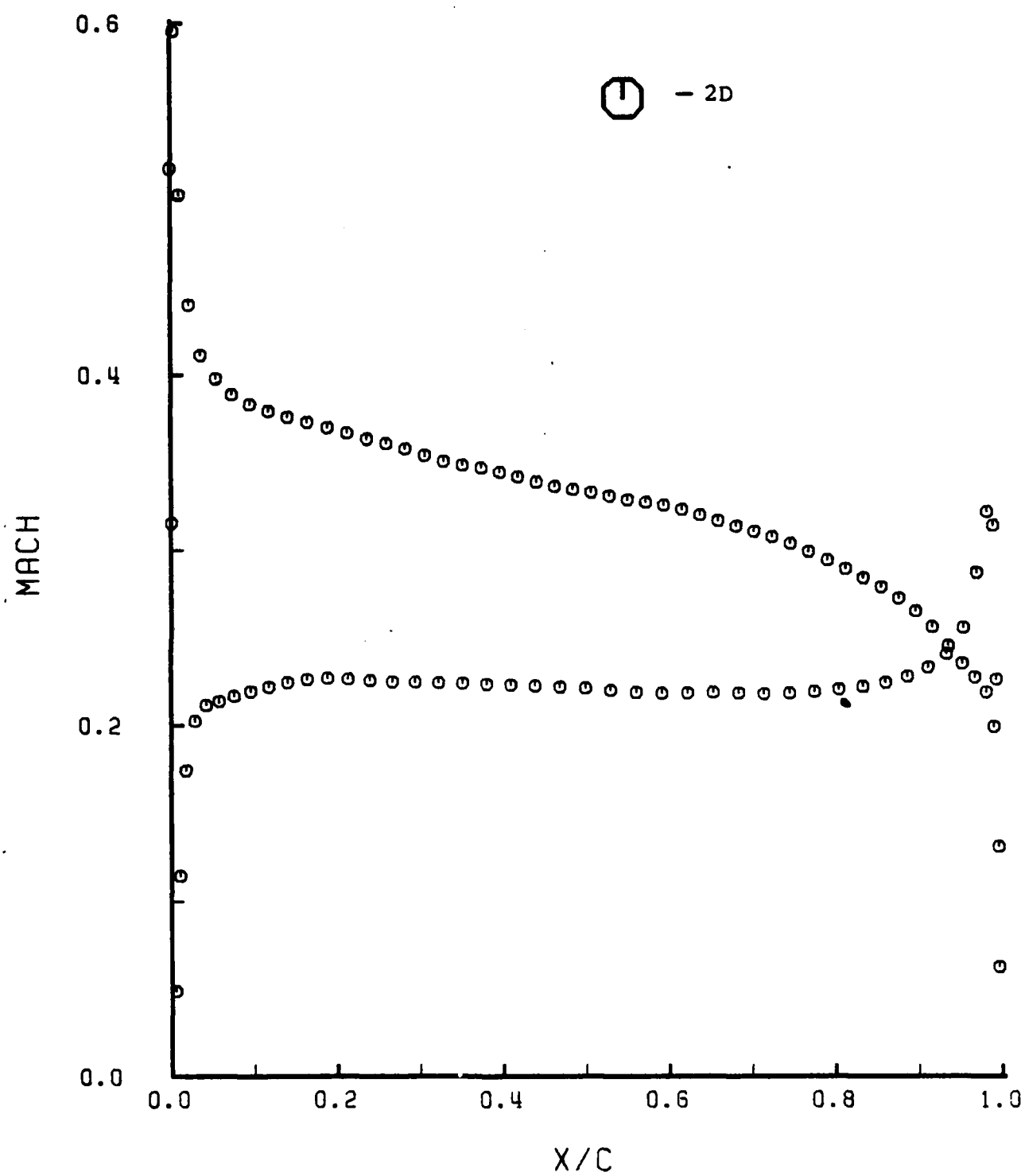


Fig. 61. 2D Code Blade Surface Mach Number Distribution
($i = 5.3$)

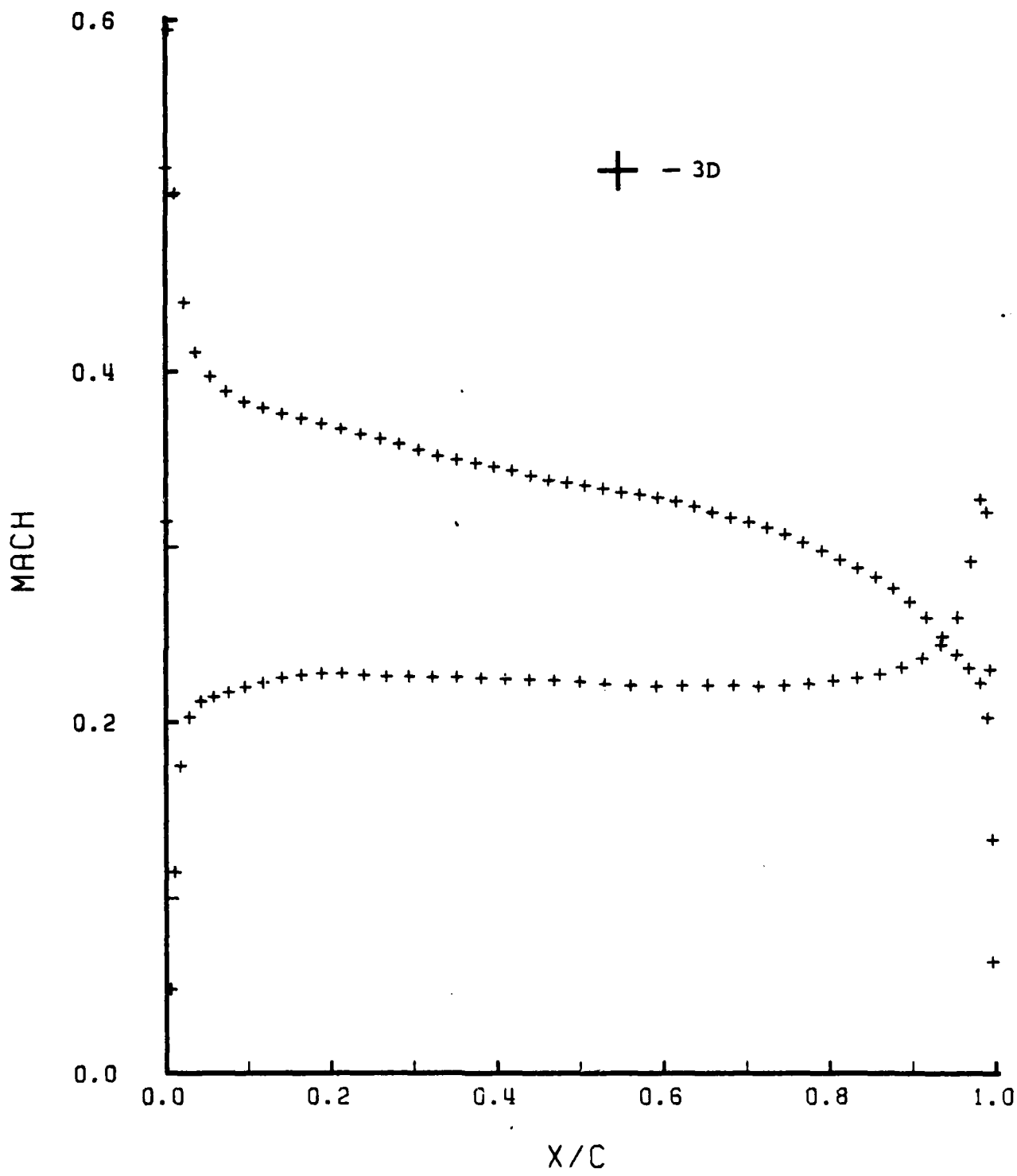


Fig. 62. 3D Code Blade Surface Mach Number Distribution
(i = 5.3)

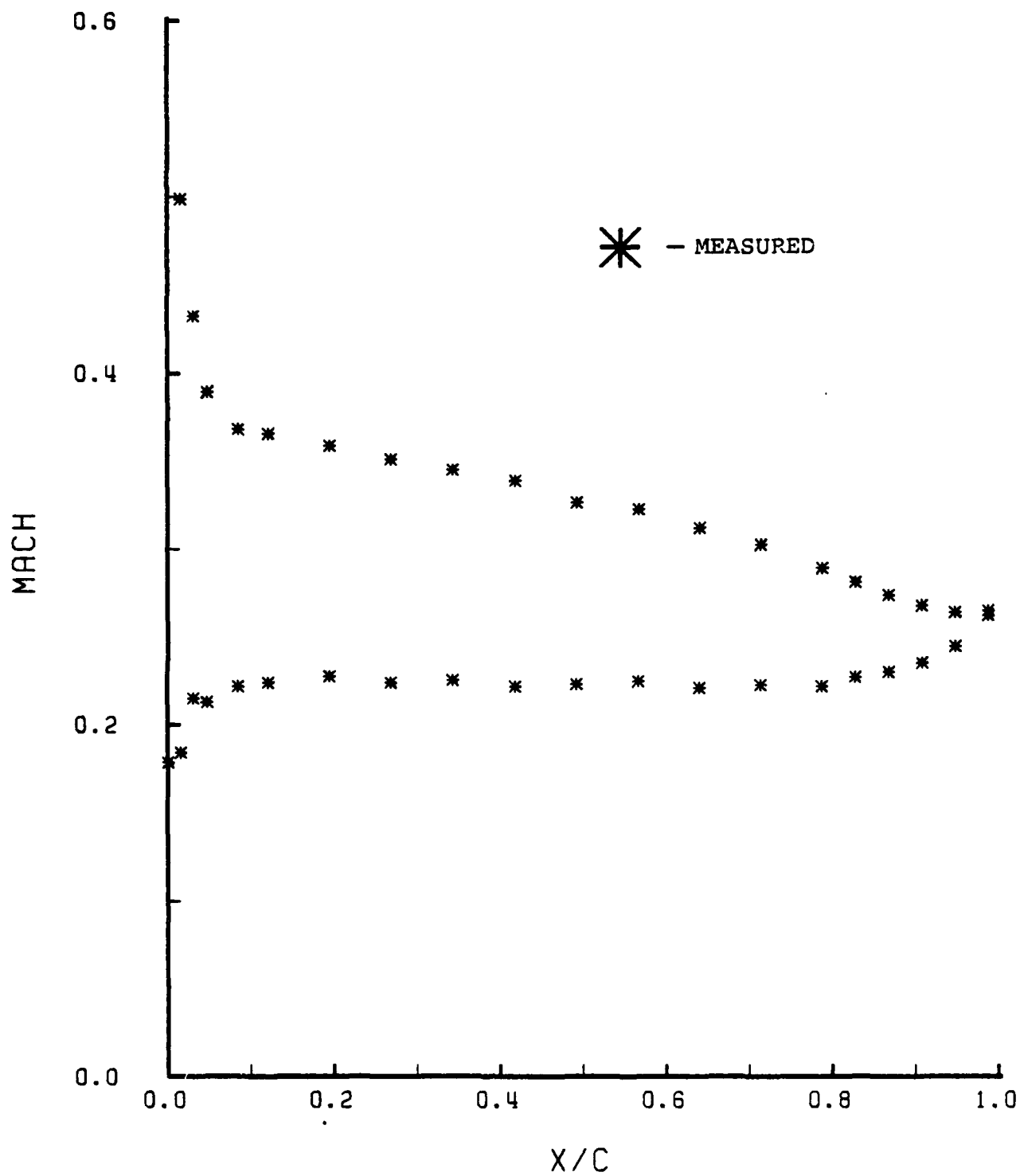


Fig. 63. Measured Blade Surface Mach Number Distribution
($i = 5.3$)

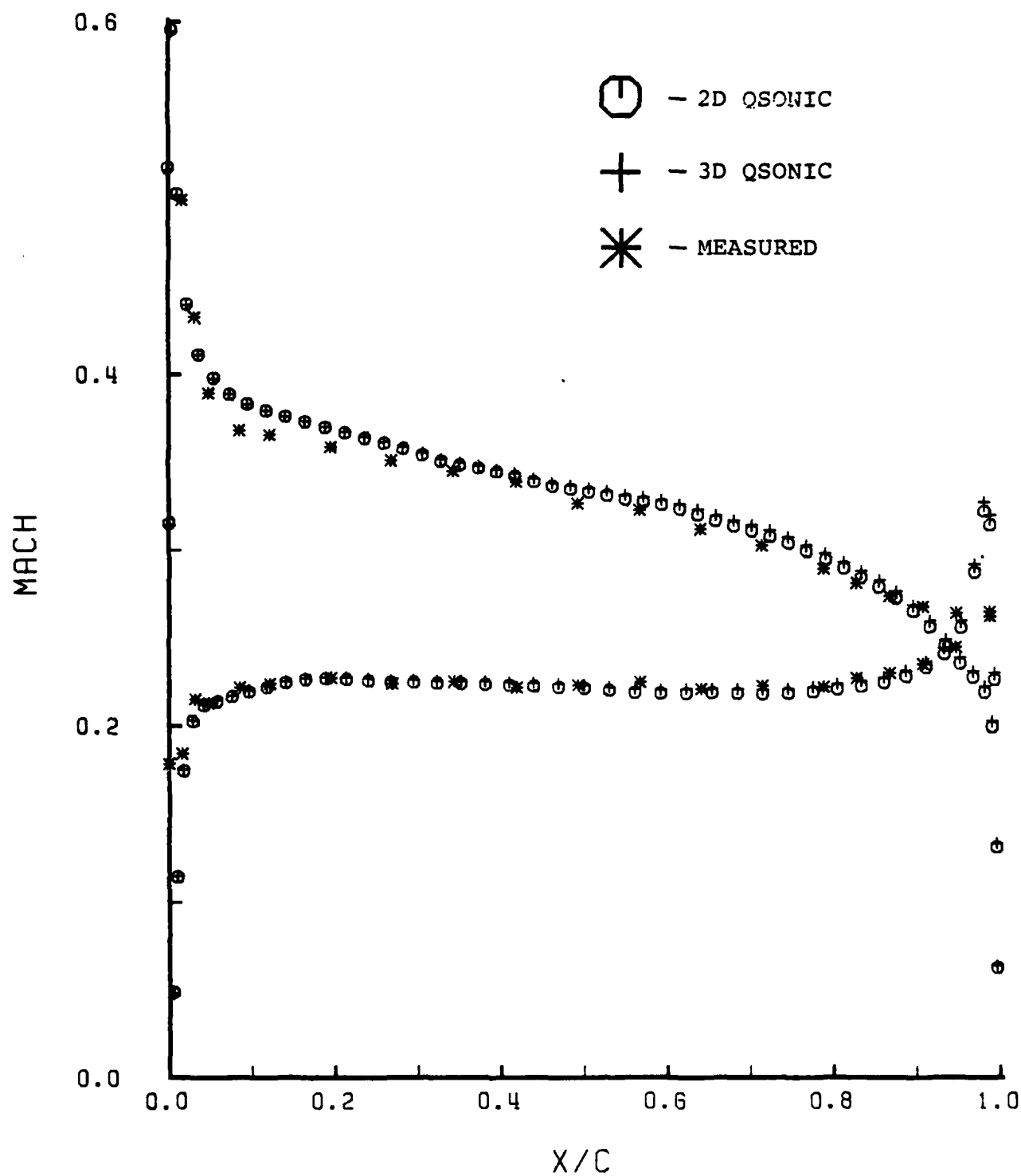


Fig. 64. Blade Surface Mach Number Distribution
($i = 5.3$)

APPENDIX A

MODIFICATION TO THE INLET GUIDE VANE SECTION
OF THE SUBSONIC CASCADE WIND TUNNEL

As discussed in Section I, Cina discovered during his test program that while the inlet flow to the test section was uniform in direction and uniform in wall static pressure, it contained a variation in velocity and stagnation pressure resulting from the wakes of the IGV's. Because the inlet guide vanes were spaced at intervals of two inches and the test cascade blades spaced at three inches, departures from strictly periodic conditions were detected from one test blade passage to another.

To alleviate this problem, the inlet guide vane arrangement was modified so that the guide vanes were placed at 1 inch intervals. In order to preserve the option of reverting to a two inch IGV arrangement and because it was not possible to machine the south wall to hold additional blades, a separate structure was placed between the bell mouth contraction and the walls of the cascade. By mounting the IGV's in a separate unit which remained fixed once installed, hardware adjustments between tests associated with a change of end wall angle were greatly simplified.

The new inlet guide vane assembly was constructed using two lengths of 10 inch steel channel as shown in Fig. A.1.

One set of guide vanes was mounted on the south side of the unit at 2 inch spacings. A second set of vanes was mounted at 2 inch spacings on the north side of the unit. When the unit was assembled the guide vanes mounted from alternate sides meshed, resulting in an inlet guide vane spacing of 1 inch. The unit was provided with a single hand crank at the east end so that the vanes would be adjusted in unison. Once installed the complete structure could be left in place when the cascade north wall was removed to adjust air inlet angle. The one inch vane spacing ensured that periodicity at the test section would result for any test blade spacing which was a multiple of 1 inch. Equally important, the wakes remaining at the inlet to the test cascade would be greatly reduced as a result of closer spacing.

Figure A.1 shows the details of the inlet guide vane unit, while Fig. A.2 shows the assembly in relation to the bellmouth contraction and the side walls. Figure A.3 shows the mechanism to adjust the inlet guide vanes. Figure A.4 shows a view of the IGV assembly from the north side. A view of the Cascade Wind Tunnel partially assembled (north side wall off) is shown in Fig. A.5.

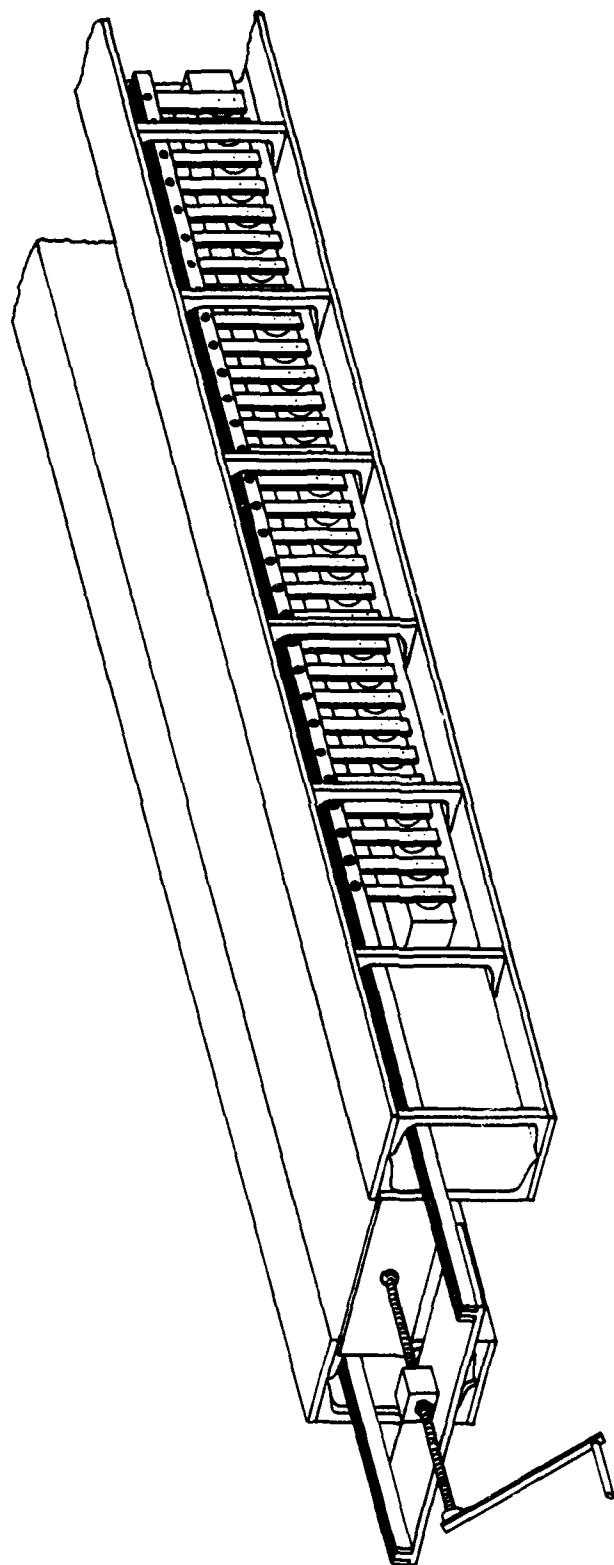


Fig. A.1. Inlet Guide Vane Assembly

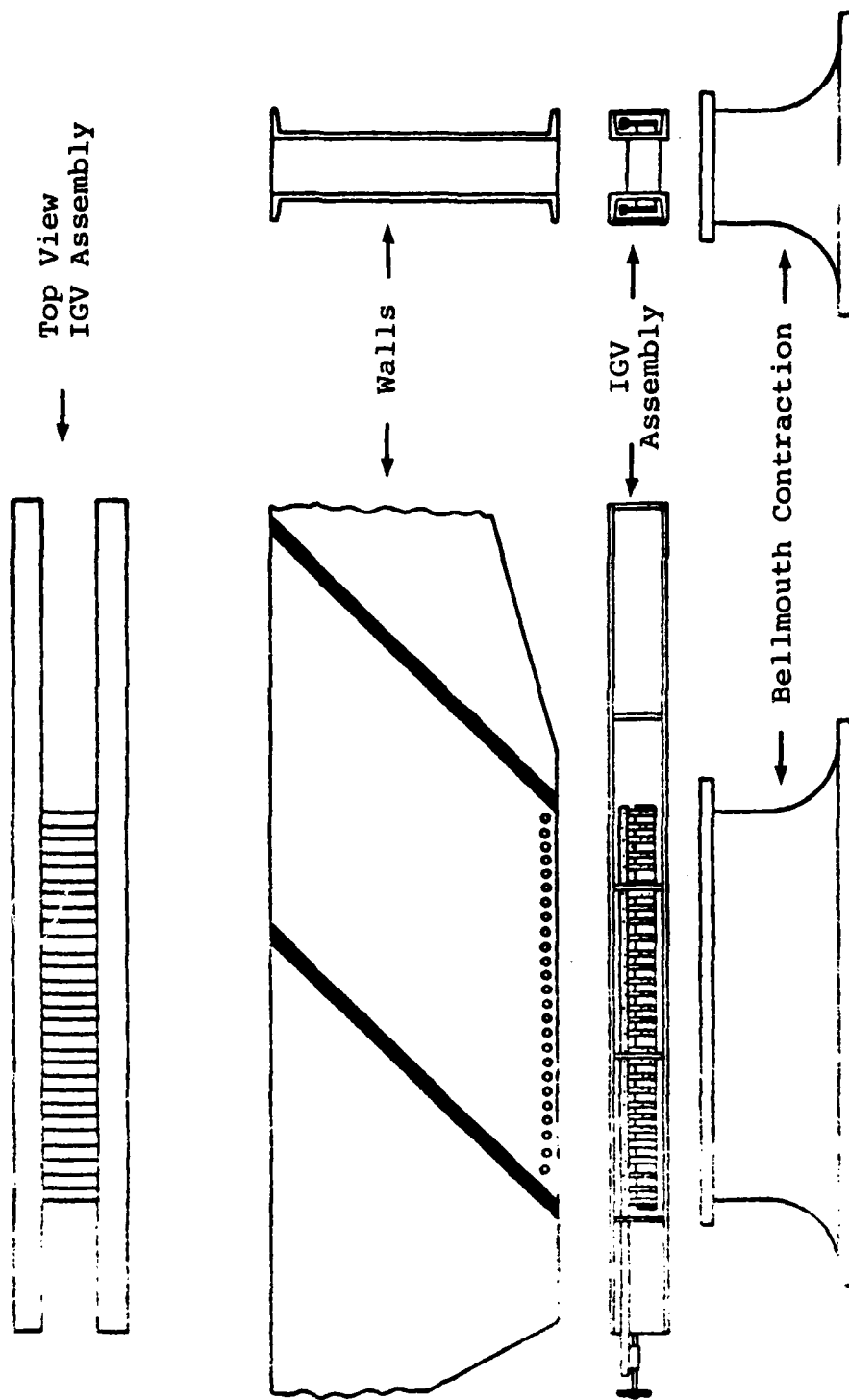


Fig. A.2. Cascade Wind Tunnel Sub-Assemblies

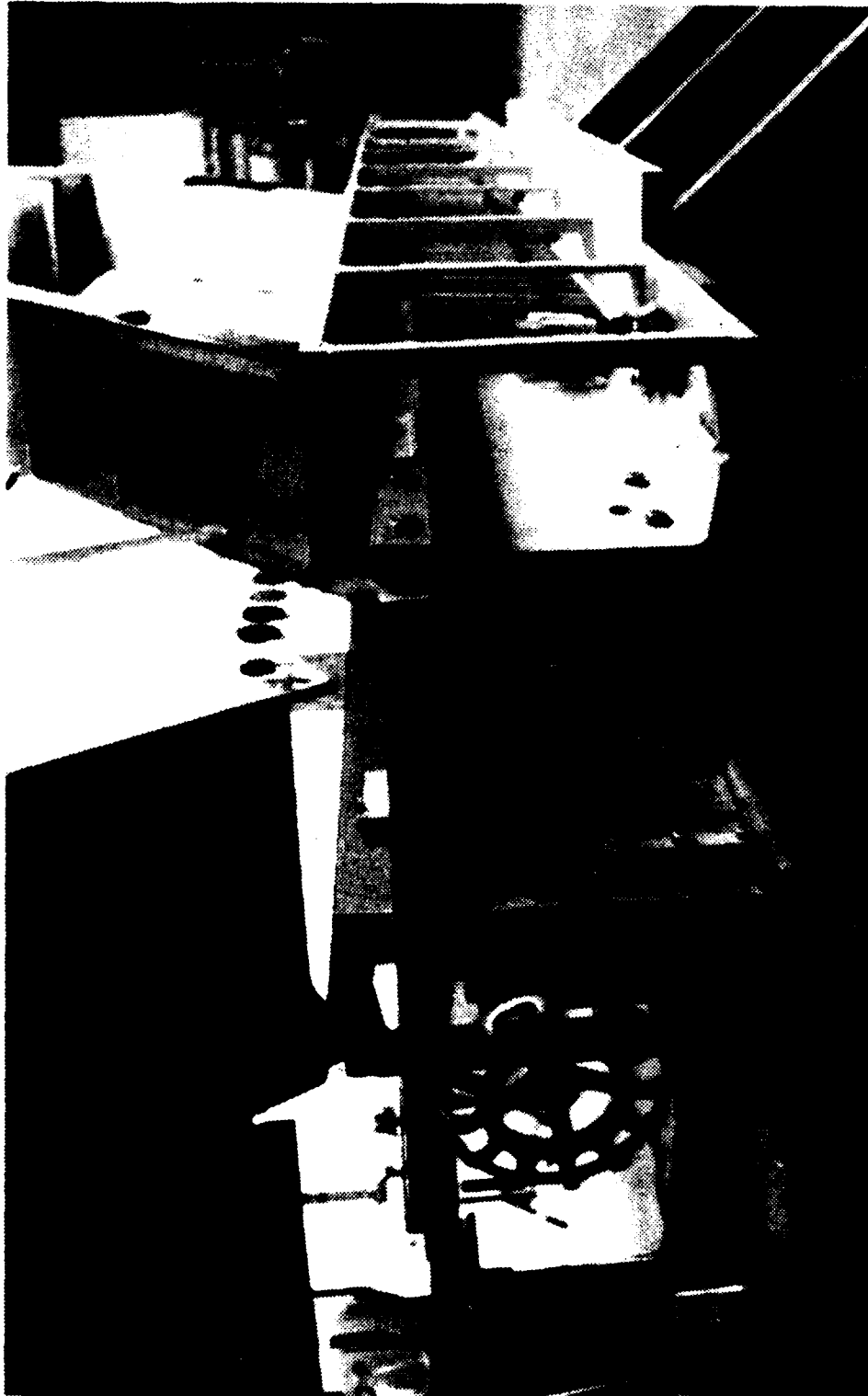


Fig. A.3. View of the IGV Adjustment Mechanism

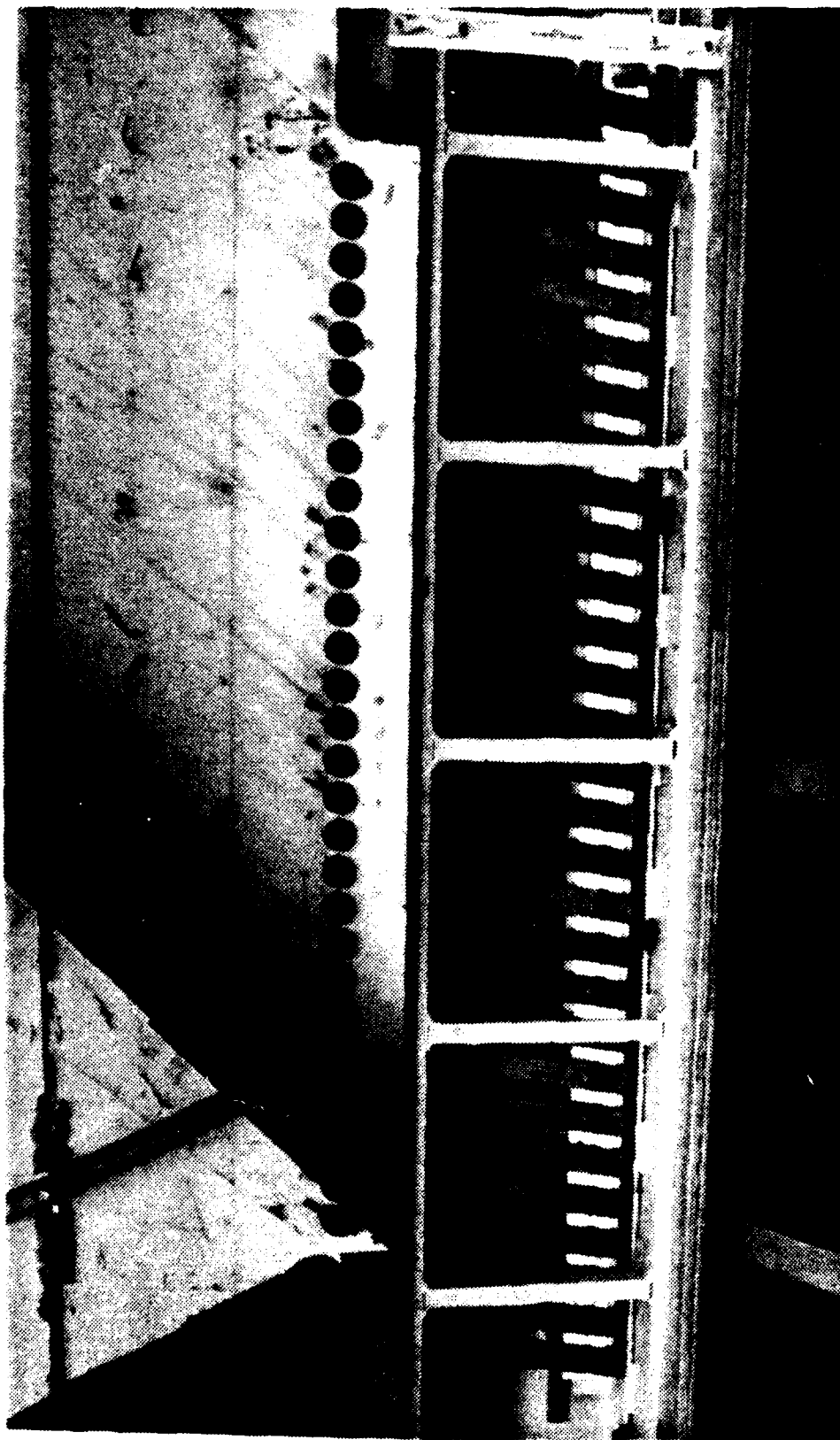


Fig. A.4. Side View of the IGV Assembly



Fig. A.5. View of the Subsonic Cascade Wind Tunnel (North Wall Removed)

APPENDIX B

SELECTION AND INSTALLATION OF SCREEN MATERIAL

Pankhurst and Holder [Ref. 17] show that the turbulence of an airstream can be increased by placing a coarse mesh across the flow upstream of the test section. One of the most effective methods which is used for the reduction of turbulence and non-uniformities also consists of placing a mesh screen across the tunnel. Screens used for this purpose are of a finer mesh and are placed at a greater distance from the test section, and normally in the low-speed region upstream of the contraction in a conventional subsonic wind tunnel. By using such a screen the large scale eddies are removed at the expense of the introduction of a greater number of smaller eddies which decay rapidly.

McEligot [Ref. 16] investigated the possibilities of reducing non-uniformities in the test section of the subsonic cascade wind tunnel. His investigation and recommendations were completed while the inlet guide vanes were still at 2 inch spacings. McEligot concluded that some modification was necessary to achieve one percent uniformity for the mean velocity at the test cascade inlet plane and suggested several options. One of the options suggested was placing the turning vanes (inlet guide vanes) at a closer pitch. As explained in Appendix A, the pitch of the inlet

guide vanes was reduced from 2 inches to 1 inch. This new inlet guide vane arrangement did result in a one percent uniformity for the mean velocity at the test cascade inlet plane.

The other approach suggested was the use of wire gauze screens. McEligot showed that the velocity distribution appeared to be largely dependent on a pressure drop coefficient K , defined by the equation

$$K = \frac{p_1 - p_2}{\frac{1}{2} \rho V^2}$$

where p_1 and p_2 are the pressures upstream and downstream of the screen respectively. This pressure drop coefficient depends mainly on the blockage coefficient β defined by the equation

$$\beta = (1 - d/l)^2$$

where d is the diameter of wire used in the screen and l is the distance between the wires. This blockage coefficient is commonly referred to as "percent open area" in catalogues of industrial wire cloth and woven wire screens.

For the velocities and flow angles used in the cascade wind tunnel, McEligot recommended using a wire gauze screen with a resistance coefficient, K , of 2.2 and a blockage coefficient, β , of 0.47. However, since the new inlet guide vane arrangement resulted in a one percent uniformity for

the mean velocity and the pressure drop across a screen with a blockage coefficient of 0.47 was expected to be higher than could be tolerated for the desired test conditions, screens with a slightly higher blockage coefficient (higher percent of open area) were selected to be tested.

The screens tested were of the following configurations:

MESH	WIRE DIAMETER (inches)	BLOCKAGE COEFFICIENT
4	.0410	.6989
5	.0410	.6320
16	.0105	.6922

Until the effectiveness of wire gauze screen in reducing non-uniformities in this facility was proven, a temporary installation of the screen material was considered adequate for testing purposes. The test screen was installed in the cascade wind tunnel by placing it between the inlet guide vane assembly and the north and south end walls. This arrangement placed the screen 7.25 inches downstream of the inlet guide vanes and 19.3 inches upstream of the lower test plane. Figure B.1 shows the installation of the wire gauze screen.

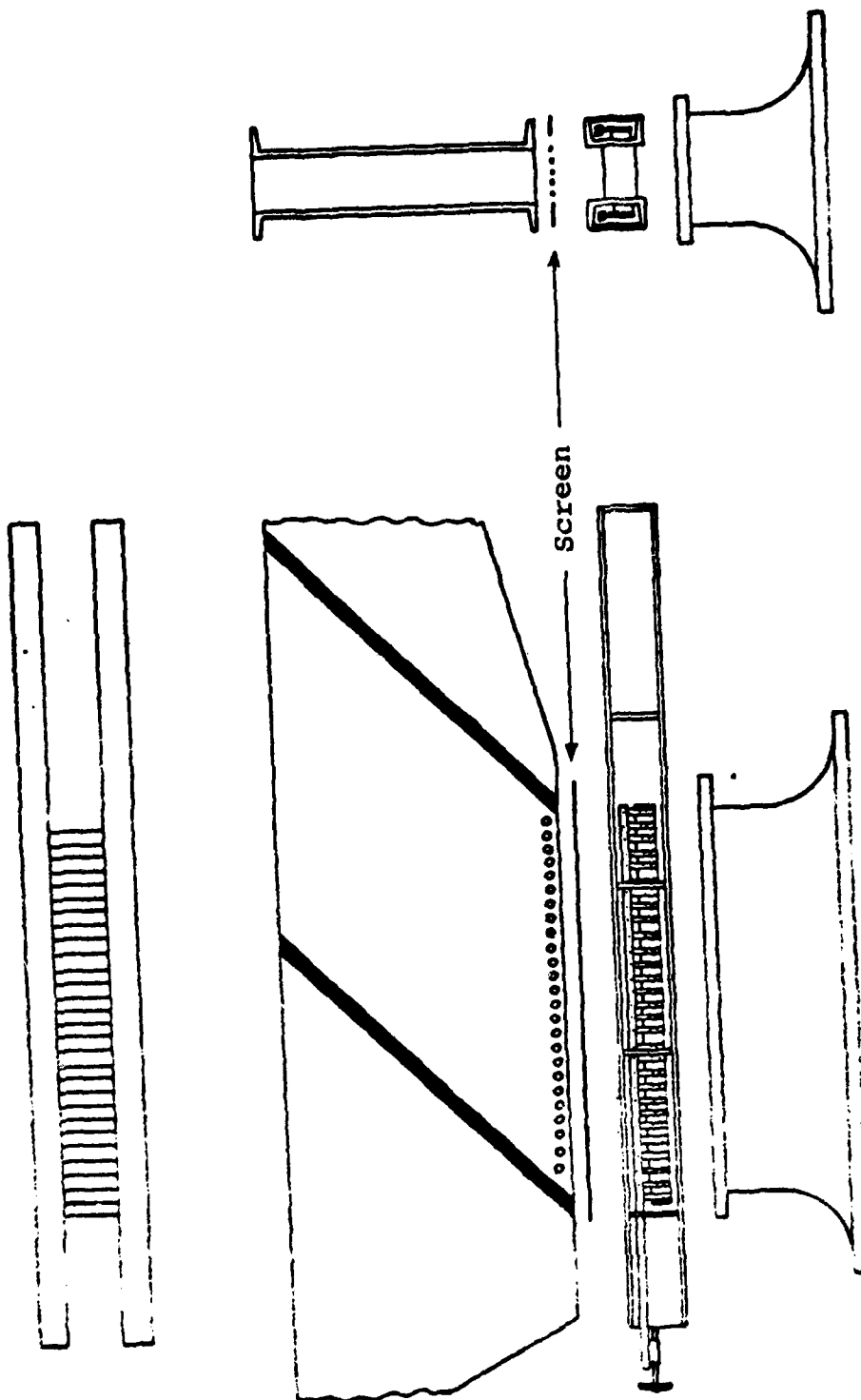


Fig. B.1. Screen Installation

APPENDIX C

CASCADE PERFORMANCE PARAMETERS

(by F. S. Cina; reproduced with minor changes from Ref. 7)

The performance of a cascade is specified in terms of the deviation angle (δ) and the loss coefficient ($\bar{\omega}$) for given inlet conditions. In Ref. 1 the loss coefficient is shown to correlate in terms of the Diffusion Factor (D). In the present work, the performance parameters were calculated using the following expressions:

1. Loss Coefficient ($\bar{\omega}$)

$$\bar{\omega} = \frac{\bar{C}_{p_{t_1}} - \bar{C}_{p_{t_2}}}{\bar{C}_{p_{t_1}} - \bar{C}_{p_1}} \quad (1)$$

where the mass averaged pressure coefficients in Eq. (1) are defined in Appendix C of Ref. 7. It is shown in Appendix C of Ref. 7 that the effect of time dependent supply conditions are removed and the effect of AVDR is included in the use of Eq. (1).

2. Diffusion Factor (D)

$$D = 1 - \frac{W_2}{W_1} + \frac{\Delta W_u}{2\sigma W_1} \quad (2)$$

3. Pressure Rise

$$C_{p\text{static}} = \frac{\bar{P}_2 - \bar{P}_1}{\bar{Q}_1} \quad (3)$$

4. Blade Surface Pressure Coefficients

$$C_{p1} = \frac{P_s - \bar{P}_1}{\bar{Q}_1} \quad (4)$$

$$C_{p2} = \frac{P_s - \bar{P}_2}{\bar{Q}_2} \quad (5)$$

5. Dimensionless Velocity

$$X = \frac{V}{V_t} = \frac{V}{\sqrt{2C_p T_t}}$$

where V is the local velocity, $V_t = \sqrt{2C_p T_t}$ is the "limiting" velocity and T_t is the stagnation temperature.

APPENDIX D

INSTRUCTIONS FOR PREPARING INPUT AND OPERATING QSONIC USING A RECTILINEAR CASCADE CONFIGURATION

D.1 BACKGROUND INFORMATION

QSONIC has the capability to calculate an axial, mixed or radial flow field and the test cascade can be rotating or stationary. The geometry of the streamsurface can be a 2D planar cascade or axisymmetric with varying channel thickness and radial position. The capabilities of QSONIC, beyond those of previous cascade analysis methods (such as described in Ref. 8) include the ability to calculate through weak shocks with a peak relative Mach number less than 1.4, and completely around both leading and trailing edge regions of a blade profile. The blade shapes in the leading and trailing edge regions are not restricted to circular arcs. Detailed instructions for preparing input for a configuration other than an axial flow, stationary and rectilinear cascade may be found in Ref. 9. What follows are instructions for preparing the input applicable to the Rectilinear Cascade facility and running QSONIC on the Naval Postgraduate School's computer and associated operating system. It is assumed that the reader has a working knowledge of the NPS computer operating system and is familiar with Refs. 13 and 14.

QSONIC operates in two parts. The first part generates a body (blade) centered mesh (geometry generation). The second part actually solves for the flow conditions at points in the mesh (flow solution). The data necessary to generate a mesh consists of a two-dimensional description of the blade shape. This is in the form of pairs of (X,Y) points on the surface, together with parameters that describe the cascade layout, such as chord and stagger angle. Additionally, parameters describing the density of mesh lines complete the input for the geometry generation.

For flow field calculations, the upstream flow conditions, convergence criteria and a schedule of meshes to be used should be input. If quasi-three dimensional effects are to be considered, a data file containing a description of the streamchannel's radial thickness and position as a function of distance along the stream surface is needed. For the case described herein, this was input by assuming a linear reduction in streamchannel thickness using a factor of $1/AVDR$. This gave excellent results. (The output of another NASA code, Meridl [Ref. 15], can be used to input data to QSONIC for compressor flow field calculations. This program has recently become operational on the NPS computer.)

The output of QSONIC consists of listings which contain an echo print of the input data, generated mesh coordinates on the blade surface, progress reports on the flow convergence and a list of the final velocities, pressures

and densities on the blade surface for each grid that was included in the schedule of solution meshes.

D.2 INPUT DESCRIPTION

The input for QSONIC falls into the following categories:

Logical and case control parameters (NAMELIST PARAMS)

Bulk data input:

For geometry generation runs (NAMELIST INSTUFF)

For flow solution runs:

Mesh point storage files

Streamchannel data file (for quasi-3D)

The following is a description of the logical and case control parameters for the Rectilinear Cascade. Except for TITLE, the format for all these variables is in namelist form. The namelist is PARAMS. This information is taken from Ref. 9 and adapted to the Cascade Wind Tunnel. Namelist variables can be entered in any order. If a default value is listed, it is not necessary to enter that particular variable. If the default value is listed as none, then a value for that parameter must be input.

The following parameters apply to both the mesh generation and the flow solution runs, but the values need not be the same.

<u>Variable Name</u>	<u>Type</u>	<u>Default Value</u>	<u>Description</u>
TITLE	Alphameric	None	This is a one-line name for the case being run. TITLE must appear on the first line of the data file that will be referred to as NAMELIST DATA.
NOFLOW	Logical	False	NOFLOW = .TRUE. if a run is to stop after generating a mesh, such as the mesh generation run. NOFLOW = .FALSE. for the flow solution run.
MS	Integer	None	MS is an array (max. dimension 10) of values (max. value = 25) for the number of grid lines in the mesh that will enclose the blade. 25 is a satisfactory number for blades of solidity near unity. As solidity increases the maximum value in MS should decrease.
NOZES	Integer	None	NOZES is an array (max. dimension 10) of values (max. value = 49) for the number of grid lines in the mesh radiating from the blade on one surface. If a value of NOZES is greater than zero, a mesh with that many lines will be developed and stored in the file MESHGEN DATA. If the value is negative, QSONIC assumes that this MESH already exists in the file MESHGEN DATA and will be read in. For Geometry Generation runs NOZES > 0 and for flow solution runs NOZES < 0. For electrostatic analog grid generator, NOZES must be odd.

BETA1 Real None

BETA1 is the flow angle at the upstream boundary, in degrees. BETA1 is measured from the aerodynamic chordline to the direction of flow; clockwise is negative. This can be obtained from the output of the cascade tunnel data reduction program 'CX4431'. It is listed as "inlet air angle." Because of a difference in definition, it is necessary to use the usual inlet air angle minus the stagger angle for BETA1.

Example: The output of CX4431 lists an inlet air angle of 42.429°. The cascade is configured with a stagger angle of 14.27°.

$$\begin{array}{r} 42.429^\circ \\ - 14.27 \\ \hline 28.16 = \text{BETA1} \end{array}$$

BETA2 Real None

BETA2 is the flow angle at the downstream boundary in degrees. It is measured from the aerodynamic chordline to direction of flow. Clockwise is negative. This can be obtained from the output of the cascade tunnel data reduction program 'CX4431'. It is listed as outlet air angle. Because of a difference in definition, it is necessary to use the outlet air angle minus the stagger angle for BETA2.

GAMMA Real 1.4

This is the ratio of specific heats. For the Subsonic Cascade Wind Tunnel the default value works well.

TOLS Real None

This is an array of dimension 10 of tolerances corresponding to MS and NOZES. Each grid solution will proceed until its TOLS value is satisfied.

MESH1	Integer	1	This is an index in the arrays MS and NOZES of the first mesh to be generated and/or used for the flow solution. For geometry generation runs, MESH1 selects one of the grids to be stored, provided NOZES(MESH1) > 0.
MESHN	Integer	None	This is an index in the arrays MS and NOZES of last grid to be calculated for geometry generation runs or used for the flow solution. For geometry generation runs, MESHN = MESH1. Subsequent flow solution runs then solve the case for all grids listed in MS and NOZES between index MESH1 and MESHN. For flow solution MESH1 normally points to the coarsest mesh. To insure equal spacing of grid lines over the entire mesh, $\frac{\text{NOZES}(I) - 1}{K(I - \text{MESH1})} = \text{Integer},$ where $\text{MESH1} \leq I \leq \text{MESHN}$ and $K = 2, 3, 4, \dots$ ($K = 2$ if grid lines are doubled between successive grids).
LAMDA0	Real	None	Stagger angle of the blade row in degrees measured from the throughflow direction to the blade chord line. (Clockwise is negative.)
CHORD	Real	None	True chord of the blade, in the same units as the blade coordinates.
S	Real	None	Blade spacing in the same units as the blade coordinates.

The following parameters are required only for the geometry generation run. QSONIC gives the user a choice of two grid (mesh) generators. The Electrostatic Analog grid generator is applicable to any blade shape and any value of stagger or turning. The interpolation scheme grid generator works for most blades except those with sharp leading edges where the edge radius is less than .5% of chord. The interpolation scheme allows the user to concentrate grid lines in areas of high interest around the blade. With the Electrostatic Analog grid generator no concentration of grid lines is available. For the cascade configuration used in this study, the interpolation scheme provided gross errors in the flow solutions, so the Electrostatic Analog grid generator was used with good results.

<u>Name</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
NED	Integer	None	Total number of body definition coordinates that are input. These are the X,Y points that describe the profile of the blade, with the first point repeated as the last point.
KN	Integer	None	KN is used to indicate which grid generator is to be used. KN = 0 will call the electrostatic analog grid generator. For interpolation scheme, KN = number of body points on upper surface from the minimum to maximum X points, inclusive.

NED and KN are required for either grid generator. If the electrostatic analog grid generator is used, no other

parameters are used. If this grid generator is used the value from NOZES that is used must be odd.

The following parameters are used only if the Interpolation Scheme is used to generate the mesh.

<u>Name</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
RLE	Real	None	RLE is the leading edge radius of the blade, with units the same as the coordinates defining the blade profile.
RTE	Real	None	RTE is the trailing edge radius of blade.
THETL	Real	0	THETL is the camber angle of the leading edge in degrees. It is measured from the aerodynamic chord to the line tangent to the mean camber line at the leading edge; clockwise is positive.
THETT	Real	0	THETT is the camber angle at the trailing edge in degrees. It is measured from the chord line to the mean camber line at the trailing edge. Clockwise is positive.
CAMPER	Integer	6	For blades whose chordline lies outside the blade profile, such as the DCA blading discussed in this report, extra grid lines surrounding the blade are needed to interpolate the blade position. The truncated value of $MS()/CAMPER$ is added to $MS()$. The maximum allowed value of $MS() + MS()/CAMPER$ is 30 grid lines. These are the grid lines that enclose the blade profile.
STABAC	Real	0.999	STABAC is used only for test cases where no blade shape is to be input. Default value is usually adequate.

CHOP	Real	0.99	If leading edge or trailing edge radius is less than 2% of chord consult Ref. 9 for clarification. Normally, $0.9 < \text{CHOP} < 1.0$.
SMOOTH	Logical	False	For automatic addition of more blade definition points in the region of the leading and trailing edges set $\text{SMOOTH} = \text{.TRUE.}$. This should be done for all blades except cusps and wedges.
LEONLY	Logical	False	$\text{LEONLY} = \text{.TRUE.}$ for smoothing about the leading edge only; this is used if the trailing edge is a cusp or a wedge. SMOOTH must also be .TRUE. .
SLP1	Real	1	These parameters control the concentration of grid lines, if desired. The default values worked well for the DCA blades reported herein. For controlling the amount and location of the concentration consult Ref. 9.
SLP2	Real	2	
SLP3	Real	1	
SLP4	Real	1	

The following logical and case control parameters are required only for flow solution runs.

<u>Name</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
MINF	Real	None	This is the Mach number at the upstream boundary. This can be determined from the non-dimensional velocity X output from the cascade tunnel data reduction program 'Redd 5' and the relationship,

$$\frac{\gamma - 1}{2} M^2 = \frac{X^2}{1 - X^2} \quad \text{which yields}$$

$$M = \left\{ \frac{X^2}{1 - X^2} \cdot \frac{2}{\gamma - 1} \right\}^{\frac{1}{2}}$$

TOLS(I) is the tolerance for MS(I) and NOZES(I). There are three forms of input permitted.

- A) $-1.0 < \text{TOLS}(I) < 0.0$: Calculations of the flow solution will proceed until the relative circulation error

$$\left| \frac{C_{\text{CALC}}^N - C_{\text{EXACT}}}{C_{\text{EXACT}}} \right| < |\text{TOLS}(I)|$$

TOLS values between -10^{-3} and -10^{-6} are typical values for grids. This method of input is appropriate only for lifting (non-symmetric blades) cases.

- B) $0.0 < \text{TOLS}(I) < 1.0$: Calculations of the flow solution will proceed until the average relative change in potential is less than the absolute value of TOLS(I).

$$\left(\frac{\delta \phi}{\phi} \right)_{\text{AVE}} < |\text{TOLS}(I)|$$

Typical values should be between 10^{-3} and 10^{-5} .

- C) $1.0 < \text{TOLS}(I)$: Calculations proceed until the number of iterations equals TOLS(I).

Regardless of TOLS input, the solution for each grid will stop after 300 iterations if the TOLS criteria has not yet been made. All three forms were used for the cascade configuration reported herein with no discernible differences in results.

OVEREL	Real	1.5	The default values of these parameters are adequate for flow conditions in the subsonic cascade wind tunnel.
UNDERL	Real	1.0	
SUPREL	Real	1.0	
NOWREL	Integer	20	
NOTYET	Integer	2	
TEGARD	Real	2.0	
DAMP	Real	1.0	
CII	Real	0.2	
IT	Integer	10	Number of iterations between intermediate printouts of residuals and Mach number. The information controlled by this parameter is of limited value in comparing with measured data, so a value greater than 10 reduces the amount of computer printout. For the study reported herein 40 was used.
ALLOUT	Logical	.FALSE.	To list the flow quantities at all grid points in the last mesh set ALLOUT = .TRUE.. Unless a very coarse grid is used, the output resulting from ALLOUT = .TRUE. would be extremely voluminous and of limited value. Until the cascade is configured so it is possible to take data from between the blades, ALLOUT should be .FALSE..
QUASI3	Logical	.FALSE.	QUASI3 = .TRUE. to activate streamchannel thickness and/or radius variations. The cascade wind tunnel has no radius variations, but to simulate 3-D effects the streamchannel thickness is reduced at the exit boundary by a factor of 1/AVDR. This data is placed in a file used by QSONIC if a quasi 3-D solution is desired.
NSTRM	Integer	1	This is the position of desired streamsurface data on the streamchannel file used if QUASI3 = .TRUE.. Currently the default value of 1 identifies the proper streamsurface data

in the streamchannel file. If the output from the NASA code 'Meridl' is used for the streamchannel data, then by using different values of NSTRM, different streamsurface data may be used.

RINF Real 1.0

This is the spanwise radius at the upstream boundary divided by aerodynamic chord. Radius effects are activated if RINF $\neq$ 1.0. The current version of QSONIC allows the following cases.

	<u>QUASI</u>	<u>RINF</u>	<u>Results</u>
1)	.FALSE.	1.0	Planar 2D flow
2)	.TRUE.	$\neq$ 1.0	Thickness on file; radius on file.
3)	.TRUE.	1.0	Thickness on file; constant radius.

Only 1) and 3) apply to the cascade wind tunnel.

WAKE Real 0.0
 MINF2 Real 10.0
 OMEGA Real 0.0
 VAXIAL Real 999.0
 FLOCO Real 999.0

These parameters apply only if the test cascade is rotating and/or the downstream Mach is near 1.0.

At this point all of the logical and case control parameters necessary to use QSONIC for the flow conditions possible in the subsonic cascade wind tunnel have been discussed. The following is a description of the Bulk Data input for the geometry generation run and the flow solution run.

The format for all variables in the bulk data for mesh generation (geometry) is namelist form. The namelist is INSTUFF.

<u>Name</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
H2	Complex	None	This is a table of points defining the blade profile. The real part = X, and imaginary part = Y coordinate. The table begins at the point of maximum X value at the trailing edge and proceeds clockwise back around to the first point, which is repeated. The blade must be at the stagger angle and the origin at the point of minimum X. For Electrostatic Analog grids, the stagger angle must be positive (leading edge low, trailing edge high). The maximum number of points in H2 is 99 for the interpolation scheme or 63 for the Electrostatic Analog.
BUG2	Logical	.FALSE.	BUG2 = .TRUE. for a more detailed output of geometry generation. This will include the X,Y coordinates that define the mesh as well as second derivatives at grid points on the body. Except for trouble shooting this data is of limited value at the present time since there is no graphic output.

The bulk data for flow solutions consists of a mesh file and the streamchannel data file if quasi-3D effects are to be calculated. The mesh file is created by QSONIC during the mesh (grid) generation run. No further inputs are required from the user for the mesh file.

The streamchannel data file must contain a table of streamtube thicknesses, radial positions and corresponding X values along the streamsurface.

<u>Name</u>	<u>Type</u>	<u>Default</u>	<u>Description</u>
CHO	Real	None	CHO is the aerodynamic chord multiplied by the cosine of LAMDA0. (LAMDA0 = stagger angle)
NRSP	Integer	None	NRSP is the total number of data points in each of the tables of thickness, radial position, and X location. If NRSP is 2, a linear distribution is obtained between the endpoints given. NRSP = 2 was used for the study reported herein with good results.
RM	Real	None	Array of corresponding X locations for thickness and radius data values. X = 0 represents the leading edge of blade, with the blade at stagger angle. The units can be any consistent length scale common to CHO, RM, RMSP and BESP. Inches were used in this study.
RMSP	Real	None	Spanwise radial positions of streamsurface at the X locations given in RM. RMSP was not used in the current study.
BESP	Real	None	Array of streamtube thickness values at the X locations specified in RM. For the study reported herein, at X = 0 a streamtube thickness of 1.0 was arbitrarily selected. The streamtube contraction through the test section was simulated by reducing the thickness at X = 0, by a factor of 1/AVDR at the trailing edge. Duval [Ref. 3] explains AVDR.

D.3 PREPARING INPUT FILES

QSONIC was originally configured to use several input/output devices while reading data and generating output. The input/output devices are listed below as used by QSONIC.

<u>I/O Unit</u>	<u>Usage</u>
2	File containing streamchannel data. This is used only if QUASI = .TRUE..
5	Standard card input; NAMELISTS PARAMS, INSTUF.
6	Standard printed output.
13	For mesh generation runs, coordinates of all mesh points are written here. For flow solutions, X, Y, velocities, pressures and MACH are recorded for graphic display. The program currently has no graphic output capability. No user action is necessary to create this file.
18	Used as temporary storage. No output is stored here. No user action is necessary in conjunction with this file.
23	Previously developed mesh coordinates are read in from this file during the flow solution. After a mesh generation run, the user must create this file and put in it the data from I/O unit 13, so that during the flow solution QSONIC can read in the mesh points.

QSONIC is presently configured to operate with the CMS system of the IBM 370 computer. This system provides a high degree of flexibility in parameter selection. With this system, all the input/output units previously mentioned are on the disk space assigned to the Turbopropulsion Laboratory (TPL). Access to QSONIC and the TPL disk space can be obtained through the Director of TPL.

The first step in using QSONIC is the creation of the data file necessary for the mesh generation run. This is done using the standard procedures of the XEDIT function of the CMS operating system. Reference 14 has specific instructions for creating new files. The filename and file-type for the data used in this study was NAMELIST GFOMD. Once the data file is opened, the necessary data is input beginning at column 2 of the virtual card. Since the variables are in namelist form, they can be input in any order.

Table D.1 is an example of the data file necessary for the mesh generation run. The TITLE must appear on the first line (FORMAT = 20A4). After the TITLE, the logical and case control parameters are input after the namelist &PARAMS. When all the case control parameters required are input the PARAMS namelist is closed with &END. On the next line of the data file the bulk data for the mesh generation run is input in namelist form, with the namelist &INSTUF. The $H1 = 100*(0.0,0.0)$ that appears after &INSTUF on Table D.1 was used on earlier versions of QSONIC, but is not used in the present version. It should, however, appear in the data file before the H2 variables (X,Y coordinates defining the blade profile).

At this point, some discussion of the coordinates defining the blade profile is warranted. Table D.2 is a listing of the X and Y coordinates of the DCA blading used in this study. Figure D.1 is a plot of these coordinates.

Recall that the coordinates defining the blade profile for QSONIC must be for the blade at the stagger angle and minimum X at the origin. The coordinates of Table D.2 were translated and rotated using a coordinate transformation routine for the HP-67 programmable calculator. These new coordinates appear in the namelist INSTUFF on Table D.1. Figure D.2 is a plot of the translated and rotated coordinates. It is highly recommended that such a plot be made for any new blade profiles, to ensure that the original coordinates are translated and rotated properly.

The second step in using QSONIC is the creation of the data files necessary for the flow solution run. The file used for the flow solution in this report is on the TPL disk space with a filename/filetype of NAMELIST FLOWD. The simplest way to open this data file is to use the XEDIT function, as discussed in Ref. 14, to start a new file. Then input the same data as is in the data file for the mesh generation run using the XEDIT subcommand GET (filename) (filetype). The appropriate changes and additions can then be made to this file. Table D.3 is an example of the data file just discussed.

Two more data files are required for the flow solution. The data for one of these is created by the mesh generation run. The other file contains the streamchannel thickness data for implementing quasi-3D effects.

After the mesh generation run, a file with the filename/filetype MESHGEN DATA will appear on the disk. Create a new file with the filename/filetype MESHIN DATA. This is most easily done by issuing the command 'XEDIT MESHIN DATA'; then use 'GET MESHGEN DATA'. This file contains the previously developed mesh coordinates.

The streamchannel data file should have the filename/filetype of DATA5D DATA. The format for the data file is shown below.

<u>Virtual Card</u>	<u>Column No.</u>	<u>Variable Name</u>
1	BLANK	
2	BLANK	
3	21-30	CHO
4	BLANK	
5	36-40	NRSP
6	BLANK	
7	BLANK	
8		
9		
10		
11		
12	1-80 (8F10.5)	RM
↓		
As needed		
↓		
	1-80 (8F10.5)	RMSP (not used in this study)
As needed		
	1-80 (8F10.5)	BESP

Table D.4 is an example of the data file used in the present study. Since NRSP = 2 was used for this study a

linear distribution is assumed for the streamtube thickness values and only 2 values of RM and BESP are required; therefore, only 1 virtual card was required for each array.

D.4 PROGRAM OUTPUT

The output generated by QSONIC for the geometry generation run includes a printed listing (I/O unit 6) and a mesh point file (I/O unit 13). The printed listing under the CMS system I/O unit 6 is normally the computer terminal unless the command 'FILEDEF 06 PRINTER' has been invoked. It is unusual for the program to run properly the first time, so initially it is helpful to have the printed listing appear at the terminal. Once the program is running properly the output should be sent to the line printer.

The flow solution run output consists of a printed listing (I/O unit 6) and a plot data save file (I/O unit 13). Once the flow solution is running properly the printed listing should be sent to the line printer.

Table D.5 is an example of the output generated by the geometry generation run. Figure D.3 is a plot of the grid output points on the blade surface, horizontal chord, produced by the mesh generation run. Figure D.4 is a plot of the grid output points with the blade at the stagger angle. Table D.6 is an example of the output generated by the flow solution.

A detailed explanation of the printed output for the program QSONIC may be found in Ref. 9.

D.5 RUNNING THE PROGRAM

The files on the TPL disk space that apply to QSONIC are listed below:

QSONIC EXEC	A1
QSONIC FORTRAN	A1
QSONIC TEXT	A1
NAMelist GEOM	A1
NAMelist FLOW	A1
NAMelist GEOMD	A1
NAMelist FLOWD	A1
DATA5 DATA	A1
DATA5D DATA	A1

QSONIC EXEC sets the input/output devices required to read and store data. QSONIC FORTRAN is the source program. To document the changes to QSONIC necessary to use the code with the IBM 370 operating system and serve as a reference for future users, a program listing is included at the end of this appendix. QSONIC TEXT is the computer executable code created when QSONIC FORTRAN is compiled. NAMelist GEOM and NAMelist FLOW are the data files for the geometry generation and flow generation respectively for the example in Ref. 9. DATA5 DATA is the streamchannel data required for the quasi-3D solution for the example in Ref. 9.

NAMelist GEOMD is the data file for the geometry generation for the DCA blading used in the study reported herein. NAMelist FLOWD is the file for the flow solution for the cascade configuration used in this study.

QSONIC expects the input data to be in a file on the TPL disk space named NAMelist DATA. Since the first time QSONIC is run is to develop the body centered mesh, the file NAMelist GEOMD must be renamed NAMelist DATA, using procedures specified in Ref. 13. Because QSONIC requires large amounts of virtual memory, extra storage must be defined for the code to operate. This is accomplished by issuing the command 'DEFINE STORAGE 1504K'.

With the data file renamed and more storage defined, type 'QSONIC' to load the program. The output will appear on the terminal screen unless FILEDEF 06 PRINTER was invoked prior to loading the program.

After the mesh generation is complete, rename NAMelist DATA to NAMelist GEOMD and change NAMelist FLOWD to NAMelist DATA. Create a data file with the filename/filetype MESHIN DATA. The elements of this file are the same as the elements in the file MESHGEN DATA that was created by the mesh generation run. The necessary input/output files are now configured for a flow solution run. Issue the command 'QSONIC' to begin execution.

If the program output appears at the terminal it is possible to have some I/O error messages appear with the

output. This is because the write statements in QSONIC are formatted for the 132 character long line of the printer. These errors do not affect the validity of the program output.

The explanation for any error or condition message generated by QSONIC can be found in Ref. 9.

D.6 QSONIC UPDATE

Recently an improved version of QSONIC was reported by NASA Lewis Research Center [Ref. 13]. The new version requires less virtual memory and executes approximately 30% faster than the version presently in use at NPS. Also, the output appears in a different format than is described in this appendix. Reference 18 describes the most recent version of QSONIC in detail.

TABLE D.1. INPUT DATA FOR MESH GENERATION RUN OF QSONIC

```

NASA DCA BLADES, GEOMETRY GEN 42.4,5.3
&PARAMS NOFLOW=.TRUE.,RESTAR=.FALSE.,REMESH=4,
MS=3,5,10,15,NOL=9,13,25,49,MESH1=4,MESHN=4,LAMDAO=14.27,
CHORD=5.0,IS=3.0,NED=51,KN=0.44,RIE=0.44,
THETL=-45.72,THEIT=45.72,CAMPER=5,STABAC=0.999,CHOP=0.99,
SMOOTH=.TRUE.,LEONLY=.FALSE.
&INSTUF
H1=100*(0.91E+01,0.01,0.48095E+01,0.12066E+01)
H2=(0.4226E+01,0.01,0.12096E+01,0.12096E+01)
(0.4228E+01,0.01,0.11793E+01,0.11793E+01)
(0.3381E+01,0.01,0.11232E+01,0.11232E+01)
(0.2939E+01,0.01,0.10467E+01,0.10467E+01)
(0.2503E+01,0.01,0.020745E+01,0.020745E+01)
(0.1862E+01,0.01,0.14402E+01,0.14402E+01)
(0.1232E+01,0.01,0.82E-01,0.82E-01)
(0.6158E-01,0.01,0.882E-01,0.882E-01)
(0.1801,0.01,0.5135,0.5135)
(0.1569E+01,0.01,0.3630E+01,0.3630E+01)
(0.1989E+01,0.01,0.11486E+01,0.11486E+01)
(0.2418E+01,0.01,0.13782E+01,0.13782E+01)
(0.2856E+01,0.01,0.14363E+01,0.14363E+01)
(0.3303E+01,0.01,0.14354E+01,0.14354E+01)
(0.3763E+01,0.01,0.14370E+01,0.14370E+01)
(0.4233E+01,0.01,0.12947E+01,0.12947E+01)
(0.4714E+01,0.01,0.12947E+01,0.12947E+01)
(0.4855E+01,0.01,0.12947E+01,0.12947E+01)
BUG2=
&BEND

```

TABLE D.2. TEST BLADE COORDINATES

X-COORD.	Y-PRESS.	Y-SUCT.
-0.044	0.000	0.000
-0.021	-----	0.039
0.013	-0.042	-----
0.178	0.007	0.142
0.400	0.067	0.244
0.622	0.120	0.333
0.844	0.164	0.413
1.067	0.207	0.480
1.289	0.242	0.538
1.511	0.271	0.584
1.733	0.293	0.620
1.956	0.309	0.649
2.178	0.320	0.664
2.399	0.324	0.673
2.622	0.324	0.671
2.844	0.318	0.660
3.066	0.304	0.640
3.288	0.284	0.607
3.511	0.260	0.567
3.732	0.229	0.515
3.955	0.191	0.453
4.177	0.147	0.380
4.400	0.098	0.298
4.621	0.040	0.200
4.844	-0.022	0.091
4.908	-0.042	-----
4.943	-----	0.040
4.966	0.000	0.000

TABLE D.4. DATA FILE FOR QUASI-3D SOLUTION

STREAMLINE DATA FOR DCA1

4.8554

2

0.0	4.85500
1.00000	0.98476

TABLE D.5 (Continued)

NUMBER OF BODY POINTS		PITCH		CHD	
ITER COUNT FOR INTERNAL GRID POINTS		72 ERR=		0.93749E-05	
GRID OUTPUT POINTS ON BLADE SURFACE, CHORD AT ST					
I	X	Y			
0	0.46779E+00	0.13282E+00			
1	0.46963E+00	0.12825E+00			
2	0.46774E+00	0.12400E+00			
3	0.46393E+00	0.12059E+00			
4	0.45698E+00	0.11813E+00			
5	0.44414E+00	0.11793E+00			
6	0.42698E+00	0.11879E+00			
7	0.40705E+00	0.11922E+00			
8	0.38426E+00	0.11939E+00			
9	0.35902E+00	0.11886E+00			
10	0.30367E+00	0.11774E+00			
11	0.27447E+00	0.11613E+00			
12	0.24441E+00	0.11402E+00			
13	0.21444E+00	0.11299E+00			
14	0.18384E+00	0.11078E+00			
15	0.15318E+00	0.10785E+00			
16	0.12266E+00	0.10385E+00			
17	0.92137E-01	0.09330E-01			
18	0.61648E-01	0.04288E-01			
19	0.00000E+00	0.00000E+00			

NUMBER OF BODY POINTS		PITCH		CHD	
ITER COUNT FOR INTERNAL GRID POINTS		72 ERR=		0.93749E-05	
0	0.9004390	0.0000000	0.0000000	0.0000000	0.0000000
1	0.8548302	0.0000000	0.0000000	0.0000000	0.0000000
2	0.8092813	0.0000000	0.0000000	0.0000000	0.0000000
3	0.7641715	0.0000000	0.0000000	0.0000000	0.0000000
4	0.7191616	0.0000000	0.0000000	0.0000000	0.0000000
5	0.6748901	0.0000000	0.0000000	0.0000000	0.0000000
6	0.6305587	0.0000000	0.0000000	0.0000000	0.0000000
7	0.5866467	0.0000000	0.0000000	0.0000000	0.0000000
8	0.5430135	0.0000000	0.0000000	0.0000000	0.0000000
9	0.4997607	0.0000000	0.0000000	0.0000000	0.0000000
10	0.4566267	0.0000000	0.0000000	0.0000000	0.0000000
11	0.4140718	0.0000000	0.0000000	0.0000000	0.0000000
12	0.3716765	0.0000000	0.0000000	0.0000000	0.0000000
13	0.3293214	0.0000000	0.0000000	0.0000000	0.0000000
14	0.2874650	0.0000000	0.0000000	0.0000000	0.0000000
15	0.2459480	0.0000000	0.0000000	0.0000000	0.0000000
16	0.2047305	0.0000000	0.0000000	0.0000000	0.0000000
17	0.1636926	0.0000000	0.0000000	0.0000000	0.0000000
18	0.1229141	0.0000000	0.0000000	0.0000000	0.0000000
19	0.0825748	0.0000000	0.0000000	0.0000000	0.0000000
20	0.0425948	0.0000000	0.0000000	0.0000000	0.0000000
21	0.0130938	0.0000000	0.0000000	0.0000000	0.0000000

TABLE D.5 (Cont'd)

[illegible][illegible]

TABLE D.6 (Continued)

RELAXATION BEGINS ON GRID OF 3 SURFACE CONTOURS INTERSECTED BY 17 RADIATING LINES									
SUPERSONIC RELAXATION PARAMETER, SUPREL= 0.00000E+01									
ITERATION	PHI CORRECTN	RESIDUAL AT P. E.	PAX MACH CN BLADE	CALCULATED CIRCULATION	RELATIVE CIRC ERROR	SUBSONIC RELAX. FAC.	SUPERSONIC RELAX. FAC.		
40	0.131149E-01	0.889149E-02	0.313030E+00	-0.203196E-01	-0.105212E+01	0.150000E+01	0.100000E+01		
80	0.131149E-01	0.889149E-02	0.313030E+00	-0.203196E-01	-0.105212E+01	0.150000E+01	0.100000E+01		
120	0.131149E-01	0.889149E-02	0.313030E+00	-0.203196E-01	-0.105212E+01	0.150000E+01	0.100000E+01		
160	0.131149E-01	0.889149E-02	0.313030E+00	-0.203196E-01	-0.105212E+01	0.150000E+01	0.100000E+01		
200	0.131149E-01	0.889149E-02	0.313030E+00	-0.203196E-01	-0.105212E+01	0.150000E+01	0.100000E+01		
240	0.131149E-01	0.889149E-02	0.313030E+00	-0.203196E-01	-0.105212E+01	0.150000E+01	0.100000E+01		
280	0.131149E-01	0.889149E-02	0.313030E+00	-0.203196E-01	-0.105212E+01	0.150000E+01	0.100000E+01		
300	0.131149E-01	0.889149E-02	0.313030E+00	-0.203196E-01	-0.105212E+01	0.150000E+01	0.100000E+01		
FINAL FLOW CALCULATION ON BLADE SURFACE FOR THE 3 BY 17 MESH									
RADIATING LINE NO.	X/CX	Y/CX	X	Y	MACH	STATIC PRES COEF	PHI	XVEL	YVEL
1	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
2	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
3	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
4	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
5	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
6	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
7	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
8	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
9	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
10	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
11	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
12	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
13	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
14	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
15	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
16	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391
17	0.00000	0.26029	0.40563	0.12825	0.22825	-0.27183	1.70549	0.71364	-0.05391

END OF CALCULATION ON GRID 1 OF 4

RELAXATION BEGINS ON GRID OF	5 SURFACE CONTOURS INTERSECTED BY 25 RADIATING LINES								
SUPERSONIC RELAXATION PARAMETER, SUPREL=	0.60000E+00								
RELAXATION BEGINS ON GRID OF	5 SURFACE CONTOURS INTERSECTED BY 25 RADIATING LINES								
SUPERSONIC RELAXATION PARAMETER, SUPREL=	0.60000E+00								
ITERATION	AVERAGE	MAX MACH	RELATIVE	RELATIVE	RELATIVE	RELATIVE	RELATIVE	RELATIVE	RELATIVE
	4PI CORRECTN	EN CLASH	CIRCULATED	CIRC	CIRC	CIRC	CIRC	CIRC	CIRC
	WFS/DUAL								
	WFS/DUAL								
45	-0.180530E-02	0.419742E+00	0.390200E+00	0.942737E-03	0.942737E-03	0.150000E+01	0.600000E+00	0.600000E+00	0.600000E+00
50	-0.180530E-02	0.419742E+00	0.390200E+00	0.942737E-03	0.942737E-03	0.150000E+01	0.600000E+00	0.600000E+00	0.600000E+00
55	-0.180530E-02	0.419742E+00	0.390200E+00	0.942737E-03	0.942737E-03	0.150000E+01	0.600000E+00	0.600000E+00	0.600000E+00
60	-0.180530E-02	0.419742E+00	0.390200E+00	0.942737E-03	0.942737E-03	0.150000E+01	0.600000E+00	0.600000E+00	0.600000E+00
65	-0.180530E-02	0.419742E+00	0.390200E+00	0.942737E-03	0.942737E-03	0.150000E+01	0.600000E+00	0.600000E+00	0.600000E+00
70	-0.180530E-02	0.419742E+00	0.390200E+00	0.942737E-03	0.942737E-03	0.150000E+01	0.600000E+00	0.600000E+00	0.600000E+00
75	-0.180530E-02	0.419742E+00	0.390200E+00	0.942737E-03	0.942737E-03	0.150000E+01	0.600000E+00	0.600000E+00	0.600000E+00
80	-0.180530E-02	0.419742E+00	0.390200E+00	0.942737E-03	0.942737E-03	0.150000E+01	0.600000E+00	0.600000E+00	0.600000E+00
85	-0.180530E-02	0.419742E+00	0.390200E+00	0.942737E-03	0.942737E-03	0.150000E+01	0.600000E+00	0.600000E+00	0.600000E+00
90	-0.180530E-02	0.419742E+00	0.390200E+00	0.942737E-03	0.942737E-03	0.150000E+01	0.600000E+00	0.600000E+00	0.600000E+00
95	-0.180530E-02	0.419742E+00	0.390200E+00	0.942737E-03	0.942737E-03	0.150000E+01	0.600000E+00	0.600000E+00	0.600000E+00
100	-0.180530E-02	0.419742E+00	0.390200E+00	0.942737E-03	0.942737E-03	0.150000E+01	0.600000E+00	0.600000E+00	0.600000E+00

FINAL FLOW CALCULATION ON BLADE SURFACE FOR THE 5 BY 25 MESH

DECLASS

[illegible]

END OF CALCULATION ON GRID 2 OF 4

TABLE D.6 (Continued)

RELAXATION REGIONS CN GRIC OF 15 SURFACE CONTOURS INTERSECTED BY 49 RADIATING LINES									
SUPERSONIC RELAXATION PARAMETER, SUPREL = 0.500000E+00									
ITERATION	AVERAGE	RESIDUAL	MAX MACH	CALCULATION	RELATIVE	SUBSONIC	SUPERSONIC		
AC	PHI CORRECTN	AT P. E.	CN BLADE	CIRCULATION	ERROR	RELAX. FAC.	RELAX. FAC.		
100	0.11558E-03	-0.10744E-03	0.38097E+00	0.38097E+00	0.35070E-03	0.15000E+01	0.50000E+00	0.50000E+00	0.50000E+00
150	0.11558E-03	-0.10744E-03	0.38097E+00	0.38097E+00	0.35070E-03	0.15000E+01	0.50000E+00	0.50000E+00	0.50000E+00
200	0.11558E-03	-0.10744E-03	0.38097E+00	0.38097E+00	0.35070E-03	0.15000E+01	0.50000E+00	0.50000E+00	0.50000E+00

FINAL FLOW CALCULATION ON BLADE SURFACE FOR THE 15 BY 49 MESH									
RADIATING LINE NO.	Y/CX	X	Y	MACH	STATIC PRES CDEF	PHI	XVEL	WVEL	DENSITY
1	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
3	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
4	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
5	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
6	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
7	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
8	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
9	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
10	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
11	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
12	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
13	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
14	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
15	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
16	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
17	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
18	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
19	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
20	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
21	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
22	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
23	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
24	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
25	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
26	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
27	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
28	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
29	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
30	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
31	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
32	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
33	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
34	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
35	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
36	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
37	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
38	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
39	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
40	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
41	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
42	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
43	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
44	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
45	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
46	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
47	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
48	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
49	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

END OF CALCULATION CN GRIC 3 OF 4

RELATION BEGINS ON GRID OF 15 SURFACE COUNTS INTERSECTED BY 97 RADIATING LINES						
ITERATION	AVERAGE FAT CORRECTN	RESIDUAL AFSQUAD	MAX MAGN	CIRCULATION	RELATIVE CTCR RELAX. FAC.	SUPERSATC RELAX. FAC.
55	0.952704E+04	-0.341294E-04	0.381784E+00	0.3894943E+00	0.314974E-04	0.550000E+00
56	0.9566745E+04	-0.2737351E-04	0.382889E+00	0.3894943E+00	0.314974E-04	0.550000E+00

FINAL FLOW CALCULATION ON BLADE SURFACE FOR THE 15 HY 97 MESIN

RAILWAY
LINE AC.

DENSITY
 YVEL
 XVEL
 PHJ
 PRES CUEF
 MACH
 Y
 X
 Y/CX
 P/CX
 AC

END OF CALCULATION CN GRID 4 CF 4

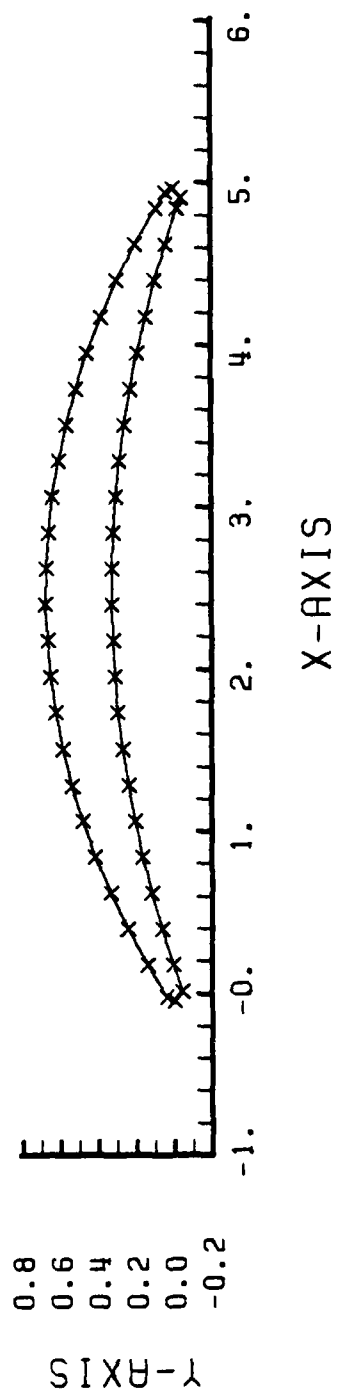


Fig. D.1. Blade Coordinates

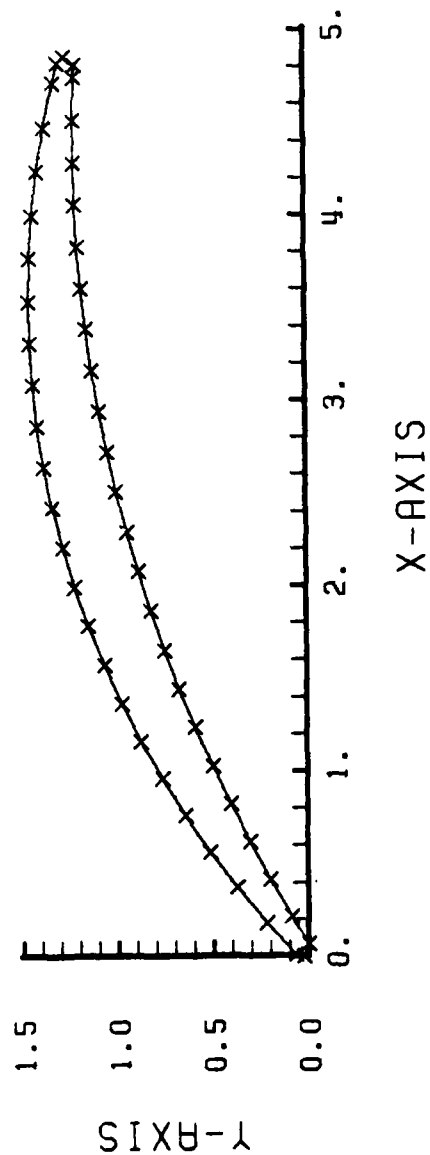


Fig. D.2. Blade Coordinates Translated and Rotated

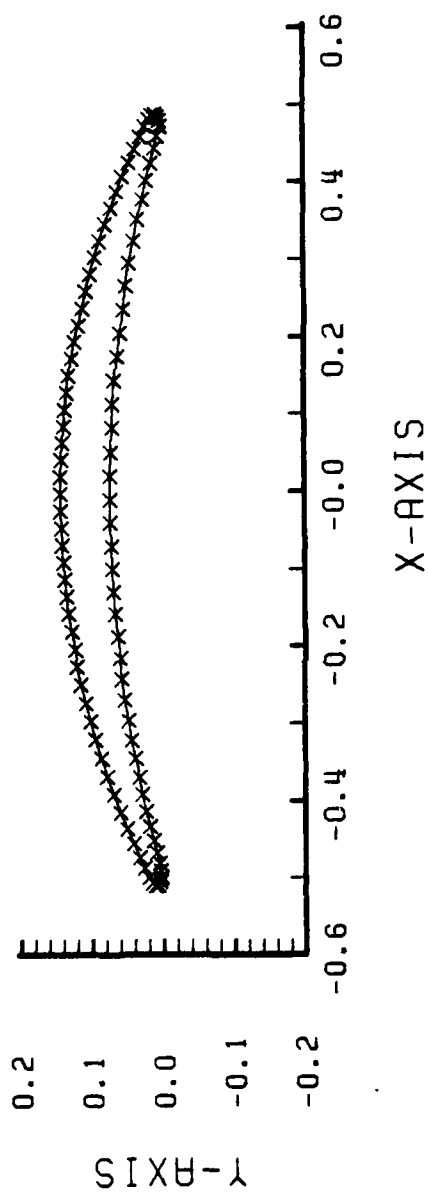


Fig. D.3. Mesh Points on Blade Surface, Horizontal Chord

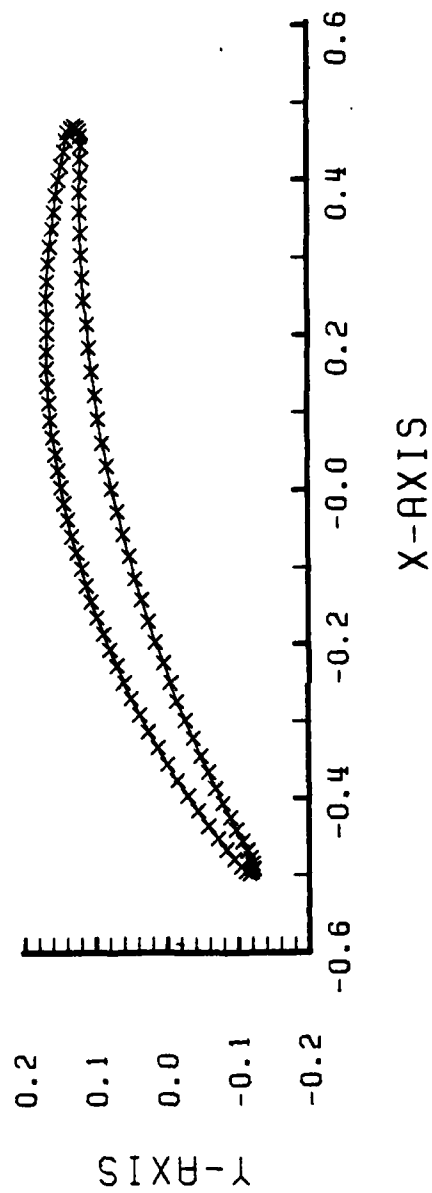


Fig. D.4. Mesh Points on Blade Surface, Chord at Stagger Angle

[illegible]

169

QSD000490
QSD000500
QSD000510
QSD000520
QSD000530
QSD000540
QSD000550
QSD000560
QSD000570
QSD000580
QSD000590
QSD000600
QSD000610
QSD000620
QSD000630
QSD000640
QSD000650
QSD000660
QSD000670
QSD000680
QSD000690
QSD000700
QSD000710
QSD000720
QSD000730
QSD000740
QSD000750
QSD000760
QSD000770
QSD000780
QSD000790
QSD000800
QSD000810
QSD000820
QSD000830
QSD000840
QSD000850
QSD000860
QSD000870
QSD000880
QSD000890
QSD000900
QSD000910
QSD000920
QSD000930
QSD000940
QSD000950
QSD000960

DATA NSTRM,QUASI3,MESH1/1,.FALSE.,1/,REMESH/4/,TEGARD/2./
1,NOFLOW/.FALSE./

DATA QLEVEL/9.1'/

READ(5,3) TITLE

FORMAT(20A4)

WRITE(6,33) QLEVEL,TITLE

FORMAT(1H1,' QSONIC VERSION 1.4.',A4,2X,21A4//)

READ(5,PARAMS)

WRITE(6,2)

FORMAT(10) CASE CONTROL INPUT ECHO '//'

WRITE(6,PARAMS)

BETA1=BETA1*PI/180.

LAMDAO=LAMDAO*PI/180.

BETA2=BETA2*PI/180.

MIN=MS(MESH1)

IF(RESTAR)MESH1=REMESH

NOZIN=IABS(NOZES(MESH1))

IF(QUASI3.AND.B2DB1.EQ.1.)CALL RBIN(NSTRM,LAMDAO)

DO 1 I=MESH1,MESHN

MESH=1

NOLD(2)=NEWM

NOLD(1)=IEXIT

NEWNOZ=NOZES(MESH)

NEWM=MS(MESH)

TOL=TOL3(MESH)

REWIND 23

REWIND 13

CALL WRAPUP(ZETA,ETA,S)

IEXIT=2*(NEWNOZ-1)

CALL WHEEL(LAMDAO,NEWM,IEXIT,NOFLOW)

IF(NOFLOW)STOP

IF(QUASI3)CALL FILLRB(NEWM,IEXIT,RINF,LAMDAO)

IF(RESTAR.OR.MESH.NE.MESH1)CALL FILPHI(NEWNOZ,NEWM,IEXIT)

IF(SLP1.GE.0.)SLP2=TEGARD

CALL USONIC

MYMESH=MESH-MESH1+1

MYMAX=MESH-MESH1+1

WRITE(6,4) MYMESH,MYMAX

FORMAT(1H0,40X,' END OF CALCULATION ON GRID',12,' OF',12)

```

1      CONTINUE
      CALL KEEPER
      STOP
      END
      BLOCK DATA

C      INITIALIZE ALL DATA IN COMMON

      LOGICAL RESTAR,LEONLY
      1,BUG,BUG2,BUG3,SMOOTH,FINEST,ALLOUT
      INTEGER CAMPER
      REAL MINF,LAMDAO
      DOUBLE PRECISION CAPK,CAPKP
      COMMON/MESHES/NOZES(10),MS(10),TOLS(10),NOLD(2)
      COMMON/PARAM/RESTAR,NOZIN,BETA1,BETA2,QINFI,C11,GAMMA,MINF,EM
      1,DELTA,BETA,TOL,BUG,BUG2,BUG3,IF,GUESS,OVEREL,TIMER,NOWREL,NOTYET
      1,DAMP,SUPREL,S,WAKE,B2OB1,UNDERL,RADRAI,RADLE,OMEGA,FLOCO,VAXIAL
      1,ALLOUT,KKMAX
      COMMON/QUASI/R(100,30),B(100,30)
      COMMON/ENTIRE/ X(100,30),Y(100,30),PHI(100,30),DELPSV(100,30)
      COMMON/GEOM2/ZETA(100),ETA(100)
      COMMON/GEOM/NEWM,RLE,RTE,A,CHORD,XUPS,XDNS,LAMDAO,CC,CAPK,
      1,CAPKP,PI,RNK,VB,KN,NED,BB,DOSURF,RFAC,SOLID,MPLS
      1,STABAC,SLP1,SLP2,SLP3,SLP4,SMOOTH,CHOP,THEIT,THETL
      1,FINEST,LEONLY,CAMPER

      COMMON/CALVEL/D55(55),PRIORS(14,30),INDEX(100,30),IINDEX(32),
      1,FAKEU(100),FAKEV(100),ICOUNT,INDEX,AMY,AMZ

      DATA RADRAI/1./,WAKE,820B1/0./,1./,R,B/6000*1./
      DATA RADLE/1./,OVEREL/1.5/,TOL/.001/
      1,PI/3.14159265/,TIMER/1./
      1,OMEGA,FLOCO,VAXIAL/0./,999./,0./
      1,SUPREL/1./,CHOP/.99/,THEIT,THEIL/0./,CAMPER/6/,EM/10./
      1,DATA NOTYET/2/,DAMP/1./,STABAC,SLP1,SLP2,SLP3,SLP4,SMOOTH/.999,
      1,1./,2./,2*1./,FALSE./,UNDERL/1./,LEONLY/.FALSE./,FINEST/.FALSE./
      DATA A/1.5/,C11/.2/,C1/1./,GUESS/10./,IT/10/,ALLOUT/.FALSE./
      DATA KKMAX/300/
      DATA DELTA,BETA,QINFI,BUG,RESTAR,GAMMA/3*1./,FALSE./,1.4/

      DATA IINDEX,IINDEX/203*0,1,98*0,2,1,96*0,2,2,4,1,96*0,1,1,1,2497*0,
      1,2,8,2,8,1,1,8,3,3,3,1,1,1,8,2,8,1,1,8,3,3,1,1,1,8,2,8,2,8,
      1,END

```

QSO01450
QSO01460
QSO01470
QSO01480
QSO01490
QSO01500
QSO01510
QSO01520
QSO01530
QSO01540
QSO01550
QSO01560
QSO01570
QSO01580
QSO01590
QSO01600
QSO01610
QSO01620
QSO01630
QSO01640
QSO01650
QSO01660
QSO01670
QSO01680
QSO01690
QSO01700
QSO01710
QSO01720
QSO01730
QSO01740
QSO01750
QSO01760
QSO01770
QSO01780
QSO01790
QSO01800
QSO01810
QSO01820
QSO01830
QSO01840
QSO01850
QSO01860
QSO01870
QSO01880
QSO01890
QSO01900
QSO01910
QSO01920

```

SUBROUTINE WHEEL(LAMDAO,NEWM,NEW2M1,NOFLOW)
  ROTATE X,Y COORDINATES THRU AN ANGLE LAMDAO(RADIANS)

  LOGICAL NOFLOW
  COMMON/ENTIRE/ X(100,30),Y(100,30),PHI(100,30),DELPVS(100,30)
  COMMON/GEOM2/ZETA(100),ETA(100)
  REAL LAMDAO
  COMPLEX H,A1
  DATA A1/(0.,1.)
  NEW2M=NEW2M1-1
  MM3=NEWM-3
  NNOZ=(NEW2M+1)/2
  NEW2M2=NEW2M1+1
  DO 1 I=1,NEW2M2
    DO 1 J=1,NEWM
      H=CMPLX(X(I,J),Y(I,J))*CEXP(A1*LAMDAO)
      X(I,J)=REAL(H)
      Y(I,J)=AIMAG(H)
      IF(.NOT.NOFLOW) RETURN
      WRITE(6,10)
      FORMAT(1H1,30X,' GRID OUTPUT POINTS ON BLADE SURFACE, CHORD ',
1      ' AT STAGGER ANGLE',/42X,' I X
      DO 2 I=1,NEW2M2
        IIM=I-1
        WRITE(6,14) IIM,X(I,NEWM),Y(I,NEWM)
        FORMAT(4IX,14,2E13.5)
      RETURN
    END
  SUBROUTINE FILPHI(NEWNOZ,NEWM,NEW2M1)
    BASED ON PHI ARRAY FROM PREVIOUS GRID, INTERPOLATES VALUES
    TO FILL THE INITIAL PHI ARRAY FOR NEXT, FINER GRID.

    INTEGER OLDNOZ,OLDM
    LOGICAL RESTAR
    COMMON/ENTIRE/ X(100,30),Y(100,30),PHI(100,30),TOPHI(100,30)
    COMMON/MESHES/ NOZES(10),MS(10),TOLS(10),NOLD(2)
    COMMON/GEOM2/ ZNEW(100),ENEW(100)
    COMMON/CALVEL/FAKEFI(4,6),FAKFI(4,6),RHOCON,QLIM,EXPO,ROTATN
1    ROTROT(1M1,1EXIT,1ICOUNT,1IDEX
1    ,FAKEV(75),1FAKEU(75)
    COMMON/PARAM/ RESTAR
    DIMENSION OLDE(100),OLDZ(100)

```

```

OLDNOZ=NOLD(1)
OLDM=NOLD(2)
NZOLD=OLDNOZ/2+1
INDEX(OLDNOZ-2,3)=0
INDEX(OLDNOZ-2,4)=0
INDEX(OLDNOZ-2,5)=0
INDEX(OLDNOZ-2,6)=0
INDEX(OLDNOZ-1,5)=0
INDEX(OLDNOZ-1,6)=0
INDEX(OLDNOZ,5)=0
INDEX(OLDNOZ,6)=0
INDEX(NZOLD-2,3)=0
INDEX(NZOLD-2,4)=0
INDEX(NZOLD-2,5)=0
INDEX(NZOLD-2,6)=0
INDEX(NZOLD-1,5)=0
INDEX(NZOLD-1,6)=0
INDEX(NZOLD,5)=0
INDEX(NZOLD,6)=0
INDEX(NZOLD+1,4)=0
INDEX(NZOLD+1,5)=0
INDEX(NZOLD+1,6)=0
INDEX(NZOLD+2,3)=0
INDEX(NZOLD+2,4)=0
INDEX(NZOLD+2,5)=0
DO 13 J1=1,NEW2M1
DO 13 J1=1,NEWM
TOPHI(I1,J1)=0.
REWIND 8

```

13

```

IF(RESTAR)GOTO10
READ(18) OLDE,OLDZ
DO 1 I1=1,NEW2M1
DO 2 I2=1,OLDNOZ
IF(OLDE(I2).GT.ENEW(I1)) GOTO3
CONTINUE
I2=OLDNOZ
DO 1 I3=1,NEWM
DO 5 I4=1,OLDM
IF(OLDZ(I4).GT.ZNEW(I3))GOTO6
CONTINUE
I4=OLDM
FACE=(ENEW(I1)-OLDE(I2))/(OLDE(I2)-OLDE(I2-1))
TERP3=
1PHI(I2-1,I4)+(PHI(I2,I4)-PHI(I2-1,I4))*FACE

```

2

3

5

6

QSO01930
QSO01940
QSO01950
QSO01960
QSO01970
QSO01980
QSO01990
QSO02000
QSO02010
QSO02020
QSO02030
QSO02040
QSO02050
QSO02060
QSO02070
QSO02080
QSO02090
QSO02100
QSO02110
QSO02120
QSO02130
QSO02140
QSO02150
QSO02160
QSO02170
QSO02180
QSO02190
QSO02200
QSO02210
QSO02220
QSO02230
QSO02240
QSO02250
QSO02260
QSO02270
QSO02280
QSO02290
QSO02300
QSO02310
QSO02320
QSO02330
QSO02340
QSO02350
QSO02360
QSO02370
QSO02380
QSO02390
QSO02400

```

1      TERPI =
      1PHI(12-1,14-1)+(PHI(12,14-1)-PHI(12-1,14-1))*FACE
      1BNP=(TERP3-TERPI)/(OLDZ(14)-OLDZ(14-1))*ZNEW(13)-
      1OLDZ(14-1))+TERPI
      1TOPHI(11,13)=80P
      22
      4      DO 7 I1=1,NEW2M1
      7      DO 7 I3=1,NEWM
      7      PHI(I1,I3)=TOPHI(I1,I3)
      22
      4      FORMAT(4I6,4E13.5)
      4      FORMAT(1H,8E13.5)
      4      WRITE(8,11) PHI
      4      DO 1313 I1=1,NEW2M1
      4      DO 1313 J1=1,NEWM
      4      TOPHI(I1,J1)=0.
      4      RETURN
      1313
      C      IF RESTART, READ PHI FROM THE RESTART FILE
      10      READ(9,11) PHI
      11      FORMAT(6E13.5)
      11      RESTART=.FALSE.
      12      REWIND 9
      12      RETURN
      12      END
      C      SUBROUTINE RBIN(NSTRM,LAMDAO)
      C      FOR QUASI3D CASES, READS UNIT 2 WITH DISTRIBUTIONS OF SPANWISE RADIUS
      C      AND CHANNEL THICKNESS.  FORMAT IS THE SAME AS PROGRAM TSONIC.
      C      REAL LAMDAO
      C      COMMON/MERTDL/ ZMSP(50,2),THSP(50,2),RMSP(50),
      C      1BESP(50),RM(50),NRSP,NSP(2),RTE(2),PLE(2),STGRF,CHO,CNBL,DM
      C      DIMENSION DUMI(50),R(2)
      C      REWIND 2
      C      ISTRM=1ABS(NSTRM)
      C      DO 1 I=1,ISTRM
      C      READ(2,3)
      C      READ(2,3)
      C      READ(2,6) A1,A2,CHO,STGRF
      C      READ(2,3)
      C      FORMAT(1H)
      C      READ(2,4) NBL,NRSP,I1,I2,I3
      C      CNBL=FLOAT(NBL)
      C      3      FORMAT(30X,I5,4I5)
      C      4      DO 10 J=1,2
      C      4      10

```

```

C      READ(2,5) RI,RO,SPLN01
      ISPL1=SPLN01
      WRITE(6,4) I,NRSP,J,ISPL1
      NSP(J)=ISPL1
      RTE(J)=RI
      RTE(J)=RO
      FORMAT(2F10.5,20X,F10.5)
      FORMAT(8F10.6)
      READ(2,6) (ZMSP(IA,J),IA=1,ISPL1)
      READ(2,6) (THSP(IA,J),IA=1,ISPL1)
C      READ X VALUES
      READ(2,6) (RM(IA),IA=1,NRSP)
C      READ CORRESPONDING SPANWISE RADIUS VALUES.
      READ(2,6) (RMSP(IA),IA=1,NRSP)
C      READ STREAMCHANNEL THICKNESS VALUES.
      READ(2,6) (BESP(IA),IA=1,NRSP)
      IF(I1.EQ.1) READ(2,6) (DUM1(IA),IA=1,NRSP)
      IF(I2.EQ.1) READ(2,6) (DUM1(IA),IA=1,NRSP)
      READ(2,4) I4
      IF(NSTRM.LT.0) RETURN
C      TRANSLATE ORIGIN OF X VALUES TO MIDCHORD AND NORMALIZE RADIUS AND
C      THICKNESS TO THE AERODYNAMIC CHORD.
      DO 7 I=1,NRSP
      RM(I)=(RM(I)-CHO/2.)/CHO*COS(LAMDAO)
      RMSP(I)=RMSP(I)/CHO*COS(LAMDAO)
      BESP(I)=BESP(I)/CHO*COS(LAMDAO)
      RETURN
      END
      SUBROUTINE FILLRB(NEWM,NEW2M1,RADLE,STAG)
C      INTERPOLATES ON THE INPUT (OR ASSUMED LINEAR) DISTRIBUTIONS OF RADIUS
C      AND THICKNESS TO FILL ARRAYS OF SPANWISE POSITION AND STREAMCHANNEL
C      THICKNESS AT ALL GRID POINTS.
C      IF RADLE=1., RADIUS IS ASSUMED CONSTANT

```

QS002890
 QS002900
 QS002910
 QS002920
 QS002930
 QS002940
 QS002950
 QS002960
 QS002970
 QS002980
 QS002990
 QS003000
 QS003010
 QS003020
 QS003030
 QS003040
 QS003050
 QS003060
 QS003070
 QS003080
 QS003090
 QS003100
 QS003110
 QS003120
 QS003130
 QS003140
 QS003150
 QS003160
 QS003170
 QS003180
 QS003190
 QS003200
 QS003210
 QS003220
 QS003230
 QS003240
 QS003250
 QS003260
 QS003270
 QS003280
 QS003290
 QS003300
 QS003310
 QS003320
 QS003330
 QS003340
 QS003350
 QS003360

```

COMMON/QUASI/R(100,30),B(100,30)
COMMON/MERIDL/ZMSP(50,2),THSP(50,2),RMSP(50,1)
1 BESPP(50),RM(50),NPSP,NP(2),RTE(2),RLE(2),STGRF,CHD,CNBL,DUMM
COMMON/ENTIRE/X(100,30),Y(100,30),PHI(100,30),DELPV(100,30)
COMMON/PARAM/RESTAR,NOZIN,BRM(17),USELER,BR(6),B2OB1,UNDERL,R2OR1
NEW2M2=NEW2M1+1
N=NRSP
CO=COS(STAG)/2.

C      NOZIN=NEW2M1/2 ON THE FIRST PASS ONLY

      IF(NOZIN.EQ.NEW2M1/2)FIXR=RADLE
      IF(NOZIN.EQ.NEW2M1/2)B21SAV=B2OB1
      IF(RADLE.NE.1..AND.R2OR1.EQ.1)R2OR1=RMSP(NRSP)/RMSP(1)
      IF(B2OB1.EQ.1.)B2OB1=BESP(NRSP)/BESP(1)

      IF(B21SAV.NE.1..OR.NOZIN.NE.NEW2M1/2.OR.USELER.EQ.0..OR.RADLE.EQ.1
      *.)GOTO15

C      FIND L.E. OF BLADE (RM+CO=0.)

      DO 13 I5=1,NRSP
      IF(RM(I5)+CO.GT.0.)GOTO14
      CONTINUE
      I5=NRSP

C      CALCULATE SPANWISE POSITION AT L.E. (FIXR) FROM THE GIVEN RADIUS
C      DISTRIBUTION.
      FIXR=
      1 RMSP(I5-1)-(RMSP(I5)-RMSP(I5-1))*(CO+RM(I5-1))/(RM(I5)-RM(I5-1))

      DO 1 I1=1,NEW2M2
      DO 1 I3=1,NEWM
      IF(B21SAV.NE.1.)GOTO12
      DO 2 I2=1,NRSP
      IF(RM(I2).GT.X(I1,I3))GOTO3
      CONTINUE
      I2=NRSP
      IF(I2.EQ.1)I2=2
      IFACE=(X(I1,I3)-RM(I2-1))/(RM(I2)-RM(I2-1))
      TERPR=RMSP(I2-1)+(RMSP(I2)-RMSP(I2-1))*FACE
      IF(RM(NRSP).LT.X(I1,I3))TERPR=RMSP(NRSP)
      IF(RM(1).GT.X(I1,I3))TERPR=RMSP(1)
      TERPB=BESP(I2-1)+(BESP(I2)-BESP(I2-1))*FACE
      IF(RADLE.NE.1.)R(I1,I3)=TERPR

```

```

      B(I1,I3)=TERPB/BESP(I1)
      GOTO 10
C   IF (B21SAV.NE.1), A LINEAR DISTRIBUTION OF THICKNESS IS ASSUMED OVER
C   3 CHORD LENGTHS.
12   B(I1,I3)=1.+(B20B1-1.)*(X(I1,I3)+1.5)/3.
      IF(RADLE.NE.1.)FIXR=(1.+(R20R1-1.))*(1.5-CO)/3.)*RADLE
      IF(X(I1,I3).LT.-1.5)B(I1,I3)=1.
      IF(X(I1,I3).GT.1.5)B(I1,I3)=B20B1
      IF(RADLE.NE.1.)R(I1,I3)=(1.+(R20R1-1.)*(X(I1,I3)+1.5)/3.)*RADLE
      IF(X(I1,I3).LT.-1.5)R(I1,I3)=RADLE
      IF(X(I1,I3).GT.1.5)R(I1,I3)=R20R1*RADLE

C   SCALE THE TANGENTIAL COORDINATE Y (R*THETA), TO THE LOCAL RADIUS.
C10  Y(I1,I3)=Y(I1,I3)*R(I1,I3)/FIXR
10   CONTINUE
1    CONTINUE
      USELER=FIXR
      RETURN
      FORMAT(2I5,6E10.4)
      END
      SUBROUTINE USONIC

C                                     FULL POTENTIAL FLOW SOLVER
C                                     FOR CASCADE GEOMETRIES

C   ETA = COMPUTATIONAL COORDINATE THAT IS CONSTANT ON LINES
C   ZETA = COMPUTATIONAL COORDINATE THAT IS CONSTANT ON LINES
C   (THESE ARE REVERSED FROM INTERPOLATION GRID GENERATOR)

C   DO (INPUT SPECIFICATION)
C   DO (INITIALIZATION OF FIELD VARIABLES)
      RELAXD=.FALSE.
      DO UNTIL (RELAXD)
        KK=KK+1
        KKK=KKK+1
        IF(KK.GT.NOWREL) OVEREL=RELSAV
        IF(KK.GT.NOTYET) SUPREL=SUPSAV

```

Q5003850
 Q5003860
 Q5003870
 Q5003880
 Q5003890
 Q5003900
 Q5003910
 Q5003920
 Q5003930
 Q5003940
 Q5003950
 Q5003960
 Q5003970
 Q5003980
 Q5003990
 Q5004000
 Q5004010
 Q5004020
 Q5004030
 Q5004040
 Q5004050
 Q5004060
 Q5004070
 Q5004080
 Q5004090
 Q5004100
 Q5004110
 Q5004120
 Q5004130
 Q5004140
 Q5004150
 Q5004160
 Q5004170
 Q5004180
 Q5004190
 Q5004200
 Q5004210
 Q5004220
 Q5004230
 Q5004240
 Q5004250
 Q5004260
 Q5004270
 Q5004280
 Q5004290
 Q5004300
 Q5004310
 Q5004320

```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
DO UNTIL (EXIT)
  I=I+1
  IM1= I-1
  IF (IM1.EQ.0) IM1=2
  DO (AXIAL LOCATION INDICATORS)
  DO UNTIL (TOP)
    J=J+1
    JM1=J-1
    IF (JM1.EQ.0) JM1=2
    DO (VERTICAL LOCATION INDICATORS)
    DO (BOUNDARY CONDITION SWITCHES)
    DO (GEOMETRY ADVANCE)
    DO (FIELD GRADIENTS)
    DO (VELOCITIES AND DENSITIES)
    DO (SUPERSONIC CORRECTION)
    DO (T.S.O'S AND RESIDUALS)
    DO ( DELPHI COEFFICIENTS AND TRANSPOSED TERMS)
  END
  DO (FORWARD ELIMINATION)
  IF (I.NE.1)
    100 (BACK SUBSTITUTION)
  J=0
END
DO (UPDATE OF FIELD VARIABLES)
DO (CONVERGENCE TESTS)
  I=0
END
DO (SURFACE FLOW CALCULATION FOR FINISHED GRID)
RETURN
ENTRY KEEPER
DO (OUTPUT OF PHI AND CALCULATION OF FINAL VELOCITY FIELD)
RETURN

LOGICAL INLET, BLADE, EXIT, BOTTOM, TOP,
1 RELAXD, SETTLE, BUG1, BUG2, BUG3, BUG22
LOGICAL DOSURF, SMOOTH, FINEST, RESTAR, ALLOUT
REAL LAB, LAD, MINF, LENGTH, JACOB, LAMDAO
REAL MACHA, MACHB, MINF2
REAL MACH (8)
INTEGER FIRSTJ
DOUBLE PRECISION CAPK, CAPKP
COMMON/ PARAM/ RESTAR, NOZIN, BEF1, BETA2, QINF1, CI, CII, GAMMA, MINF, EM
1, DELTA, BETA, TOL, BUG, BUG2, BUG3, IT, GUESS, OVEREL, FIXR, NOWREL, NOTYET
1, DAMP, SUPREL, S, WAKE, B2OB1, UNDERL, RADLAT, RADLE, OMEGA, FLOCO, VAXIAL,
1 ALLOUT, KKMAX
COMMON/ QUASI/ RADIUS (100,30), HEIGHT (100,30)
COMMON/ ENTIRE/ X (100,30), Y (100,30), PHI (100,30), DELPSV (100,30)

```

```

COMMON/GEOM2/ZETA(100),ETA(100)
COMMON/CROSS/ ZMACH(100,30)
COMMON/GEOM/NOZ,M,RLE,RTE,A,CHORD,XUPS,XONS,LAMDAO,CC,CAPK,
1CAPKP,PI
1,RNK,V8,KN,NED,BB,DOSURF,RFAC,SOLID,MPLS
1,STABAC,SLP1,SLP2,SLP3,SLP4,SMOOTH,CHOP,THETT,THETL
1,FINEST
COMMON/CALVEL/FAKEFI(4,6),FAKFI(4,6),RHOCON,QLIM,EXPO,ROGTATN,
1ROTROT,IM1,IEEXIT,PRIORS(14,30),INDEX(100,30),INDEX(32),FAKEU(100)
1,FAKEV(100),ICOUNT,INDEX,AMY,AMZ
COMMON/VELOUT/D(100),E(100),F(100),DELPHI(100),TENOLF(100,30),
1DUMAX(100,30),DUMBY(100,30),BCPR,FADBOP,XOCX,KK
1DIMENSION RHOS(8),FI(8),FI2(3),FI3(3),FI9(8),SPAN(8),
1EMACH(100,30)
2,COSX(8),LENGTH(8),DEDX(8),DZDX(8),DEDY(8),
3DZDY(8),T(8),DPHIDZ(8),DPHIDE(8),U(8),V(8),UN(8),RHO(8),
4RHOUL(8),ONOFF(8),DELPML(3)
5,HYTE(8)
NAMELIST/ABSLUT/MINF,MINF2,QINF2,ABETA1,ABETA2,AQINF1,AQINF2,AMINF
1,AEML,ROTATN
1,NAMELIST/PARAMS/ IEEXIT,NOZ,M,BETA1,BETA2,LAMDAO
1,S,CHORD,RLE,RTE,NED,KN
1,STABAC,SLP1,SLP2,SLP3,SLP4,CHOP,THETT,THETL
1,QINF1,C1,C11,GAMMA,MINF,EM,DELTA,BETA,WAKE,B2OB1,RADRAT
1,OMEGA,FLOCC,VAXIAL, SUPREL,NOWREL,NOTYET,DAMP,
1,TOL,BUG,IT,GUESS,OVEREL,BUG3
1,SMOOTH,FINEST,BUG,BUG2,BUG3
NAMELIST/YTESD/ KK,AIJ,AIJM1,AIJPI,TRANSP,I,J,CIRCO
NAMELIST/CONST/CIRCLN,MINF2,B2OB1,R2OR1,RINF,RATLE
DATA AQINF1/1./,AVGFAC,TEST/1.1./

COMMON/HIBALL/ SUPERG(46,30,100)
XDIF(A,B,C,D)=.5*(A+B-C-D)

C DO (INPUT SPECIFICATION)
C
C PROCEDURF ( INPUT SPECIFICATION)
C
C IF(NOZ-1.NE.NOZIN) GOTO 201
C IF(NOZ-1.EQ.NOZIN) THEN

```

Q5005290
Q5005300
Q5005310
Q5005320
Q5005330
Q5005340
Q5005350
Q5005360
Q5005370
Q5005380
Q5005390
Q5005400
Q5005410
Q5005420
Q5005430
Q5005440
Q5005450
Q5005460
Q5005470
Q5005480
Q5005490
Q5005500
Q5005510
Q5005520
Q5005530
Q5005540
Q5005550
Q5005560
Q5005570
Q5005580
Q5005590
Q5005600
Q5005610
Q5005620
Q5005630
Q5005640
Q5005650
Q5005660
Q5005670
Q5005680
Q5005690
Q5005700
Q5005710
Q5005720
Q5005730
Q5005740
Q5005750
Q5005760

```

RELSAV=OVEREL
UNDSAV=UNDERL
SUPSAV=SUPREL
SHIFTM=ABS(CI)
IF(RADRAT.EQ.1.)IRADLE=1.
IF(RADRAT.EQ.1.)FIXR=1.
IF(SHIFTM.EQ.0.)SHIFTM=1.
IF(FLOCO.EQ.999.)AND.OMEGA.NE.0.)
1FLOCO=VAXIAL/OMEGA/RADLE
IF(FIXR.LT.1.E-06)FIXR=1.
ROTATN=COS(BETA1+LAMDA0)/FLOCO
IF(OMEGA.EQ.0.)AND.FLOCO.EQ.999.)ROTATN=0.
ROTROT=ROTATN*2
BETA1=(BETA1+LAMDA0)
BETA2=(BETA2+LAMDA0)

```

C DO (EXIT MACH NUMBER ITERATION)

C PROCEDURE (EXIT MACH NUMBER ITERATION)

```

FAB=(GAMMA-1.)/2.
ROT FAC=ROTROT*(RADRAT**2-1.)
IF(EM.NE.10.)GOTO205
IF(EM.EQ.10.)THEN
PIOP2=1./(RADRAT-WAKE/S)
SETTLE=.FALSE.
IF(MINF.GT.0.)EM=MINF*.8
IF(MINF.LT.0.)EM=1.0+ABS(MINF)
MINF=ABS(MINF)
FAC1=1./(1.+FAB*MINF*MINF*(1.+ROTFAC))
FAC1=FAC1**((GAMMA+1.)/4./FAB)*COS(BETA1)/COS(BETA2)/B2081
1*PIOP2*MINF+FAB*EM*EM)
FAC2=1./(1.+(GAMMA+1.)/4./FAB)
FAC3=FAC2**((GAMMA+1.)/4./FAB)
EFF=EM*FAC3-FAC1
DFDM=FAC3*(1.-FAC2*EM*EM*(GAMMA+1.)/2.)
DM=-EFF/DFDM
SETTLE=ABS(DM)/EM.LT..001
EM=EM+DM
IF(.NOT.SETTLE)GOTO208
END
MINF2=EM
MINF=ABS(MINF)
QINF2=SQR((1.+ROT FAC+1./FAB/MINF/MINF)/(1.+1./FAB/EM/EM))
AQINF2=QINF2
ABETA2=BETA2*180./PI

```

208

C 205

```

C
ABETA1=BEETA1*180./PI
IF(ROTATN.EQ.0.) GO TO 210
IF(ROTATN.NE.0.) THEN
  SINBET=SIN(BEETA1)+ROTATN
  COSBET=COS(BEETA1)
  ABETA1=ATAN2(SINBET,COSBET)*180./PI
  SINBET=QINF2*SIN(BEETA2)
  COSBET=QINF2*COS(BEETA2)
  ABETA2=ATAN2(SINBET,COSBET)*180./PI
  AQINF1=COS(BEETA1)/COS(PI/180.*ABETA1)
  AQINF2=QINF2*COS(BEETA2)/COS(PI/180.*ABETA2)
  AMINF=MINF*COS(BEETA1)/COS(PI/180.*ABETA1)
  AMINF=EM*COS(BEETA2)/COS(ABETA2*PI/180.)
  WRITE(6,779)
779 FORMAT(1,ROTATION EFFECTS INCLUDED,ABSOLUTE AND RELATIVE VELOCITY,
1,TRIANGLES HAVE -)
  WRITE(6,ABS LUT)
  BETAL=ABETA1*PI/180.-LAMDAO
  BETA2=ABETA2*PI/180.-LAMDAO
END
C
P1OP2=1./RADAT
CIRCO=RADLE/FIXR*S*(AQINF1*SIN(LAMDAO+BETA1)-AQINF2*SIN(LAMDAO+
1,BETA2))/P1OP2
CIRCOO=CIRCO
IF(CIRCO.EQ.0.) CIRCOO=1.
CIRCLN=CIRCO
R2ORI=RADLE
RINF=RADLE
RATLE=FIXR
WRITE(6,780)
780 FORMAT(1,CALCULATED FLOW PARAMETERS)
C
WRITE(6,CONST)
END
C
IF(NOZ-1.NE.NOZIN) SUPSAV=SUPSAV-CII
201 IF(SUPSAV.LT..5) SUPSAV=.5
OVEREL=1.
UNDEREL=1.
SUPREL=1.
NZZ=2*NOZ-2
KKK=0
IEXIT=NZZ
I67=M-14+1
I67=MAX0(I67,I)
2
FORMAT(6E13.5)

```

```

KK=0
M3=M-3
ITIEX=IEXIT-1
WRITE(6,777) M3,ITIEX,SUPSAV
FORMAT(1H1,12X)
1. RELAXATION BEGINS ON GRID OF 'I3,' SURFACE CONTOURS ',
1. INTERSECTED BY 'I3,' RADIATING LINES' /IHO,30X,' SUPERSONIC ',
1. RELAXATION ',
1. PARAMETER, SUPREL= 'E13.5)
WRITE(6,778)
FORMAT(IHO,3X,' ITERATION AVERAGE RESIDUAL ',
1. MAX MACH
1. CALCULATED RELATIVE CIRC SUBSONIC SUPERSONIC '/6X,
1. COUNT PHI CORRECTN AT T. E. ON BLADE CIRCULATION',
1. ERROR RELAX. FAC. RELAX. FAC. '/')
IF(WAKE-EQ.0.) GOTO212
IF(WAKE.NE.0.) THEN
XMAX=0.
DO 101 I1=1,IEXIT
IF(X(I1,M).GT.XMAX) XMAX=X(I1,M)
IF(X(I1,M).GT.XMAX) ITE=I1
CONTINUE
X44=.95*X(ITE,M)
X12=(X(2,4)-X44)*SLP3
M02=M/2
DO 109 I105=1,IEXIT
IF(ABS(X(I105,M02)-X12-X44).LE.1.E-02) GOTO1091
CONTINUE
CONTINUE
S105=RADLE/FIXR*S*RADIUS(I105,M02)/RADLE
BDELT=WAKE/S105*SLP4
DO 119 I=1,IEXIT
DO 118 J=1,M
IF(X(I,J).GT.X44) TERPIT=AMIN1((X(I,J)-X44)/X12,1.)
IF(X(I,J).GT.X44) HEIGHT(I,J)=HEIGHT(I,J)*(1.-TERPIT*BDELT)
CONTINUE
CONTINUE
CONTINUE
END
END
C DO (INITIALIZATION OF FIELD VARIABLES)
C
C PROCEDURE (INITIALIZATION OF FIELD VARIABLES)

```

```

212 CORB=COS(LAMDAO+BETA1)*AQINF1*RADLE
COSB=COS(LAMDAO+BETA1)*AQINF1
S1B8= SIN(LAMDAO+BETA1)*AQINF1*RADLE
S1NB= SIN(LAMDAO+BETA1)*AQINF1
IF(RESTAR.OR.NOZ-1.NE.NOZIN)GOTO219
IEXP1=IEXIT+1
BBLR=BEETA1+BEETA2+LAMDAO+ROTA TN
DO 135 I=1,IEXP1
DO 134 J=1,M
PHI(I,J)=COSB*X(I,J)+SINB*Y(I,J)
IF(ROTA TN.NE.0.)PHI(I,J)=PHI(I,J)*RADLE
RDDDD=RADLE/HEIGHT(I,J)/RADIUS(I,J)
IF(BBLR.EQ.0.)PHI(I,J)=PHI(I,J)*RDDDD
CONTINUE
CONTINUE
NOZM1=NOZ-1
NOZP1=NOZ+1
DO 141 I=NOZM1,NOZP1
DO 140 J=3,5
PHI(I,J)=COSB*X(I,J)+SINB*Y(I,J)
IF(ROTA TN.NE.0.)PHI(I,J)=PHI(I,J)*RADLE
RDDDD=RADLE/HEIGHT(I,J)/RADIUS(I,J)
IF(BBLR.EQ.0.)PHI(I,J)=PHI(I,J)*RDDDD
CONTINUE
CONTINUE
DO 153 J=1,6
DO 152 I=1,4
I10=IEXIT+I-3
IF(J.LE.5.AND.J.GT.2.AND.
1(NOZ-1.NE.NOZIN.OR.RESTAR)
1.AND.I1.NE.4)PHI(I1,J)=(COSB*X(I1,J)+SINB*Y(I1,J))
1
IF(ROTA TN.NE.0.)PHI(I1,J)=PHI(I1,J)*RADLE
RDDDD=RADLE/HEIGHT(I1,J)/RADIUS(I1,J)
IF(BBLR.EQ.0.)PHI(I1,J)=PHI(I1,J)*RDDDD
IF(J.LE.5.AND.J.GT.2.AND.
1(NOZ-1.NE.NOZIN.OR.RESTAR)
1.AND.I10.GT.IEXIT-2)PHI(I10,J)=(COSB*X(I10,J)+SINB*Y(I10,J))
1
IF(ROTA TN.NE.0.)PHI(I10,J)=PHI(I10,J)*RADLE
RDDDD=RADLE/HEIGHT(I10,J)/RADIUS(I10,J)
IF(BBLR.EQ.0.)PHI(I10,J)=PHI(I10,J)*RDDDD
CONTINUE
CONTINUE
DO 161 J=1,M
PHI(I1,J)=
1PHI(I1,IEXIT-1,J)-RADLE/FIXR*S*(AQINF1*SIN(LAMDAO+BETA1)-AQINF2*SIN(
1LAMDAO+BETA2)/PIOP2)

```

```

QSD06730
QSD06740
QSD06750
QSD06760
QSD06770
QSD06780
QSD06790
QSD06800
QSD06810
QSD06820
QSD06830
QSD06840
QSD06850
QSD06860
QSD06870
QSD06880
QSD06890
QSD06900
QSD06910
QSD06920
QSD06930
QSD06940
QSD06950
QSD06960
QSD06970
QSD06980
QSD06990
QSD07000
QSD07010
QSD07020
QSD07030
QSD07040
QSD07050
QSD07060
QSD07070
QSD07080
QSD07090
QSD07100
QSD07110
QSD07120
QSD07130
QSD07140
QSD07150
QSD07160
QSD07170
QSD07180
QSD07190
QSD07200

```

```

161  PHI(IEXIT+1,J)=PHI(3,J)+
      1  RADLE/FIXR*S*(AQINF1*SIN(LAMDAO+BETA1)-AQINF2*SIN(LAMDAO+
      1  BETA2)/PIOP2)
      I=0
      J=0
      RHOCN=(GAMMA-1.)/2.*MINF*MINF
      EXPO= 1./((GAMMA-1.))
      QLIIM=1.+1./RHOCN
      BMZ=1./MINF/MINF
      BMY=(GAMMA-2.)/2.
      AMZ=.5/BMZ
      BMX=8*MY*AMZ/2.
      BMY=-BMX/BMZ
      BIGK=RADLE/FIXR*COS(BETA1)
      UOOUT=AQINF2*COS(ABETA2*PI/180.)
      DPTH0=AQINF2*SIN(ABETA2*PI/180.)
      UOIN=AQINF1*COS(ABETA1*PI/180.)
      DPTH1=AQINF1*SIN(ABETA1*PI/180.)*RADLE/FIXR
      INDEX(N0Z-2,3)=3
      INDEX(N0Z-2,4)=3
      INDEX(N0Z-2,5)=3
      INDEX(N0Z-2,6)=1
      INDEX(N0Z-1,5)=2
      INDEX(N0Z-1,6)=1
      INDEX(N0Z,5)=2
      INDEX(N0Z,6)=1
      INDEX(N0Z+1,4)=2
      INDEX(N0Z+1,5)=4
      INDEX(N0Z+1,6)=1
      INDEX(N0Z+2,3)=1
      INDEX(N0Z+2,4)=1
      INDEX(N0Z+2,5)=1
      INDEX(IEXIT-2,3)=3
      INDEX(IEXIT-2,4)=3
      INDEX(IEXIT-2,5)=1
      INDEX(IEXIT-2,6)=1
      INDEX(IEXIT-1,5)=1
      INDEX(IEXIT,5)=2
      INDEX(IEXIT,6)=1
      IF(B20B1*RADLE.EQ.1.)GOTO 2691
      C  FOR QUASI - 3D CASES, ADJUST CHANNEL AREA AT FARFIELD ELEMENTS WHERE
      C  IT DROPS BELOW SONIC THROAT AREA BASED ON THE INPUT DISTRIBUTIONS.
      DO 162 J1=1,IEXPI
      DO 162 J1=1,M

```

```

QSO07210
QSO07220
QSO07230
QSO07240
QSO07250
QSO07260
QSO07270
QSO07280
QSO07290
QSO07300
QSO07310
QSO07320
QSO07330
QSO07340
QSO07350
QSO07360
QSO07370
QSO07380
QSO07390
QSO07400
QSO07410
QSO07420
QSO07430
QSO07440
QSO07450
QSO07460
QSO07470
QSO07480
QSO07490
QSO07500
QSO07510
QSO07520
QSO07530
QSO07540
QSO07550
QSO07560
QSO07570
QSO07580
QSO07590
QSO07600
QSO07610
QSO07620
QSO07630
QSO07640
QSO07650
QSO07660
QSO07670
QSO07680

```

```

IF (INDEX(11,J1),EQ,0) GOTO 162
ROTFAC=ROTR0T*(RAIUS(11,J1)/RADLEJ**2-1.)/+1.
IF (11.LE.NOZ-3;OR:11.GE.NOZ+3) GOTO 1622
ASTAR=MINF#COS(BEETA1)/(ROTFAC#MINF#MINF#FAB+1.)*((GAMMA+1.)/4.
11/FAB)*RADLE#((1.+MINF#MINF#SIN(BEETA1)**2)*FAB+1.)*((GAMMA+1.)/4.
11/FAB)
WMACH=MINF
GOTO 1623
1622 ASTAR=EM#COS(BEETA2)/(ROTFAC#EM#EM#FAB+1.)*((GAMMA+1.)/4.
11/FAB)*RADLE#((1.+EM#EM#SIN(BEETA2)**2)*FAB+1.)*((GAMMA+1.)/4.
WMACH=EM
1623 IF (ASTAR.LE.RADIUS(11,J1)*HEIGHT(11,J1)) GOTO 162
WRITE(6,1666) 11,J1,X1,Y1,WMACH,ASTAR
1666 FORMATT:OQUASI-3D CHANNEL CROSS SECTION AT BOUNDARY ELEMENT I = ,13
1, J = ,13, X = ,E13.5, FALLS BELOW SONIC THROAT AREA./
1, BASED ON MACH = ,E13.5, AT FAR FIELD STATION,
1, PROCEEDING WITH AREA = 1.01 * THROAT = 1.01 * .E12.5)
RRR=ASTAR/HEIGHT(11,J1)
IF (RADLE.NE.1..AND..M2001.NE.1..RADIUS(11,J1)=(RRR#RADIUS(11,J1))/2
C IF (RADLE.NE.1..AND..M2001.NE.1..HEIGHT(11,J1)=ASTAR*.01/RADIUS(11,J1)
C IF (B2001.NE.1..AND..ADLE.EQ.1..HEIGHT(11,J1)=1..1#ASTAR
C IF (B2001.EQ.1..AND..ADLE.NE.1..RADIUS(11,J1)=A..AR*.01
HEIGHT(11,J1)=ASTAR*.01/RADIUS(11,J1)
CONTINUE
162 END
C RELAXD= .FALSE.
2691 KK=KK+1
269 INDEX=1
ICOUNT=0
KKK=KKK+1
IF (KK.GT.NOZREL) OVEREL=RELSAV
IF (KK.GT.NOZVET) SUPREL=SUPSAV
IF (NOZ-1.NE.NOZIN)
1 UNDERL=UNDSAV#FLOAT(KKK)/FLOAT(NOZVET)*(OVEREL-UNDSAV)
I=I+1
IM1=I-1
IF (IM1.EQ.0) IM1=2
C DO (AXIAL LOCATION INDICATORS)
C PROCEDURE (AXIAL LOCATION INDICATORS)
INLET=.FALSE.
EXIT = .FALSE.

```

QSD08170
QSD08180
QSD08190
QSD08200
QSD08210
QSD08220
QSD08230
QSD08240
QSD08250
QSD08260
QSD08270
QSD08280
QSD08290
QSD08300
QSD08310
QSD08320
QSD08330
QSD08340
QSD08350
QSD08360
QSD08370
QSD08380
QSD08390
QSD08400
QSD08410
QSD08420
QSD08430
QSD08440
QSD08450
QSD08460
QSD08470
QSD08480
QSD08490
QSD08500
QSD08510
QSD08520
QSD08530
QSD08540
QSD08550
QSD08560
QSD08570
QSD08580
QSD08590
QSD08600
QSD08610
QSD08620
QSD08630
QSD08640

```
IF(I.EQ.1) INLET =.TRUE.
IF( I.EQ. IEXIT) EXIT=.TRUE.
END
```

C 275

```
J=J+1
JM1=J-1
IF(JM1.EQ.0) JM1=2
```

C DO (VERTICAL LOCATION INDICATORS)

C PROCEDURE (VERTICAL LOCATION INDICATORS)

```
BOTTOM = .FALSE.
TOP = .FALSE.
BLADE = .FALSE.
IF(J.EQ.1) BOTTOM =.TRUE.
IF(J.EQ.M) TOP =.TRUE.
BLADE = TOP
END
```

C

C DO (BOUNDARY CONDITION SWITCHES)

C PROCEDURE (BOUNDARY CONDITION SWITCHES)

```
DO 209 I1=1,8
ONOFF(I1)=1.
IF(INLET) THEN
ONOFF(3)= 0.
ONOFF(4)= 0.
ONOFF(5)= 0.
ONOFF(6)= 0.
END
```

209

```
IF(BOTTOM) THEN
ONOFF(5)= 0.
ONOFF(6)= 0.
ONOFF(7)= 0.
ONOFF(8)= 0.
END
IF(TOP) THEN
ONOFF(1)= 0.
ONOFF(2)= 0.
ONOFF(3)= 0.
ONOFF(4)= 0.
END
```

```
IF(.NOT.INLET.AND..NOT.BOTTOM.AND..NOT.TOP ) GOT0235
```

```

2099      IF(TOP)IS=1
C          IF(INLET)IS=3
          IF(BOTTOM)IS=5
          IE=IS+3
          ON 2099 I1=IS,IE
          ONOFF(I1)=0.
          END

2099      DO (GEOMETRY ADVANCE)
C
C          PROCEDURE (GEOMETRY ADVANCE)
235      HITE(1)=XDIF(HEIGHT(I,J),HEIGHT(I+1,J),0.,0.)
          IF(I.EQ.IEXIT)HITE(1)=HEIGHT(I,J)
          HITE(2)=XDIF(HEIGHT(I,J),HEIGHT(I+1,J),0.,0.)
          HITE(3)=HITE(2)
          HITE(8)=HITE(1)
          SPAN(1)=XDIF(RADIUS(I,J),RADIUS(I+1,J),0.,0.)
          IF(I.EQ.IEXIT)SPAN(1)=RADIUS(I,J)
          SPAN(1)=SPAN(1)/FIXR
          SPAN(2)=XDIF(RADIUS(I,J),RADIUS(I+1,J),0.,0.)
          SPAN(2)=SPAN(2)/FIXR
          SPAN(3)=SPAN(2)
          SPAN(8)=SPAN(1)
          T(6) = -T(3)*ONOFF(6)
          T(7) = -T(2)*ONOFF(7)

C          CALCULATE THE METRICS ON THE FIRST ITERATION
C          STORE THEM FOR SUBSEQUENT ITERATIONS

          IF(KK.GT.2)GOTO237

C          CALCULATE VELOCITIES AT UP AND DOWNSTREAM BOUNDARY ELEMENTS TO SAT
C          MASS FLOW
C          INDEX(I,J)>0 POINTS TO BOUNDARY ELEMENTS AFFECTED

          IF(INDEX(I,J).EQ.0.OR.KK.GT.1)GOTO2037
          ICOUNT=ICOUNT+1
          IIS=INDEX(ICOUNT)
          IDIF=INDEX(I,J)+IIS-1
          DO 9999 I11=IIS,IDIF
              I1=I11
              IF(I11.EQ.1.AND.I1.EQ.IDIF)I1=8
              ROTATX=ROTATN*SPAN(I1)*FIXR/RADLE
              BIGKH=8IGK/HITE(I1)/SPAN(I1)

```

QSO09130
QSO09140
QSO09150
QSO09160
QSO09170
QSO09180
QSO09190
QSO09200
QSO09210
QSO09220
QSO09230
QSO09240
QSO09250
QSO09260
QSO09270
QSO09280
QSO09290
QSO09300
QSO09310
QSO09320
QSO09330
QSO09340
QSO09350
QSO09360
QSO09370
QSO09380
QSO09390
QSO09400
QSO09410
QSO09420
QSO09430
QSO09440
QSO09450
QSO09460
QSO09470
QSO09480
QSO09490
QSO09500
QSO09510
QSO09520
QSO09530
QSO09540
QSO09550
QSO09560
QSO09570
QSO09580
QSO09590
QSO09600

```

UO=UOOUT
DPHDT H=DPHTHO
IF(IICOUNT.GE.11.AND.IICOUNT.LE.24)UO=UOIN
IF(IICOUNT.GE.11.AND.IICOUNT.LE.24)DPHDT H=DPHTHI
FAKEV(IIDEX)=DPHDT H/SPAN(11)
VREL2=(FAKEV(IIDEX)-ROTATX)*#2
ROTATX=ROTATX*ROTATX-ROTATN*ROTATN
SETTLE=.FALSE.
YEW=UO
IF(NOTYET.GT.KKMAX)YEW=BIGKH
JCOUNT=0
JSQAR=YEW*VREL2
IF(JSQAR.GE.QLIM)JSQAR=.99*QLIM
RHOFAC=RHOCON*(1-JSQAR)+1.
RHOFAC=RHOCON*ROTATX*RHOCON
IF(RHOFAC.LT.0.OR.NOTYET.GT.KKMAX)RHOFAC=1.
EFF=YEW-BIGKH*RHOCON*(-EXPO)
DFDU=1-MINF*MINF*YEW*BIGKH*RHOCON*(-EXPO)
DU=-EFF/DFDU
SETTLE=ABS(DU)/YEW.LT..001.OR.JCOUNT.GT.20
YEW=YEW+DU
JCOUNT=JCOUNT+1
GO TO 1999
IF(.NOT.SETTLE)
IF(JCOUNT.GT.20)YEW=BIGKH
IF(JCOUNT.GT.20)WRITE(6,1677) I,J,11
FORMAT('OCALCULATION OF VELOCITY AT BOUNDARY ELEMENT I=',13,
1,J=' ',11=' ',13=' ',EXCEEDS 20 ITERATIONS. PROCEEDING WITH',
1,'INCOMPRESSIBLE ASSUMPTION (VELOCITY.EQ.MASS FLOW/AREA)',)
FAKEV(IIDEX)=YEW
IDEX=IDEX+1
ICOUNT=ICOUNT-1
IDEX=IDEX-INDEX(I,J)

C 2037
C
IF(KK.LE.2) THEN
IF(TOP) GO TO 239
IF(.NOT.TOP) THEN
XABMXB=.5*XDIF(X(I+1,J+1),X(I,J),X(I+1,J+1))*5
YABMYA=.5*XDIF(Y(I+1,J+1),Y(I,J),Y(I+1,J+1))
XABMYB=.5*XDIF(X(I+1,J+1),X(I,J),Y(I+1,J+1))
YABCMYB=.5*XDIF(Y(I+1,J+1),Y(I,J),X(I+1,J+1))
LAB=SQRT(XABMXA*XABMYA+SPAN(1)*YABMYA)
HAB=SQRT(XABMXB*XABMYB+SPAN(2)*YABMYB)
HBC=SQRT(XBCMXB*YABMYA+SPAN(3)*YBCMYB)
COSX(1)=YABMYA*SPAN(1)+YABMYB*SPAN(2)+YBCMYB*SPAN(3)/LAB

```

```

C239
C
COSX(2)=-YABMYB*SPAN(2)/HAB
COSX(3)=-YBCHYB*SPAN(3)/HBC
COSY(1)=-XABMXA/LAB
COSY(2)=-XABMXB/HAB
COSY(3)=-XBCMXB/HBC
END
IF(BOTTOM)GOTO241
IF(.NOT.BOTTOM) THEN
  XADMXA =.5*XDIF(X(I+1,JM1),X(I,JM1),X(I,J),X(I+1,J))
  YADMYA =.5*XDIF(Y(I+1,JM1),Y(I,JM1),Y(I,J),Y(I+1,J))
  LAD = SQR1(XADMXA**2+YADMYA**2)*SPAN(8)*YADMYA
  COSX(8)=-YADMYA*SPAN(8)/LAD
  COSY(8)=-XADMXA/LAD
END
LENGTH(1) = LAB
LENGTH(2) = HAB
LENGTH(3) = HBC
LENGTH(8) = LAD
EXDIF1=X(I+1,J)-X(I,J)
EXDIF2=X(IM1,J)-X(I+1,J)-Y(I,J)
YDIF1=SPAN(1)*Y(I+1,J)-Y(I,J)
YDIF2=Y(IM1,J)-Y(I,J)
SLEN=EXDIF1*EXDIF1+YDIF1*YDIF1
ZETDF1 = ZETA(J+1) - ZETA(J)
ZETDF2 = ZETA(JM1) - ZETA(J)
ETADF1 = ETA(I+1) - ETA(I)
ETADF2 = ETA(IM1) - ETA(I)
CELL = POINT A
EK4 = 2.*YABMYA/ZETDF1
EK3 = (Y(I+1,J)-Y(I,J))/ETADF1
EK2 = 2.*XABMXA/ZETDF1
EK1 = (X(I+1,J)-X(I,J))/ETADF1
L = 1
DO (TRANSFORMATION MATRIX)
  PROCEDURE (TRANSFORMATION MATRIX)
  JACOB=EK1*EK4-EK2*EK3
  IF(JACOB.EQ.0.) GOTO243
  IF(JACOB.NE.0.) THEN
    DEX(L)=EK4/JACOB
    DEX(L)=-EK2/JACOB/SPAN(1)
    DZDX(L)=-EK3/JACOB
    DZDY(L)= EK1/JACOB/SPAN(1)
  END
END
EK4 = 2.*YADMYA/ZETDF2
CELL = IV, POINT A
EK2 = 2.*XADMXA/ZETDF2
L=8

```

```

Q5009610
Q5009620
Q5009630
Q5009640
Q5009650
Q5009660
Q5009670
Q5009680
Q5009690
Q5009700
Q5009710
Q5009720
Q5009730
Q5009740
Q5009750
Q5009760
Q5009770
Q5009780
Q5009790
Q5009800
Q5009810
Q5009820
Q5009830
Q5009840
Q5009850
Q5009860
Q5009870
Q5009880
Q5009890
Q5009900
Q5009910
Q5009920
Q5009930
Q5009940
Q5009950
Q5009960
Q5009970
Q5009980
Q5009990
Q5010000
Q5010010
Q5010020
Q5010030
Q5010040
Q5010050
Q5010060
Q5010070
Q5010080

```

QSO10090
 QSO10100
 QSO10110
 QSO10120
 QSO10130
 QSO10140
 QSO10150
 QSO10160
 QSO10170
 QSO10180
 QSO10190
 QSO10200
 QSO10210
 QSO10220
 QSO10230
 QSO10240
 QSO10250
 QSO10260
 QSO10270
 QSO10280
 QSO10290
 QSO10300
 QSO10310
 QSO10320
 QSO10330
 QSO10340
 QSO10350
 QSO10360
 QSO10370
 QSO10380
 QSO10390
 QSO10400
 QSO10410
 QSO10420
 QSO10430
 QSO10440
 QSO10450
 QSO10460
 QSO10470
 QSO10480
 QSO10490
 QSO10500
 QSO10510
 QSO10520
 QSO10530
 QSO10540
 QSO10550
 QSO10560

```

C      DO (TRANSFORMATION MATRIX)
C      JACOB=EK1*EK4-EK2*EK3
C      IF (JACOB.EQ.0.) GOTO245
C      IF (JACOB.NE.0.) THEN
C      DEDX(L)= EK4/JACOB
C      DEDY(L)=-EK2/JACOB/SPAN(8)
C      DZDX(L)=-EK3/JACOB
C      DZDY(L)= EK1/JACOB/SPAN(8)
C      END
C      CELL I1, POINT B
C      EK4 = (Y(I,J+1)-Y(I,J))/ZETDF1
C      EK3 = 2.*Y8CMY8/ETADF2
C      EK2 = (X(I,J+1)-X(I,J))/ZETDF1
C      EK1 = 2.*X8CMX8/ETADF2
C      L=3
C      DO (TRANSFORMATION MATRIX)
C      JACOB=EK1*EK4-EK2*EK3
C      IF (JACOB.EQ.0.) GOTO247
C      IF (JACOB.NE.0.) THEN
C      DEDX(L)= EK4/JACOB
C      DEDY(L)=-EK2/JACOB/SPAN(3)
C      DZDX(L)=-EK3/JACOB
C      DZDY(L)= EK1/JACOB/SPAN(3)
C      END
C      CELL I1, POINT B
C      EK1 = 2.*X8BMYB/ETADF1
C      EK3 = 2.*Y8BMYB/ETADF1
C      L=2
C      DO (TRANSFORMATION MATRIX)
C      JACOB=EK1*EK4-EK2*EK3
C      IF (JACOB.EQ.0.) GOTO249
C      IF (JACOB.NE.0.) THEN
C      DEDX(L)= EK4/JACOB
C      DEDY(L)=-EK2/JACOB/SPAN(2)
C      DZDX(L)=-EK3/JACOB
C      DZDY(L)= EK1/JACOB/SPAN(2)
C      END
C      T(1)=LENGTH(1)*HITE(1)*(DEDX(1)*COSX(1)+DEDY(1)*COSY(1)
C      1)*#ONOFF(1)
C      T(2)=LENGTH(2)*HITE(2)*(DZDX(2)*COSX(2)+DZDY(2)*COSY(2)
C      1)*#ONOFF(2)
C      T(3)=LENGTH(3)*HITE(3)*(DZDX(3)*COSX(3)+DZDY(3)*COSY(3)
C      1)*#ONOFF(3)
C      T(4)=LENGTH(4)*HITE(4)*(DEDX(4)*COSX(4)+DEDY(4)*COSY(4)
C      1)*#ONOFF(4)
C      T(5)=LENGTH(5)*HITE(5)*(DEDX(5)*COSX(5)+DEDY(5)*COSY(5)
C      1)*#ONOFF(5)
C      T(6)=LENGTH(6)*HITE(6)*(DEDX(6)*COSX(6)+DEDY(6)*COSY(6)
C      1)*#ONOFF(6)
C      T(7)=LENGTH(7)*HITE(7)*(DEDX(7)*COSX(7)+DEDY(7)*COSY(7)
C      1)*#ONOFF(7)
C      T(8)=LENGTH(8)*HITE(8)*(DEDX(8)*COSX(8)+DEDY(8)*COSY(8)
C      1)*#ONOFF(8)
C      STORE METRICS AND GEOMETRY AT POINT I,J
  
```


OD-A120 456

PRELIMINARY MEASUREMENTS AND CODE CALCULATIONS OF FLOW
THROUGH A CASCADE OF DCA BLADING AT A SOLIDITY OF 167
(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA W D MOLLOY

3/3

UNCLASSIFIED

JUN 82

F/G 21/5

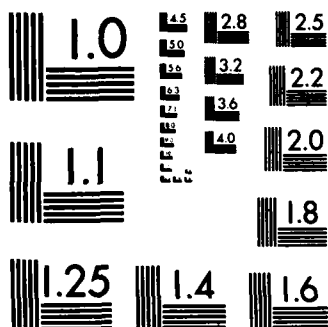
NL

END

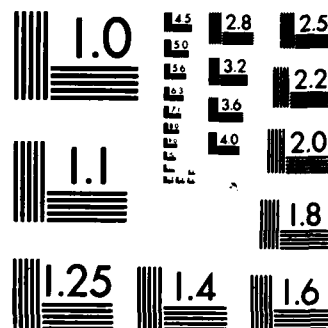
FILMED

202

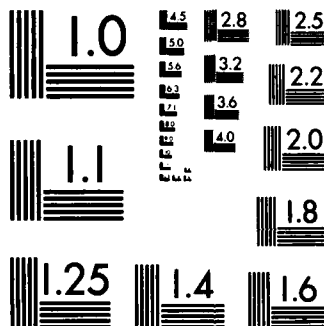
DTIC



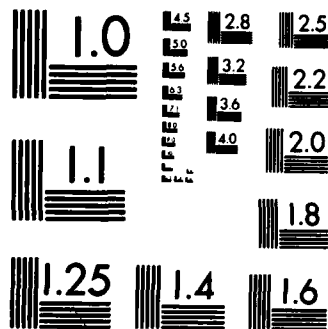
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



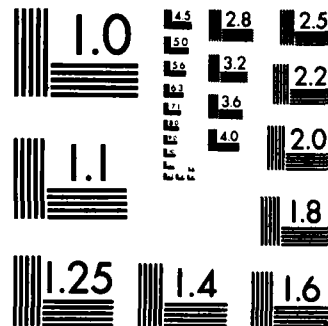
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Q5011530
Q5011540
Q5011550
Q5011560
Q5011570
Q5011580
Q5011590
Q5011600
Q5011610
Q5011620
Q5011630
Q5011640
Q5011650
Q5011660
Q5011670
Q5011680
Q5011690
Q5011700
Q5011710
Q5011720
Q5011730
Q5011740
Q5011750
Q5011760
Q5011770
Q5011780
Q5011790
Q5011800
Q5011810
Q5011820
Q5011830
Q5011840
Q5011850
Q5011860
Q5011870
Q5011880
Q5011890
Q5011900
Q5011910
Q5011920
Q5011930
Q5011940
Q5011950
Q5011960
Q5011970
Q5011980
Q5011990
Q5012000

```

C      PRIORS(2,J) = - T(1)
      END

C      DO (FIELD GRADIENTS)

C      PROCEDURE ( FIELD GRADIENTS)
DO 438 I1=1,3
  FI1(I1)=PHI(I1+1,J)
  FI2(I1)=PHI(I1+1,J+1)
  FI3(I1)=PHI(I1,J+1)
  FI9(I1)=PHI(IM1,J+1)
  FI4=PHI(IM1,J+1)
  FI5=PHI(IM1,J+1)
  FI6=PHI(IM1,J+1)
  FI7=PHI(IM1,J+1)
  FI8=PHI(I1+1,J+1)
  FI9(8)=PHI(I1+1,J)
  I10=I1-1
  IEXIT=3
438

C      CELL I, POINT A
DPHIDZ(1)=XOIF(FI3(1),FI2(1),FI9(1),FI1(1))/ZETDF1
DPHIDE(1)=(FI1(1)-FI9(1))/ETADF1
C      CELL I, POINT B
DPHIDZ(2)=XOIF(FI1(2),FI2(2),FI9(2),FI3(2))/ETADF1
DPHIDE(2)=(FI3(2)-FI9(2))/ZETDF1
C      CELL I, POINT B
DPHIDZ(3)=XOIF(FI5,FI4,FI9(3),FI3(3))/ETADF2
DPHIDE(3)=(FI3(3)-FI9(3))/ZETDF1
C      CELL I, POINT A
DPHIDZ(8)=(FI1(8)-FI9(8))/ETADF1
DPHIDE(8)=XOIF(FI7,FI8,FI9(8),FI1(8))/ZETDF2
C      END

C      DO (VELOCITIES AND DENSITIES)

C      PROCEDURE ( VELOCITIES AND DENSITIES)

      U(6) = U(3)
      V(6) = V(3)
      U(7) = U(2)

```

```

C
V(7) = V(2)
DO 541 I1=1,8
  IF(I1.GE.4.AND.I1.NE.8) GOTO 541
  IF(I1.LT.4.OR.I1.EQ.8) THEN
    U(I1)=DPHIDE(I1)*DEDX(I1)+DPHIDZ(I1)*DZDX(I1)
    V(I1)=DPHIDE(I1)*DEDY(I1)+DPHIDZ(I1)*DZDY(I1)
    IF(INDEX(I1).EQ.0)GOTO5411
    ICOUNT=ICOUNT+1
    ILS=INDEX(ICOUNT)
    IF(I1.LE.ILS)IIP=ILS
    IF(I1.GT.ILS)IIP=ILS-1
    IF(I1.S.EQ.1.AND.IIP=8)EQ.IDIF)IIP=8
    IF(I1.NE.IIP)GOTO5410
    IF(I1P.GT.IDIF.AND.IIP.NE.8)GOTO5410
    V(I1)=FAKEU(IDEX)
    U(I1)=IDEX+1
    IIP=IIP+1
  IF(I1.NE.8) ICOUNT=ICOUNT-1
  IF(I1)=V(I1)-SPAN(I1)*FIXR/RADLE*ROTATN
  UN(I1)=U(I1)*COSX(I1)+V(I1)*COSY(I1)
END
CONTINUE
GOTO 2003
IF(.NOT.TOP) GOTO 2003
U8=SQRT(U(8)*U(8)+V(8)*V(8))*ABS(EXDIF1)/SQRT(SLEN)
IF(KK.GT.7.OR.ABS(FLOCO).GT..01)U(8)=SIGN(U8,U(8))
V(8)=U(8)*YDIF1/EXDIF1
UN(8)=U(8)*COSX(8)+V(8)*COSY(8)
END
U(4) = PRIORS(3,J)
U(5) = PRIORS(4,J)
V(4) = PRIORS(5,J)
V(5) = PRIORS(6,J)
PRIORS(3,J)=U(1)
PRIORS(4,J)=U(8)
PRIORS(5,J)=V(1)
PRIORS(6,J)=V(8)
RHO(7)=RHO(2)
RHO(6)=RHO(3)
RHO(4)=PRIORS(7,J)
RHO(5)=PRIORS(8,J)
ONSUM=0
DO 595 I1=1,8
  IF(KK.LE.2.OR.MOD(KK,2).EQ.0) THEN
    IF(I1.GE.4.AND.I1.NE.8) GOTO 2005
    IF(I1.LT.4.OR.I1.EQ.8) THEN
      RHOFAC=1.

```

```

05012010
05012020
05012030
05012040
05012050
05012060
05012070
05012080
05012090
05012100
05012110
05012120
05012130
05012140
05012150
05012160
05012170
05012180
05012190
05012200
05012210
05012220
05012230
05012240
05012250
05012260
05012270
05012280
05012290
05012300
05012310
05012320
05012330
05012340
05012350
05012360
05012370
05012380
05012390
05012400
05012410
05012420
05012430
05012440
05012450
05012460
05012470
05012480

```

```

RHO(I1)=1.,OR.I.EQ.IEXIT).AND.J.LE.4.AND.ROTATN.NE.O.
IF((I.EQ.2.OR.I.EQ.IEXIT).GOTO 2005
1.OR.KK.LE.NOTYET).GOTO 2005
QSQR=U(I1)*U(I1)+(V(I1)*V(I1))
IF(QSQR.GE.QLIM)QSQR=.99*QLIM
RHOFAC=RHOCON*(1.-QSQR)*ONOFF(I1)+1.
RHO(I1)=1.-AMZ*(QSQR-1.)+AMV*(QSQR-1.)*(QSQR-1.)
IF(QSQR.LT..65.OR.QSQR.GT.1.44)RHO(I1)=RHOFAC**EXPO
C CORRECT DENSITY FOR ROTATION
ROTATX=SPAN(I1)*FLXR/RADLE
IF(ROTRTN.NE.O..AND.J.GT.3)
1RHO(I1)=(RHO(I1)**(1./EXPO))+RHOCON*ROTRTN*((ROTATX)
1**2-1.)*ONOFF(I1)**EXPO
2005 QSQR=U(I1)*U(I1)+V(I1)*V(I1)
IF(I.GT.1)
1MACH(I1)=MINF*SQRT(QSQR)/RHO(I1)**(1./EXPO)
1MACH(I1)=MINF*SQRT(QSQR/RHO(I1)**(1./EXPO))
IF(MACH(I1).GT.2.)MACH(I1)=2.
CONTINUE
PRIORS(7,J)=RHO(I1)
PRIORS(8,J)=RHO(8)
C
595 DO (SUPERSONIC CORRECTION)
C
C PROCEDURE ( SUPERSONIC CORRECTION)
ONSUM=0.
RHSUM=RHO(4)+RHO(5)+RHO(6)+RHO(7)
EMACH(I,J)=0.
XVELIJ=0.
DO 608 I3=1,8
RHOSS(I3)=0.
ONSUM=ONSUM+ONOFF(I3)
YVELIJ=0.
SMALLA=0.
SMALLB=0.
ZETDF1=ZETA(J+1)-ZETA(J)
IF(J.EQ.M)ZETDF1=ZETA(J)-ZETA(J-1)
EMACH(I,J)=(ONOFF(I1)*MACH(1)+ONOFF(8))/ (ONOFF(1)+ONOFF(8)
1)
EMACH(IEXIT+1,J)=EMACH(3,J)
EMACH(1,J)=EMACH(IEXIT-1,J)

```

```

C      IF(KK.LE.2.OR.MOD(KK,IT).EQ.0.OR.MOD(KK,IT).GE.IT/2) THEN
C      STORE DENSITIES FROM SURFACE OF ELEMENT IN DUMMY ARRAYS
      DUMBX(1,J)=(ONOFF(1)*RHO(1)+ONOFF(8)*RHO(8))/(ONOFF(1)+ONOFF(8))
      IF(.NOT.TOP) DUMBY(1,J)=(ONOFF(2)*RHO(2)+ONOFF(3)*RHO(3))/
      (ONOFF(2)+ONOFF(3))
      IF(.NOT.TOP) ZMACH(1,J)=(ONOFF(2)*MACH(2)+ONOFF(3)*MACH(3))/
      (ONOFF(2)+ONOFF(3))
      IF(TOP) DUMBY(1,J)=2.*DUMBY(1,J-1)-DUMBY(1,J-2)
      IF(TOP) ZMACH(1,J)=2.*ZMACH(1,J-1)-ZMACH(1,J-2)
      DUMBX(1,EXIT+1,J)=DUMBX(3,J)
      ZMACH(1,J)=ZMACH(1,EXIT-1,J)
      DUMBY(1,M+1)=2.*DUMBY(1,1)-DUMBY(1,2)
      ZMACH(1,M+1)=2.*ZMACH(1,1)-ZMACH(1,2)

C      CALCULATE DENSITY CORRECTION TERMS, MU*DRHODS*DS
      IF(J.GT.5.OR.I.LT.NOZ-1.OR.I.GT.NOZ+1) THEN
      IF(J.LE.5.AND.I.GE.NOZ-1.AND.I.LE.NOZ+1) GOTO20001
      DO 694 I1=2,8,6
      IF(TOP.AND.I1.NE.8) GOTO 694
      IF(.NOT.TOP.OR.I1.EQ.8) THEN
      I1P1=I1+1
      IF(I1.EQ.8) I1P1=1
      ONSUM=ONOFF(I1)+ONOFF(I1P1)
      XVELIJ=U(I1)*ONOFF(I1)/ONSUM
      YVELIJ=XVELIJ+U(I1P1)*ONOFF(I1P1)/ONSUM
      YVELIJ=V(I1)*ONOFF(I1)/ONSUM
      YVELIJ=YVELIJ+V(I1P1)*ONOFF(I1P1)/ONSUM
      SMALLA=(U(I1)*DZDX(I1)+V(I1)*DZDY(I1))/ONSUM
      SMALLA=SMALLA+(U(I1P1)*DZDX(I1P1)+V(I1P1)*DZDY(I1P1))/ONSUM
      SMALLB=(U(I1)*DEDX(I1)+V(I1)*DEDY(I1))/ONSUM
      SMALLB=SMALLB+(U(I1P1)*DEDX(I1P1)+V(I1P1)*DEDY(I1P1))/ONSUM
      IF(KK.LE.NOTIYET) GOTO694
      IF(KK.GT.NOTIYET) THEN
      IF(BLADE) SMALLA=0.
      SIGNOA=SIGN(1.,SMALLA)
      IF(SMALLA.EQ.0.) SIGNOA=1.
      FLOWMAT=AMAX1(0.,SIGNOA)
      FLOWNOB=SIGN(1.,SMALLB)
      FLOWBY=AMAX1(0.,SIGNOB)
      QUE=SQRT(XVELIJ**2+YVELIJ**2)
      SLP2=TEGARD
      IF(ABS(SMALLB)/QUE.GT.SLP2)QUE=SQRT(SMALLA**2+SMALLB**2)

```

QS012970
 QS012980
 QS012990
 QS013000
 QS013010
 QS013020
 QS013030
 QS013040
 QS013050
 QS013060
 QS013070
 QS013080
 QS013090
 QS013100
 QS013110
 QS013120
 QS013130
 QS013140
 QS013150
 QS013160
 QS013170
 QS013180
 QS013190
 QS013200
 QS013210
 QS013220
 QS013230
 QS013240
 QS013250
 QS013260
 QS013270
 QS013280
 QS013290
 QS013300
 QS013310
 QS013320
 QS013330
 QS013340
 QS013350
 QS013360
 QS013370
 QS013380
 QS013390
 QS013400
 QS013410
 QS013420
 QS013430
 QS013440

013450
 013460
 013470
 013480
 013490
 013500
 013510
 013520
 013530
 013540
 013550
 013560
 013570
 013580
 013590
 013600
 013610
 013620
 013630
 013640
 013650
 013660
 013670
 013680
 013690
 013700
 013710
 013720
 013730
 013740
 013750
 013760
 013770
 013780
 013790
 013800
 013810
 013820
 013830
 013840
 013850
 013860
 013870
 013880
 013890
 013900
 013910
 013920

```

    MYJ=J-IFIX(SIGNOA)
    MYI=I-IFIX(SIGNOB)
    JYM=MYJ+IFIX(FLWAT)
    IYM=MYI+IFIX(FLWB Y)
    IF(MYJ.EQ.0)MYJ=2
    IF(MYI.EQ.0)MYI=IEXIT-2
    IYM=IYM-1
    IF(IYM.EQ.0)IYM=1
    JYM=JYM-1
    IF(JYM.EQ.0)JYM=1
    IF(I1.EQ.2)GOTO2019
    IF(I1.EQ.8)THEN
      MACHA=(EMACH(I,J))+EMACH(I,MYJ))/2.
      MACHB=(EMACH(I,J))+EMACH(I,MYJ))/2.
      SWICHA=-AMAX1(0.,1./SHIFTM#2-1./MACHA/MACHB)/QUE
      SWICHB=-AMAX1(0.,1./SHIFTM#2-1./MACHB/MACHB)/QUE
    C
  C
    RHOS(2)=MU*DRHOS#DS AT TOP OF ELEMENT
    RHOS(8)=MU*DRHOS#DS ON FORWARD SWEEPING EDGE OF ELEMENT
    RHOS(11)=SWICHB#SIGNOB#SMALLB*(DUMBX(I,J)-DUMBX(MYI,J))+
    1 SWICHA#SIGNOA#SMALLA*(DUMBX(I,J)-DUMBX(I,MYJ))
    C
    C
    C
    GOTO 2020
    MACHA=(ZMACH(I,J))+ZMACH(I,MYJ))/2.
    MACHB=(ZMACH(I,J))+ZMACH(I,MYJ))/2.
    SWICHA=-AMAX1(0.,1./SHIFTM#2-1./MACHA/MACHB)/QUE
    SWICHB=-AMAX1(0.,1./SHIFTM#2-1./MACHB/MACHB)/QUE
    RHOS(11)=SWICHB#SIGNOB#SMALLB*(DUMBX(I,J)-DUMBX(MYI,J))+
    1 SWICHA#SIGNOA#SMALLA*(DUMBX(I,J)-DUMBX(I,MYJ))
    C
    C
    END
    RHOS(11P1)=RHOS(11)
    RHN(11)=RHOS(11)+DAMP#RHOS(11)*ONOFF(11)
    IF(RHOS(11).LE.0.)RHOS(11)=RHOS(11)-2.*
    1DAMP#RHOS(11)*ONOFF(11)
    PHO(11P1)=RHOS(11P1)+DAMP#RHOS(11P1)*ONOFF(11P1)
    IF(RHOS(11P1).LE.0.)RHOS(11P1)=RHOS(11P1)-2.*
    1DAMP#RHOS(11P1)*ONOFF(11P1)
    PRIORS(7,J)=RHOS(11)
    PRIORS(8,J)=RHOS(11)
    FORMAT(2I8,7E13.5)
    3334
    C
    C
    END
    END
    CONTINUE
    694
    C
    CONTINUE
    20001
    C
    CONTINUE
    C
    END
  
```

```

C      DO ( T'S,Q'S AND RESIDUALS)
C      PROCEDURE ( T'S,Q'S AND RESIDUALS)
C      NET MASS FLOW TERM
C      ONSUM=0.
C      FLONET = -RHOUL(2)*ONOFF(7)-RHOUL(3)*ONOFF(6)+PRIORS(9,J)*ONOFF(4
1) + PRIORS(10,J)*ONOFF(5)
C      DO 711 I1=1,8
C      ONSUM=ONSUM+ONOFF(I1)
C      IF(I1.GE.4.AND.I1.NE.8) GOTO 711
C      IF(I1.LT.4.OR.I1.EQ.8) THEN
C      ROSUM=ROSUM+RHO(I1)
C      RHOUL(I1)=RHO(I1)*UN(I1)*LENGTH(I1)* HITE(I1)*ONOFF(I1)
C      FLONET =FLONET + RHOUL(I1)
C      END
C      CONTINUE
C      ROAVG=ROSUM/ONSUM
C      TENOLF(I,J)=FLONET
C      PRIORS(9,J) = -RHOUL(I)
C      PRIORS(10,J) = -RHOUL(8)
C      RESIDUAL TOTAL
C      R=-FLONET
C      END
C      IF(BUG) DO (FIRST DEBUG LIST)
C      PROCEDURE (FIRST DEBUG LIST)
C      BUG AND BUG3 ARE NOT INCLUDED IN NAMELIST PARAMS. THEY MUST BE
C      ADDED OR SET HERE MANUALLY IN ORDER TO ACTIVATE THESE PRINTOUTS.
C      IF(.NOT.EXIT.OR.MOD(KK,I1).NE.0.OR..NOT.TOP.OR..NOT.BUG) GOTO 2023
C      IF(EXIT.AND.MOD(KK,I1).EQ.0.AND.TOP.AND.BUG) THEN
C      WRITE(6,YTESD)
C      IF(.NOT.BUG3) GOTO2025
C      IF(BUG3) THEN
C      WRITE(6,7001)
C      FORMAT(10ARRAY OF POTENTIAL CORRECTIONS,BLADE SURFACE=FAR RIGHT',
1, COLUMN,1)
C      DO 732 I2=1, IEXIT
C      WRITE(6,91)(DELPV(I2,J2),J2=167,M)
C      WRITE(6,7002)
C      FORMAT(10ARRAY OF POTENTIAL VALUES,BLADE SURFACE=FAR RIGHT ',
7001
732
7002

```

```

1' COLUMN')
DO 738 I2=1, IEXIT
WRITE(6,9) (PHI(I2, J2), J2=167, M)
END
2025 WRITE(6,7003)
7003 FORMAT(10ARRAY OF RESIDUAL MASS FLOWS, BLADE SURFACE=FAR RIGHT',
1' COLUMN')
DO 745 I2=1, IEXIT
WRITE(6,9) (ENOLF(I2, J2), J2=167, M)
END
7004 FORMAT(10ARRAY OF MACH NUMBERS, BLADE SURFACE=FAR RIGHT COLUMN')
DO 750 I2=1, IEXIT
WRITE(6,9) (EMACH(I2, J2), J2=167, M)
REWIND 8
WRITE(8,2) PHI
END
9 FORMAT(1H, 14E9.3)
END

C DO ( DELPHI COEFFICIENTS AND TRANSPOSED TERMS)
C PROCEDURE ( DELPHI COEFFICIENTS AND TRANSPOSED TERMS)
C AIJPI*DELPHI(I, J+1)+AIJ*DELPHI(I, J)+AIJMI*DELPHI(I, J-1)= R + BIGZ

DELPM1(2) = DELPHI(J)
IF(I.EQ.2) DELPM1(2)=DELPV(IEXIT-1, J)
IF(KK.GT.2.AND.MOD(KK,1).NE.0) GO TO 2029
IF(KK.LE.2.OR.MOD(KK,1).EQ.0) THEN
AIJPI=(T(2)*RHO(2)+T(3)*RHO(3))/ZETDF1
AIJMI=(T(6)*RHO(6)+T(7)*RHO(7))/ZETDF2
TI18=(T(1)*RHO(1)+T(8)*RHO(8))/ETADF1
ZIMIJ=(T(4)*RHO(4)+T(5)*RHO(5))/ETADF2
AIJ=-1.*(AIJMI+AIJPI+TI18+ZIMIJ)
AIPIJ=0.
ZJPI=0.
ZJMI=0.

C 2023
C SAVE COEFFICIENTS UNTIL THE NEXT TIME DENSITY IS UPDATED (WHEN KK IS
C DIVISIBLE BY IT).
SUPERG(43, J, I)=AIJ
SUPERG(44, J, I)=AIJMI
SUPERG(45, J, I)=AIJPI
SUPERG(46, J, I)=ZIMIJ
ZIMIJ=ZIMIJ*DELPM1(2)

```

```

C      SINCE (KK.LE.2.OR.MOD(KK,IT).EQ.0)
      GOTO 2030
C      IF ITERATION COUNT EXCEEDS 2 AND IS NOT DIVISIBLE BY IT, OBTAIN
C      COEFFICIENTS FROM THE LAST CALCULATION.
2029  AIJ=SUPERG(43,J,I)
      AIJMI=SUPERG(44,J,I)
      AIJPI=SUPERG(45,J,I)
      ZIMIJ=ZIMIJ*DELPMI(2)
      ZJPI=0.
      ZJMI=0.
      END
C2030  BIGZ = -(ZIPIJ + ZIMIJ + ZJPI + ZJMI)
      IF(J.EQ.4)BIGZ=8IGZ-AIJMI*DELPV(TEXIT+2-I,5)
      TRANSP = R +BIGZ
C      ELIMINATE CHANGES TO PHI(BY ZEROING COEFFICIENTS)ON ALL DUMMY LINES
C      BEYOND PERIODIC LINE.
      IF(.NOT.(J.LT.4.OR.(J.LE.5.AND.
1(I.LE.3.OR.I.GE.TEXIT-1.OR.I.EQ.NOZ-1.OR.
1I.EQ.NOZ.OR.I.EQ.NOZ+1)))) GOTO 2031
C      AIJ=0.
      AIJMI=0.
      AIJPI=0.
      TRANSP=0.
C2031  STORE COEFFICIENTS FOR LATER TRIDIAGONAL SOLUTION OF THIS LINE
      PRIORS(11,J)=AIJ
      PRIORS(12,J)=AIJPI
      PRIORS(13,J)=AIJMI
      PRIORS(14,J)=TRANSP
C      PROCEED TO NEXT VALUE OF J ON THIS RADIATING LINE
C      IF(.NOT.TOP)GOTO275
      END
C      DO (FORWARD ELIMINATION)
C      PROCEDURE (FORWARD ELIMINATION)
      DO 828 J=1,M
      JM1=J-1
      IF(JM1.EQ.0) JM1=2

```

QSO15370
QSO15380
QSO15390
QSO15400
QSO15410
QSO15420
QSO15430
QSO15440
QSO15450
QSO15460
QSO15470
QSO15480
QSO15490
QSO15500
QSO15510
QSO15520
QSO15530
QSO15540
QSO15550
QSO15560
QSO15570
QSO15580
QSO15590
QSO15600
QSO15610
QSO15620
QSO15630
QSO15640
QSO15650
QSO15660
QSO15670
QSO15680
QSO15690
QSO15700
QSO15710
QSO15720
QSO15730
QSO15740
QSO15750
QSO15760
QSO15770
QSO15780
QSO15790
QSO15800
QSO15810
QSO15820
QSO15830
QSO15840

```

BOTTOM = .FALSE.
TOP = .FALSE.
BLADE = .FALSE.
IF(J.EQ.1) BOTTOM = .TRUE.
IF(J.EQ.M) TOP = .TRUE.
BLADE = TOP

```

C RECOVER COEFFICIENTS

```

AIJ=PRIORS(11,J)
AIJPI=PRIORS(12,J)
AIJMI=PRIORS(13,J)
TRANSP=PRIORS(14,J)
IF(BOTTOM) GOTO 828
IF(.NOT.BOTTOM) THEN
  FFAC=AIJ-AIJMI*E(JMI)
  E(J)=+AIJPI/FFAC
  F(J)= (TRANSP - AIJMI*F(JMI))/FFAC
  FORMAT(9E13.5)
END

```

3333

C

828 CONTINUE

C

```

IF(I.NE.1)
  100 (BACK SUBSTITUTION)

```

C

C PROCEDURE (BACK SUBSTITUTION)

```

DELPHI(M)=F(M)
OV=OVEREL
IF(KKK.LT.1) THEN OV=UNDERL
IF(EMACH(I,M).GE.1) OV=SUPREL*(1.-D(M)*(1.-CI))
DELPSV(I,M)=DELPHI(M)*OV
J=M-1
IF(J.LT.1) GOTO 857
FORMAT(3I6,9E12.5)
BOTTOM = .FALSE.
TOP = .FALSE.
BLADE = .FALSE.
IF(J.EQ.1) BOTTOM = .TRUE.
IF(J.EQ.M) TOP = .TRUE.
BLADE = TOP
IF(.NOT.BOTTOM) DELPHI(J)=E(J)*DELPHI(J+1)+F(J)
OV=OVEREL
IF(KKK.LT.1) THEN OV=UNDERL

```

856

3335

```

      IF(EMACH(I,J).GE.1.)OV=SUPREL*(1.-D(J))*(1.-CI)
      DELPSV(I,J)=DELPHI(J)*OV
      J=J-1
      GOT0856
      C CONTINUE
      C END
      J=0
      C GO TO NEXT RADIATING LINE
      IF(.NOT.EXIT)GOTO272
      TOT=0.
      C IF ALL LINES ARE COMPLETED, THEN
      C DO (UPDATE OF FIELD VARIABLES)
      C PROCEDURE (UPDATE OF FIELD VARIABLES)
      DO 871 I =2, IEXIT
      DO 871 J1=1,M
      PHI(I,J1)=PHI(I,J1)+DELPSV(I,J1)
      IF(J1.GE.4.AND.I.GT.1.AND.(ABS(DELPSV(I,J1)).LT.TEST*AVGFAC*GUESS
      1/FLOAT(IEXIT-1))/FLOAT(M-3).OR.KKK.LT.NOTVET))
      I/TOT=TOT+ABS(DELPSV(I,J1))
      SSFAC=RADE/FIXR*SS*ACQINF1*SIN(LAMDAO+BETA1)
      DO 880 I1=2, IEXIT
      IF(I1.EQ.NOZ) GOTO 880
      IF(I1.NE.NOZ) THEN
      SFAC=SSFAC*FLOAT(ISIGN(1,I1-NOZ))
      C CORRECT THE DUMMY LINES BEYOND PERIODIC BOUNDARY USING PERIODICITY
      C Y CONDITION.
      DO 878 J1=1,3
      PHI(I1,J1)=PHI(IEXIT+2-I1,8-J1)+SFAC
      C END
      C CONTINUE
      C UPDATE DUMMY LINES ON EACH SIDE OF BRANCH C
      DO 890 J1=1,M
      PHI(I,J1)=

```

203

C PROCEDURE (SURFACE FLOW CALCULATION FOR FINISHED GRID)

```
WRITE(6,781) M3,ITIEX
ICOUNT=0
IDEX=1
DO 937 I=1, IEXIT
  IIM=I-1
  IM1=IIM
  IF (IM1.EQ.0) IM1=2
  FIRSTJ=M-2
  LINEI=1
```

C DO (VELOCITY CALCULATION ALONG RADIATING LINE)
CALL SPEEDS(LINEI,FIRSTJ,

```
937 IF (IIM.GT.0)
  1WRITE(6,782) IIM,XOCX,FADBOP,X(I,M),Y(I,M),D(M),BOPR,
  1PHI(I,M),E(M),F(M),DELPHI(M)
781 1FORMAT(1H0/1H0,22X,
  1'FINAL FLOW CALCULATION ON BLADE SURFACE FOR THE ',I3,' LINE NO. ',
  1' BY ',I3,' MESH '//,
  1' RADIATING',50X,' STATIC MACH PRES ',
  1' X/CX Y/CX X YVEL DENSITY '//)
  1COEF PHI XVEL
  1FORMAT(16,4X,10F10.5)
  1END
```

RETURN

ENTRY KEEPER

C DO (OUTPUT OF PHI AND CALCULATION OF FINAL VELOCITY FIELD)

C PROCEDURE (OUTPUT OF PHI AND CALCULATION OF FINAL VELOCITY FIELD)

```
10 REWIND 7
  10FORMAT(5E15.8)
  10ENT=1.
```

QSO16810
QSO16820
QSO16830
QSO16840
QSO16850
QSO16860
QSO16870
QSO16880
QSO16890
QSO16900
QSO16910
QSO16920
QSO16930
QSO16940
QSO16950
QSO16960
QSO16970
QSO16980
QSO16990
QSO17000
QSO17010
QSO17020
QSO17030
QSO17040
QSO17050
QSO17060
QSO17070
QSO17080
QSO17090
QSO17100
QSO17110
QSO17120
QSO17130
QSO17140
QSO17150
QSO17160
QSO17170
QSO17180
QSO17190
QSO17200
QSO17210
QSO17220
QSO17230
QSO17240
QSO17250
QSO17260
QSO17270
QSO17280

```

131 EX1=1.
    IF(ALLOUT)WRITE(6,131) M3,ITEX
    FORMAT(1H1,38X,' CALCULATED FLOW FIELD AT ALL GRID POINTS '/1H0,
138X,
1. SURFACE CONTOUR NO. 1 = PERIODIC BOUNDARY'/40X,'SURFACE CONT',
1. OUR NO.
1.3 = BLADE SURFACE'/1H0,31X,' RADIATING LINE NO. 1 BEGINS AT',
1.3 DOWNSTREAM,
1. INFINITY.'/42X,' AND IS REPEATED AS LINE NO. ',13)
    IF(ALLOUT)WRITE(6,13)
13 FORMAT(1H0,15X,'SURFACE ',74X,'FLOW STATIC'/16X,'CONTOUR ',
1. X/CX
1. PHI
1. PRES COEF.'/15X,' NUMBER',
1. XVEL
1. YVEL
1. DENSITY
1. MACH
1. ANGLE',
1. INDEX=1
1. COUNT=0
1. 969 I=1, IEXIT
1. M=I-1
1. IF(I.GT.1.AND.ALLOUT)WRITE(6,14) 11M
14 FORMAT(1. ORADIATING LINE.'/ NUMBER ',13)
1. M1=I-1
1. IF(1M1.EQ.0) 1M1=2
1. FIRSTJ=1
1. LINEI=1

C DO (VELOCITY CALCULATION ALONG RADIATING LINE)
  CALL SPEEDS(LINEI,FIRSTJ)

969 CONTINUE
972 DO 972 I=2, IEXIT
975 E(I)=5*(DELPSV(I,4)+DELPSV(IEXIT+2-I,4))
    DELPSV(I,4)=E(I)
    REWIND 13
    MM3=M-3
    NNOZ=NNOZ-1

C WRITE THE PLOT DATA SAVE FILE

1 WRITE(13,1) NNOZ,MM3
  FORMAT(13,1)
  WRITE(13,2) ((X(I,J),I=2,NZZ),J=4,M),((Y(I,J),I=2,NZZ),J=4,M),
  WRITE(13,2) ((DUMBX(I,J),I=2,NZZ),J=4,M),((DUMBY(I,J),I=2,NZZ),
  1J=4,M),((DELPSV(I,J),I=2,NZZ),J=4,M)
  WRITE(13,2) ((TENOLF(I,J),I=2,NZZ),J=4,M)

```

```

QSO17290
QSO17300
QSO17310
QSO17320
QSO17330
QSO17340
QSO17350
QSO17360
QSO17370
QSO17380
QSO17390
QSO17400
QSO17410
QSO17420
QSO17430
QSO17440
QSO17450
QSO17460
QSO17470
QSO17480
QSO17490
QSO17500
QSO17510
QSO17520
QSO17530
QSO17540
QSO17550
QSO17560
QSO17570
QSO17580
QSO17590
QSO17600
QSO17610
QSO17620
QSO17630
QSO17640
QSO17650
QSO17660
QSO17670
QSO17680
QSO17690
QSO17700
QSO17710
QSO17720
QSO17730
QSO17740
QSO17750
QSO17760

```

```

IF(BUG3)
1WRITE(6,2)((X(I,J),I=2,NZZ),J=4,M),((Y(I,J),I=2,NZZ),J=4,M)
IF(BUG3)
1WRITE(6,2)((DUMBX(I,J),I=2,NZZ),J=4,M),((DUMBY(I,J),I=2,NZZ),
1J=4,M)
END

```

C

```

RETURN

```

```

END
SUBROUTINE SPEEDS(I,MM2)

```

CC
CC
CC
CC
CC

```

VELOCITY CALCULATION ALONG ONE RADIATING LINE (I), BEGINNING AT J
= MM2 AND GOING TO J=M (THE BLADE SURFACE). ARRAYS DELPHI, DELPS
V D,E,F,PRIORS,DUMBY, AND DUMBX ARE REUSED (OVERWRITTEN) IN THE PRO
CESS.

```

```

LOGICAL INLET, BLADE, EXIT, BOTTOM, TOP
LOGICAL DOSURF, SMOOTH, FINEST, RESTAR, ALLOUT
REAL LAB, LAD, MINF, LENGTH, JACOB, LAMDAO
REAL MACH(8)
DOUBLE PRECISION CAPK, CAPKP
COMMON/PARAM/RESTAR, NOZIN, BETA1, BETA2, QINF1, CI, CII, GAMMA, MINF, EM
1, DELTA, BETA, TOL, BUG, BUG2, BUG3, IT, GUESS, OVERREL, FIXR, NOWREL, NOTVET
1, DAMP, SUPREL, S, WAKE, B2OB1, UNDERL, RADRAI, RADLE, OMEGA, FLOCO, VAXIAL,
1 ALLOUT
COMMON/QUASI/ RADIUS(100,30), HEIGHT(100,30)
COMMON/ENTIRE/ X(100,30), Y(100,30), PHI(100,30), DELPSV(100,30)
COMMON/GEOM2/ZETA(100), ETA(100)
COMMON/CROSS/ ZMACH(100,30)
COMMON/GEOM/NOZ,M,RLE,RFE,A,CHORD,XUPS,XONS,LAMDAO,CC,CAPK,
1CAPKP,PI
1,RNK,VB,KN,NED,BB,DOSURF,RFAC,SOLID,MPLS
1,STABAC,SLP1,SLP2,SLP3,SLP4,SMOOTH,CHOP,THETT,THETL
1,FINEST

```

QSO17770
QSO17780
QSO17790
QSO17800
QSO17810
QSO17820
QSO17830
QSO17840
QSO17850
QSO17860
QSO17870
QSO17880
QSO17890
QSO17900
QSO17910
QSO17920
QSO17930
QSO17940
QSO17950
QSO17960
QSO17970
QSO17980
QSO17990
QSO18000
QSO18010
QSO18020
QSO18030
QSO18040
QSO18050
QSO18060
QSO18070
QSO18080
QSO18090
QSO18100
QSO18110
QSO18120
QSO18130
QSO18140
QSO18150
QSO18160
QSO18170
QSO18180
QSO18190
QSO18200
QSO18210
QSO18220
QSO18230
QSO18240

```

COMMON/CALVEL/FAKEFI(4,6), FAKFI(4,6), RHOCON, QIM, EXPO, ROTATN,
1 ROTROT(1M), IEXIT, PRIORS(14,30), INDEX(100,30), IDEX(32), FAKEU(100),
1 FAKEV(100), ICOUNT, IDEX, AMY, AMZ
COMMON/VELOUT/D(100), E(100), F(100), DELPHI(100), TENOLF(100,30),
1 DUMBX(100,30), DUMBY(100,30), BQPR, FADBOP, XOCX, KK
1 DIMENSION RHOS(8), FI(8), FI2(3), FI3(3), FI9(8), SPAN(8),
1 EMACH(100,30)
2 COSX(8), COSY(8), LENGTH(8), DEDX(8), DZDX(8), DEDY(8),
3 ZDY(8), F(8), DPHIDZ(8), DPHIDE(8), U(8), V(8), UN(8), RHO(8),
4 RHOUL(8), ONOFF(8), DELPM1(3)
5 HITE(8)

COMMON/HIBALL/ SUPERG(46,30,100)

XDIF(A,B,C,D)=-.5*(A+B-C-D)

PROCEDURE (VELOCITY CALCULATION ALONG RADIATING LINE)
DO (AXIAL LOCATION INDICATORS)
INLET=.FALSE.
EXIT=.FALSE.
IF(I.EQ.1) INLET=.TRUE.
IF(I.EQ. IEXIT) EXIT=.TRUE.
XLOW=0.
XHI=0.
YLOW=1.E10
DO 1111 I=1, IEXIT
IF(XLOW.LT.X(I1,M)) GOTO11
XLOW=X(I1,M)
IF(XHI.GT.X(I1,M)) GOTO111
XHI=X(I1,M)
IF(YLOW.LT.Y(I1,M)) GOTO1111
YLOW=Y(I1,M)
CONTINUE
DO 1427 J=MM2,M
J3M=J-3
JMI=J-1
IF(JMI.EQ.0) JMI=2
DO (VERTICAL LOCATION INDICATORS)
BOTTOM=.FALSE.
TOP=.FALSE.
BLADE=.FALSE.
IF(J.EQ.1) BOTTOM=.TRUE.
IF(J.EQ.M) TOP=.TRUE.
BLADE=.TOP

```

QSO18250
 QSO18260
 QSO18270
 QSO18280
 QSO18290
 QSO18300
 QSO18310
 QSO18320
 QSO18330
 QSO18340
 QSO18350
 QSO18360
 QSO18370
 QSO18380
 QSO18390
 QSO18400
 QSO18410
 QSO18420
 QSO18430
 QSO18440
 QSO18450
 QSO18460
 QSO18470
 QSO18480
 QSO18490
 QSO18500
 QSO18510
 QSO18520
 QSO18530
 QSO18540
 QSO18550
 QSO18560
 QSO18570
 QSO18580
 QSO18590
 QSO18600
 QSO18610
 QSO18620
 QSO18630
 QSO18640
 QSO18650
 QSO18660
 QSO18670
 QSO18680
 QSO18690
 QSO18700
 QSO18710
 QSO18720

Q5019210
Q5019220
Q5019230
Q5019240
Q5019250
Q5019260
Q5019270
Q5019280
Q5019290
Q5019300
Q5019310
Q5019320
Q5019330
Q5019340
Q5019350
Q5019360
Q5019370
Q5019380
Q5019390
Q5019400
Q5019410
Q5019420
Q5019430
Q5019440
Q5019450
Q5019460
Q5019470
Q5019480
Q5019490
Q5019500
Q5019510
Q5019520
Q5019530
Q5019540
Q5019550
Q5019560
Q5019570
Q5019580
Q5019590
Q5019600
Q5019610
Q5019620
Q5019630
Q5019640
Q5019650
Q5019660
Q5019670
Q5019680

```

C      T(31)=SUPERG(41,J,I)
      T(8)=SUPERG(42,J,I)
      CELL=I, POINT C
      T(5)=PRIORS(1,J)*ONOFF(5)
      T(4)=PRIORS(2,J)*ONOFF(4)
      PRIORS(1,J)=-T(1)
      PRIORS(2,J)=-T(1)
      DO (FIELD, GRAZIENT) 3
      CELL=I, I1=I+1, J1=J+1
      FI1(I1)=PHI(I+1,J+1)
      FI2(I1)=PHI(I+1,J+1)
      FI3(I1)=PHI(I+1,J+1)
      FI4(I1)=PHI(I+1,J+1)
      FI5(I1)=PHI(I+1,J+1)
      FI6(I1)=PHI(I+1,J+1)
      FI7(I1)=PHI(I+1,J+1)
      FI8(I1)=PHI(I+1,J+1)
      FI9(I1)=PHI(I+1,J+1)
      I10=I-1, IEXIT+3
      END
C      CELL=I, POINT A
      DPHIDZ(1)=XDIF(FI3(1),FI2(1),FI9(1),FI1(1))/ZETDF1
      DPHIDE(1)=(FI1(1)-FI9(1))/ETADF1
      CELL=I, POINT B
      DPHIDE(2)=XDIF(FI1(2),FI2(2),FI9(2),FI3(2))/ETADF1
      DPHIDZ(2)=(FI1(2)-FI9(2))/ZETDF1
      CELL=I, POINT 8
      DPHIDE(3)=XDIF(FI5,FI4,FI9(3),FI3(3))/ETADF2
      DPHIDZ(3)=(FI3(3)-FI9(3))/ZETDF1
      CELL=IV, POINT A
      DPHIDE(8)=(FI1(8)-FI9(8))/ETADF1
      DPHIDZ(8)=XDIF(FI7,FI8,FI9(8),FI1(8))/ZETDF2
      DO (BOUNDARY CONDITION SWITCHES)
      DO 1306 I1=1,8
      ONOFF(I1)=1
      IF(.NOT. INLET.AND..NOT.BOTTOM.AND..NOT.TOP ) GOTO2060
      IF(TOP) IS=1
      IF(INLET) IS=3
      IF(BOTTOM) IS=5
      IE=IS+3
      DO 2059 I1=IS,IE
      ONOFF(I1)=0
      DO (VELOCITIES AND DENSITIES)
      U(6)=U(3)

```

```

V(6) = V(3)
V(7) = U(2)
DO 1337 I1 = 1, 8
  IF (I1 .GE. 4 .AND. I1 .NE. 8) GOTO 1337
  IF (I1 .LT. 4 .OR. I1 .EQ. 8) THEN
    U(I1) = DPHIDE(I1) * DEDX(I1) + DPHIDZ(I1) * DZDX(I1)
    V(I1) = DPHIDE(I1) * DEDY(I1) + DPHIDZ(I1) * DZDY(I1)
    IF (INDEX(I1, J) .EQ. 0) GOTO 5411
    ICOUNT = ICOUNT + 1
    IIS = INDEX(I1, J)
    IF (I1 .LE. IIS) IIP = IIS
    IF (I1 .EQ. IIS) IIP = IIP - 1
    IF (I1 .EQ. IIP) IIP = 8
    IF (I1 .NE. IIP) GOTO 5410
    IF (I1 .P. GT. 1) DIF .AND. IIP .NE. 8) GOTO 5410
    V(I1) = FAKEV(IDEX)
    U(I1) = FAKEU(IDEX)
    IDEX = IDEX + 1
    IIP = IIP + 1
  IF (I1 .NE. 8) ICOUNT = ICOUNT - 1
  V(I1) = V(I1) - SPAN(I1) * FIXR/RADLE * ROTATN
  UN(I1) = U(I1) * COSX(I1) + V(I1) * COSY(I1)
END
CONTINUE
GOTO 2064
IF (.NOT. TOP) THEN
  U(8) = (DPHIDE(8) * ETADFI - RADIUS(I, J) / RADLE * ROTATN * YDIFI) * EXDIFI / SLEN
  V(8) = (DPHIDE(8) * ETADFI - RADIUS(I, J) / RADLE * ROTATN * YDIFI) * YDIFI / SLEN
  ROTATX = ROTATN * RADIUS(I, J) / RADLE
  U8 = SQR(T((DPHIDE(8) * ETADFI / EXDIFI) ** 2 + (DPHIDE(8) * ETADFI / YDIFI -
    1 - ROTATX) ** 2) * EXDIFI / SQR(SLEN)
  U8 = SQR(T(U(8) * U(8) + V(8) * V(8)) * ABS(EXDIFI) / SQR(SLEN)
  U(8) = SIGN(U8, U(8))
  V(8) = SIGN(V8, V(8))
  UN(8) = U(8) * COSX(8) + V(8) * COSY(8)
END
GOTO 2064
U(4) = PRIORS(3, J)
U(5) = PRIORS(4, J)
V(4) = PRIORS(5, J)
V(5) = PRIORS(6, J)
PRIORS(3, J) = U(1)
PRIORS(4, J) = U(8)
PRIORS(5, J) = V(1)
PRIORS(6, J) = V(8)
RHO(7) = RHO(2)
RHO(6) = RHO(3)
RHO(4) = PRIORS(7, J)

```

```

RHO(5)=PRIORS(8,J)
ONSUM=0.
DO 1391 I1=1,8
IF(KK.LE.2.OR.MOD(KK,2).EQ.0) THEN
IF(I1.GE.4.AND.I1.NE.8) GOTO 2066
IF(I1.LT.4.OR.I1.EQ.8) THEN
RHO(5)=1.
RHO(5)=1.
IF(I1.EQ.1.EQ.I.EXIT).AND.J.LE.4.AND.ROTATN.NE.0.
1 OR.KK.LF.NOTYET) GOTO 2066
QSQR=U(I1)*V(I1)+(V(I1)*V(I1))
IF(QSQR.GE.QLIM)QSQR=.99*QLIM
RHO(5)=RHOCON*(1.-QSQR)*ONOFF(I1)+1.
RHO(5)=1.-AMZ*(QSQR-1.)+AMY*(QSQR-1.)*(QSQR-1.)
IF(QSQR.LT..65.OR.QSQR.GT.1.44)RHO(5)=RHO(5)*EXP
ROTATX=SPAT(I1)*FIXR
IF(ROTROT.NE.0.:AND.J.GT.3)
1RHO(5)=(RHO(5))*((1./EXPO)+RHOCON*ROTROT*((ROTATX/RADLE)
1#2-1.)*ONOFF(I1))*EXP
I#2-1.)*ONOFF(I1))*EXP
END
END
QSQR=U(I1)*V(I1)+V(I1)*V(I1)
IF(I1.GT.1)MINF=SQRT(QSQR)/RHO(5)**(1./EXPO)
1MACH(I1).GT.2.)MACH(I1)=2.
IF(MACH(I1).GT.2.)MACH(I1)=2.
CONTINUE
PRIORS(7,J)=RHO(5)
PRIORS(8,J)=RHO(8)
IF(I1.LE.1.OR.J.LE.3) GOTO 1427
IF(I1.GT.1.AND.J.GT.3) THEN
F(J)=1.
F(J)=0.
ONSUM=ONOFF(1)+ONOFF(8)+ONOFF(4)+ONOFF(5)
E(J)=(U(1)*ONOFF(1)+U(8)*ONOFF(8)+U(4)*ONOFF(4)+U(5)
1*ONOFF(5))/ONSUM
F(J)=(V(1)*ONOFF(1)+V(8)*ONOFF(8)+V(4)*ONOFF(4)+V(5)
1*ONOFF(5))/ONSUM
DUMBY(I,J)=E(J)
DUMBY(I,J)=F(J)
EFFAC=1.-E(J)*E(J)-F(J)*F(J)
DELPHI(J)=(RHO(5)*ONOFF(1)+RHO(8)*ONOFF(8)+RHO(4)*ONOFF(4)+RHO(5)
1*ONOFF(5))/ONSUM
D(J)=0
D(J)=MINF*SQRT((F(J)*F(J)+E(J)*E(J))/DELPHI(J))*((1./EXPO))
IF(J.LE.6.AND.(I.LE.4.OR.I.GE.IEXIT-2))D(J)=EM
IF(J.LE.6.AND.I.GE.NOZ-1.AND.I.LE.NOZ+1)D(J)=MINF
DELPSV(I,J)=D(J)
END

```

```

2076 IF(E(J).NE.0.)BOPS=ATAN2(F(J),E(J))*180./PI
      BOPR=-999
      IF(D(J).NE.0.)
        1BOPR=2./GAMMA*(DELPHI(J)*(1.-EFFAC)/D(J)/D(J)-1./MINF/MINF)
        IF(MINF.LT.-25)BOPR=1.+DELPHI(J)*(EFFAC-1.)
        XOCX=(X(I,J)-XLOW)/(XHI-XLOW)
        FADBOP=(Y(I,J)-YLOW)/(YHI-YLOW)
        IF(ALLOUT.AND.MM2.EQ.1)
          1WRITE(6,12) J3M,XOCX,FADBOP,PHI(I,J),E(J),F(J),DELPHI(J)
        1D(J),BOPS,BOPR
        TENOLF(I,J)=BOPR
      END
      C
      1427 CONTINUE
      12 FORMAT(15X,14,4X,9F10.5)
      RETURN
      END
      SUBROUTINE WRAPUP(ZETA,ETA,S)
      C
      GEOMETRY GENERATION FOR CASCADE POTENTIAL FLOW SOLVER
      LOGICAL DOSURF,SMOOTH,BUG2,ORTHO
      REAL LAMDAO
      COMPLEX HI,H2,ORIGIN,COWISE
      COMPLEX AI/(0,1.)/
      DOUBLE PRECISION CAPK,CAPKP
      DIMENSION ZETA(100),EFA(100),DUMBE(100),H1(100),H2(100)
      DIMENSION XB(200),YB(200),DUMBX(100,30)
      COMMON/ENTIRE/ X(100,30),Y(100,30),PHI(100,30),DUMBY(100,30)
      COMMON/GEOM/NOZ,M,RLE,RTE,A,CHORD,XUPS,XDNS,LAMDAO,CC,CAPK
      CAPKP,PI
      1,RNK,V8,KN,NED,8B,DOSURF,RFAC,SOLID,MPLS
      1,STABAC,SLP1,SLP2,SLP3,SLP4,SMOOTH,CHOP,THETT,THETL
      NAMELIST/INSTUF/H1,H2,IMAXY,BUG2
      DATA ORTHO/.FALSE./,BUG2/.FALSE./
      MAGIC=0
      MAGICM=0
      REWIND 18
      RECORD OLD COMPUTATIONAL VARIABLES FOR INTERPOLATION IN FILPHI
      WRITE(18) ETA,ZETA
      PI=3.14159265

```

```

SO=CHORD
C   IF (NDZ.LT.0) OBTAIN MESH FROM STORAGE FILE
C
C   IF (NOZ.LE.0) GOTO201
C   IF (NOZ.GT.0) GENERATE A NEW MESH
C   IEXIT=2*NOZ-1
C   NZZ=1EXIT
C   DO SURF=.LE.0) DO SURF=.FALSE.
C   IF (NED.LE.0) NED=1
C   IF (DO SURF) READ(5,INSTUF,END=15)
C   WRITE(6,12) THE FOLLOWING IS THE FIRST ECHO PRINT)
C   FORMAT(37H)
C   WRITE(6,15) INSTUF)
C   IF (DO SURF) READ(5,INSTUF,END=16)
C   WRITE(6,11)
C   FORMAT(38H) THE FOLLOWING IS THE SECOND ECHO PRINT)
C   WRITE(6,11) INSTUF)
C   IF (KN.EQ.0) ORTHO=.TRUE.
C   IF (ORTH) GOTO203
C   IF (.NOT.ORTH) REARRANGE H2 INPUT INTO H1(UPPER SURFACE), AND H2(L
C   OWER SURFACE)
C   UNSTAGGERED DATA TABLES, FOR THE INTERPOLATION SCHEME
C
C   NEDKN1=NED-KN+1
C   ORIGIN=H2(NEDKN1)
C   DO 205 I1=1,KN
C   I5=NED-KN+I1
C   H1(I1)=H2(I5)
C   CDWISE=(H1(I1)-ORIGIN)*CEXP(-AI*LAMDAO)
C   H1(I1)=CDWISE
C   CONTINUE
C   DO 206 I1=1,NEDKN1
C   I6=NED-KN-I1+2
C   IF (I6.LE.0) I6=1
C   H1(KN+I1)=H2(I6)
C   DO 209 I1=1,NEDKN1
C   H2(I1)=H1(KN+I1)
C   CDWISE=(H2(I1)-ORIGIN)*CEXP(-AI*LAMDAO)
C   H2(I1)=CDWISE
C   CONTINUE
C   NED=NED+1
C   GOTO233
C
C   IF (ORTH) REARRANGE H2 INPUT TO START AT MINIMUM Y POINT RATHER T
C   HAN MAX X,

```

QSO21610
 QSO21620
 QSO21630
 QSO21640
 QSO21650
 QSO21660
 QSO21670
 QSO21680
 QSO21690
 QSO21700
 QSO21710
 QSO21720
 QSO21730
 QSO21740
 QSO21750
 QSO21760
 QSO21770
 QSO21780
 QSO21790
 QSO21800
 QSO21810
 QSO21820
 QSO21830
 QSO21840
 QSO21850
 QSO21860
 QSO21870
 QSO21880
 QSO21890
 QSO21900
 QSO21910
 QSO21920
 QSO21930
 QSO21940
 QSO21950
 QSO21960
 QSO21970
 QSO21980
 QSO21990
 QSO22000
 QSO22010
 QSO22020
 QSO22030
 QSO22040
 QSO22050
 QSO22060
 QSO22070
 QSO22080

```

C      FOR ELECTROSTATIC ANALOG SCHEME
203      T2=1.E10
          IT2=0
          DO 2333 I1=1,NED
            IF (AIMAG(H2(I1)).GE.T2) GOTO2333
            IT2=I1
            T2=AIMAG(H2(IT2))
          CONTINUE
2333      DO 2333 I1=1,NED
          I7=IT2+I1-1
          IF (I7.EQ.NED) IT3=I1
          IF (I7.GT.NED) I7=I1-IT3+1
2333      H1(I1)=H2(I7)
          T2=-1.E10
          IT2=0
          DO 2033 I1=1,NED
            IF (AIMAG(H1(I1)).LE.T2) GOTO2033
            T2=AIMAG(H1(I1))
            IT2=I1
2033      H2(I1)=H1(I1)

IMAXY=I2T

C      LOAD COMPLEX INPUT INTO XBOD AND YBOD ARRAYS

233      CONTINUE
          DO 213 K=1,NED
            KMKN=K-KN
            IF (K.GT.KN) GOTO217
            IF (K.LE.KN) THEN
              XB(K)=REAL(H1(K))
              YB(K)=AIMAG(H1(K))
              GOTO213
            ELSE
              XB(K)=REAL(H2(KMKN))
              YB(K)=AIMAG(H2(KMKN))
            CONTINUE
217      IF (CHORD.EQ.1.) GOTO219
213      IF (CHORD.NE.1.) NORMALIZE COORDINATES TO AERODYNAMIC CHORD
C          DO 221 I2=1,NED
          XB(I2)=XB(I2)/CHORD
          YB(I2)=YB(I2)/CHORD
          RTE=RTE/CHORD
          S=S/CHORD
          CHORD=1.
C      END
  
```

```

219 C IF(.NOT.ORTHO) GOTO225
C IF(ORTHO) DO ELECTROSTATIC ANALOG GRID
DO 227 I2=1,NED
HI(I2)=CMPLX(XB(I2),YB(I2))
WRITE(6,9)
FORMAT(1H1,40X,' ELECTROSTATIC ANALOG GRID GENERATED')
CALL PJGRID(ZETA,ETA,HL,IMAXY,S)
GOTO226
C OTHERWISE DO INTERPOLATION SCHEME GRID
WRITE(6,10)
FORMAT(1H1,40X,' INTERPOLATION SCHEME GRID GENERATED')
CALL GRIDDL(ZETA,ETA,XB,YB,S)
C WRITE MESH POINTS STORAGE FILE
WRITE(13,1) NOZ,M,KN,NED
WRITE(13,2) ((X(I,J),I=1,NZZ),J=1,M),((Y(I,J),I=1,NZZ),J=1,M),
1,(ZETA(I),I=1,NZZ), (ZETA(J),J=1,M)
GOTO202
C FOR EXISTING MESH FILES, NORMALIZE PARAMETERS AND READ FILE
IF(CHORD.EQ.1.) GOTO231
IF(CHORD.NE.1.) THEN
RLE=RLE/CHORD
PTE=PTE/CHORD
S=S/CHORD
CHORD=1.
END
READ(23,1) NOZZ,MM,KN,NED
MAGIC=ABS((NOZZ-1)/(NOZ+1))
MAGICM=(MM-1)/(M-1)
M=MM
NOZ=NOZZ
IEXIT=2*NOZ-1
NOZZ=IEXIT
READ(23,2) ((X(I,J),I=1,NZZ),J=1,M),((Y(I,J),I=1,NZZ),J=1,M)

```

QSO222090
QSO222100
QSO222110
QSO222120
QSO222130
QSO222140
QSO222150
QSO222160
QSO222170
QSO222180
QSO222190
QSO222200
QSO222210
QSO222220
QSO222230
QSO222240
QSO222250
QSO222260
QSO222270
QSO222280
QSO222290
QSO222300
QSO222310
QSO222320
QSO222330
QSO222340
QSO222350
QSO222360
QSO222370
QSO222380
QSO222390
QSO222400
QSO222410
QSO222420
QSO222430
QSO222440
QSO222450
QSO222460
QSO222470
QSO222480
QSO222490
QSO222500
QSO222510
QSO222520
QSO222530
QSO222540
QSO222550
QSO222560

```

1,(ETA(I),I=1,NZZ),(ZETA(J),J=1,M)
C IF FINEST MESH IS DESIRED, SKIP SELECTION PROCESS
C IF(MAGIC.LE.1.AND.MAGICM.LE.1) GO TO 202
C IF(MAGIC.GT.1.OR.MAGICM.GT.1) SELECT AND LOAD GRID LINES DESIRED
C O FORM
A COARSE MESH
DO 235 J=1,M,MAGICM
JON=(J-1)/MAGICM+1
ZETA(JON)=ZETA(J)
DO 239 I=1,NZZ,MAGIC
ION=(I-1)/MAGIC+1
DUMBX(ION,JON)=X(I,J)
DUMBY(ION,JON)=Y(I,J)
IF(MOD(ION,2).EQ.0) ION=ION-1
239
C RADIATING LINE TO UPSTREAM INFINITY MUST BE INCLUDED
DUMBX(ION,JON)=X(NZZ,J)
DUMBY(ION,JON)=Y(NZZ,J)
NOZZ=(ION+1)/2
C RADIATING LINE TO DOWNSTREAM INFINITY MUST BE INCLUDED
DUMBX(NOZZ,JON)=X(NOZZ,J)
DUMBY(NOZZ,JON)=Y(NOZZ,J)
235
DO 243 I=1,ION
IOLD=(I-1)*MAGIC+1
ETA(I)=ETA(IOLD)
C BLADE SURFACE MUST BE LAST CLOSED CURVE
DUMBX(I,JON)=X(IOLD,M)
DUMBY(I,JON)=Y(IOLD,M)
C PERIODIC BOUNDARY MUST BE FIRST CLOSED CURVE
DUMBX(I,1)=X(IOLD,1)
DUMBY(I,1)=Y(IOLD,1)
243
DUMBX(ION,JON)=X(NZZ,M)
DUMBY(ION,JON)=Y(NZZ,M)
DUMBX(ION,1)=X(NZZ,1)
DUMBY(ION,1)=Y(NZZ,1)
DUMBX(NOZZ,M)=X(NOZZ,M)
DUMBY(NOZZ,M)=Y(NOZZ,M)
DUMBX(NOZZ,1)=Y(NOZZ,1)
DUMBY(NOZZ,1)=X(NOZZ,1)

```

QS022570
 QS022580
 QS022590
 QS022600
 QS022610
 QS022620
 QS022630
 QS022640
 QS022650
 QS022660
 QS022670
 QS022680
 QS022690
 QS022700
 QS022710
 QS022720
 QS022730
 QS022740
 QS022750
 QS022760
 QS022770
 QS022780
 QS022790
 QS022800
 QS022810
 QS022820
 QS022830
 QS022840
 QS022850
 QS022860
 QS022870
 QS022880
 QS022890
 QS022900
 QS022910
 QS022920
 QS022930
 QS022940
 QS022950
 QS022960
 QS022970
 QS022980
 QS022990
 QS023000
 QS023010
 QS023020
 QS023030
 QS023040

```

DO 251 J=1, JON
DO 251 I=1, ION
X(I,J)=DUMBX(I,I,J)
Y(I,J)=DUMBY(I,I,J)
Z(I,J)=DUMBZ(I,I,J)
ETA(I,ION)=ETA(NZZ)
ETA(ION)=ETA(NZZ)
NZZ=ION
IEXIT=NZZ
NOZ=NOZZ
M=JON

```

251

C END OF SELECTION PROCESS

202 CONTINUE
IF(.NOT.BUG2) GOTO255

C IF(BUG2) THEN

3 WRITE(6,3) NOZ,M,KN,NED
FORMAT(10NOZ=' ',13,' M= ',13,' KN= ',13,' NED= ',13)
NEDM1=NED-1

C FOR A NEW GEOMETRY GENERATION, CALCULATE THE SECOND DERIVATIVES
C N BLADE
C SURFACE USING SMOOTHED INPUT COORDINATES

```

DO 257 I=2, NEDM1
DUMBX(I,15)=1.E06
IF(XB(I),NE,XB(I-1))-AND.XB(I).NE.XB(I+1).AND.
1XB(I+1).NE.XB(I-1))
1DUMBX(I,15)=((YB(I+1)-YB(I))-YB(I-1))/
1(XB(I)-XB(I-1)))/(XB(I+1)-XB(I-1))*2.
CONTINUE

```

257

```

WRITE(6,5)
FORMAT(1H0,35X,' SMOOTHED BLADE SURFACE INPUT, HORIZONTAL CHORD')
FORMAT(1H0,35X,' X GRID POINTS, AS STORED ON MESH FILE')
FORMAT(1H0,20X,' Y GRID POINTS, AS STORED ON MESH FILE')
FORMAT(1H0,20X,' X GRID POINTS, AS STORED ON MESH FILE')
FORMAT(35X,14,3E13.5)
WRITE(6,4) (I,XB(I),YB(I),DUMBX(I,15),I=1,NED)
WRITE(6,6) (X(I,J),I=1,NZZ),J=1,M)
WRITE(6,7)

```

5

55

6

7

4

QSO23050
QSO23060
QSO23070
QSO23080
QSO23090
QSO23100
QSO23110
QSO23120
QSO23130
QSO23140
QSO23150
QSO23160
QSO23170
QSO23180
QSO23190
QSO23200
QSO23210
QSO23220
QSO23230
QSO23240
QSO23250
QSO23260
QSO23270
QSO23280
QSO23290
QSO23300
QSO23310
QSO23320
QSO23330
QSO23340
QSO23350
QSO23360
QSO23370
QSO23380
QSO23390
QSO23400
QSO23410
QSO23420
QSO23430
QSO23440
QSO23450
QSO23460
QSO23470
QSO23480
QSO23490
QSO23500
QSO23510
QSO23520

```

14  WRITE(6,2) ((Y(I,J),I=1,NZZ),J=1,M)
15  FORMAT(4I5)
2   FORMAT(1H,14E9.3)
    FORMAT(6E13.5)

255  IF(KN.NE.0)GOTO 261
    IF(OROTH) TRANSLATE THE ORIGIN OF X POINTS TO MIDCHORD (INTERP SCH
    EME
    C   ALREADY TRANSLATED
    C
    C   DO 263 I1=1,NZZ
    C   DO 263 J1=1,M
    C   X(I1,J1)=X(I1,J1)-.5
    C
    C   REARRANGE (X,Y) TABLES SO THAT X(2,J),Y(2,J) AND X(NZZ+1,J),Y(NZZ
    C   +1,J) GO
    C   TO DOWNSTREAM INFINITY, AND X(NOZ+1,J),Y(NOZ+1,J) GOES TO UPSTREA
    C   M INFINITY
    C   NUMBERING THE LINES CLOCKWISE AROUND THE BLADE. AT PRESENT, LINE
    C   1 GOES
    C   TO UPS. INF., NOZ GOES TO DOWNS. INF. AND NZZ TO UPS. INF.
    C
    C   ALSO REALIGN ETA SO THAT VALUES WILL GO FROM ZERO (LINE 2) TO 2*
    C   CAPKP (LINE
    C   NZZ+1), INSTEAD OF -CAPKP (LINE 1) TO +CAPKP (LINE NZZ).
    C
261  DO 271 I1=1,NZZ
    DO 271 J1=1,M
    I2=I1+NOZ-1
    DUMBE(I1)=-ETA(I2)
    DUMBX(I1,J1)=X(I1,J1)
    DUMBY(I1,J1)=Y(I1,J1)
    X(I1,J1)=X(I1+NOZ-1,J1)
    Y(I1,J1)=Y(I1+NOZ-1,J1)
    NOZP1=NOZ+1
    DO 279 I1=NOZP1,NZZ
    DO 279 J1=1,M
    I2=I1-NOZ+1
    DUMBE(I1)=-ETA(I2)+2.*ETA(1)
    DUMBX(I1,J1)=X(I1,J1)
    DUMBY(I1,J1)=Y(I1,J1)
    X(I1,J1)=DUMBX(I1-NOZ+1,J1)
    Y(I1,J1)=DUMBY(I1-NOZ+1,J1)
    C
    C   MOVE ALL VALUES UP 3 LOCATIONS IN J TO MAKE ROOM FOR 3 DUMMY ELLI
    C   PSES
    C

```

```

C      BEYOND THE PERIODIC LINE.
2088      I1=IEXIT
2955      IF(I1.LT.1)GOTO2872
2087      J1=M
          IF(J1.LT.1)GOTO2871
          X(I1,J1+3)=X(I1,J1)
          IF(I1.EQ.1)ZETA(J1+3)=ZETA(J1)
          Y(I1,J1+3)=Y(I1,J1)
287      J1=J1-1
          GOTO2087
2871      I1=I1-1
          GOTO2088

C      FINISH LOADING ETA: CREATE 3 DUMMY LINES USING PERIODIC RELATION
C      OF X,Y VALUES
2872      I1=IEXIT
295      IF(I1.LT.2)GOTO2951
          ETA(I1)=DUMBE(I1)
          DO 299 J1=1,3
              SIGEN=1+.NOZ/SIGEN=-1.
              IF(I1.EQ.NOZ)SIGEN=0.
              IF(I1.J1)=X(IEXIT-I1+1,8-J1)+S*SIN(LAMDAO)*SIGEN
              IF(I1.J1)=Y(IEXIT-I1+1,8-J1)+S*COS(LAMDAO)*SIGEN
299      ETA(I1+1)=ETA(I1)

C      LOAD SPECIAL POINTS (J=4 NOW EQUALS PERIODIC LINE POINTS)
2951      I1=I1-1
          GOTO 295
          ETA(IEXIT+1)=2.*ETA(I1,4)
          X(IEXIT,1)=2.*X(IEXIT,4)
          X(IEXIT,2)=1.5*X(IEXIT,4)
          Y(IEXIT,1)=1.2*Y(IEXIT,4)
          Y(IEXIT,2)=1.5*Y(IEXIT,4)
          X(NOZ,1)=2.*X(NOZ,4)
          X(NOZ,2)=1.5*X(NOZ,4)
          Y(NOZ,1)=2.*Y(NOZ,4)
          Y(NOZ,2)=1.5*Y(NOZ,4)

```

```

Y(NOZ,3)=1.2*Y(NOZ,4)
MP3=M+3
C      MOVE ALL DATA UP ONE I LOCATION TO MAKE ROOM FOR A REPEATED RADIA
C      TING LINE
C      NEXT TO DOWNSTREAM INFINITY LINE .
DO 2103 J1=1,MP3
  I1=IEXIT
  IF(I1.LT.1)GOTO21071
  X(I1+1,J1)=X(I1,J1)
  Y(I1+1,J1)=Y(I1,J1)
  DUMBX(I1,J1)=0.
  DUMBY(I1,J1)=0.
  I1=I1-1
  GOTO2107
21071  X(2,J1)=X(IEXIT+1,J1)
       Y(2,J1)=Y(IEXIT+1,J1)
       X(1,J1)=X(IEXIT,J1)
       Y(1,J1)=Y(IEXIT,J1)
       X(IEXIT+2,J1)=X(3,J1)
       Y(IEXIT+2,J1)=Y(3,J1)
2103
ETA(IEXIT+2)=ETA(3)+2.*ETA(1)
ETA(1)=-ETA(3)
C      UPDATE COUNTERS
NZZ=NZZ+1
IEXIT=IEXIT+1
NOZ=NOZ+1
M=M+3
ZETA(M+1)=ZETA(M)+ZETA(M)-ZETA(M-1)
I67=M-14+1
I67=MAX0(I67,1)
I671=I67+1
IF(.NOT.BUG2) RETURN
C      FOR NEW GEOMETRY GENERATIONS, CALCULATE SECOND DERIVATIVES ON BLA
C      DE SURFACE
C      USING THE FINAL GRID POINTS.

```

```

0S024970
0S024980
0S024990
0S025000
0S025010
0S025020
0S025030
0S025040
0S025050
0S025060
0S025070
0S025080
0S025090
0S025100
0S025110
0S025120
0S025130
0S025140
0S025150
0S025160
0S025170
0S025180
0S025190
0S025200
0S025210
0S025220
0S025230
0S025240
0S025250
0S025260
0S025270
0S025280
0S025290
0S025300
0S025310
0S025320
0S025330
0S025340
0S025350
0S025360
0S025370
0S025380
0S025390
0S025400
0S025410
0S025420
0S025430
0S025440
0S025450
0S025460
0S025470
0S025480
0S025490
0S025500
0S025510
0S025520
0S025530
0S025540
0S025550
0S025560
0S025570
0S025580
0S025590
0S025600
0S025610
0S025620
0S025630
0S025640
0S025650
0S025660
0S025670
0S025680
0S025690
0S025700
0S025710
0S025720
0S025730
0S025740
0S025750
0S025760
0S025770
0S025780
0S025790
0S025800
0S025810
0S025820
0S025830
0S025840
0S025850
0S025860
0S025870
0S025880
0S025890
0S025900
0S025910
0S025920
0S025930
0S025940
0S025950
0S025960
0S025970
0S025980
0S025990
0S026000
0S026010
0S026020
0S026030
0S026040
0S026050
0S026060
0S026070
0S026080
0S026090
0S026100
0S026110
0S026120
0S026130
0S026140
0S026150
0S026160
0S026170
0S026180
0S026190
0S026200
0S026210
0S026220
0S026230
0S026240
0S026250
0S026260
0S026270
0S026280
0S026290
0S026300
0S026310
0S026320
0S026330
0S026340
0S026350
0S026360
0S026370
0S026380
0S026390
0S026400
0S026410
0S026420
0S026430
0S026440
0S026450
0S026460
0S026470
0S026480
0S026490
0S026500
0S026510
0S026520
0S026530
0S026540
0S026550
0S026560
0S026570
0S026580
0S026590
0S026600
0S026610
0S026620
0S026630
0S026640
0S026650
0S026660
0S026670
0S026680
0S026690
0S026700
0S026710
0S026720
0S026730
0S026740
0S026750
0S026760
0S026770
0S026780
0S026790
0S026800
0S026810
0S026820
0S026830
0S026840
0S026850
0S026860
0S026870
0S026880
0S026890
0S026900
0S026910
0S026920
0S026930
0S026940
0S026950
0S026960
0S026970
0S026980
0S026990
0S027000
0S027010
0S027020
0S027030
0S027040
0S027050
0S027060
0S027070
0S027080
0S027090
0S027100
0S027110
0S027120
0S027130
0S027140
0S027150
0S027160
0S027170
0S027180
0S027190
0S027200
0S027210
0S027220
0S027230
0S027240
0S027250
0S027260
0S027270
0S027280
0S027290
0S027300
0S027310
0S027320
0S027330
0S027340
0S027350
0S027360
0S027370
0S027380
0S027390
0S027400
0S027410
0S027420
0S027430
0S027440
0S027450
0S027460
0S027470
0S027480
0S027490
0S027500
0S027510
0S027520
0S027530
0S027540
0S027550
0S027560
0S027570
0S027580
0S027590
0S027600
0S027610
0S027620
0S027630
0S027640
0S027650
0S027660
0S027670
0S027680
0S027690
0S027700
0S027710
0S027720
0S027730
0S027740
0S027750
0S027760
0S027770
0S027780
0S027790
0S027800
0S027810
0S027820
0S027830
0S027840
0S027850
0S027860
0S027870
0S027880
0S027890
0S027900
0S027910
0S027920
0S027930
0S027940
0S027950
0S027960
0S027970
0S027980
0S027990
0S028000
0S028010
0S028020
0S028030
0S028040
0S028050
0S028060
0S028070
0S028080
0S028090
0S028100
0S028110
0S028120
0S028130
0S028140
0S028150
0S028160
0S028170
0S028180
0S028190
0S028200
0S028210
0S028220
0S028230
0S028240
0S028250
0S028260
0S028270
0S028280
0S028290
0S028300
0S028310
0S028320
0S028330
0S028340
0S028350
0S028360
0S028370
0S028380
0S028390
0S028400
0S028410
0S028420
0S028430
0S028440
0S028450
0S028460
0S028470
0S028480
0S028490
0S028500
0S028510
0S028520
0S028530
0S028540
0S028550
0S028560
0S028570
0S028580
0S028590
0S028600
0S028610
0S028620
0S028630
0S028640
0S028650
0S028660
0S028670
0S028680
0S028690
0S028700
0S028710
0S028720
0S028730
0S028740
0S028750
0S028760
0S028770
0S028780
0S028790
0S028800
0S028810
0S028820
0S028830
0S028840
0S028850
0S028860
0S028870
0S028880
0S028890
0S028900
0S028910
0S028920
0S028930
0S028940
0S028950
0S028960
0S028970
0S028980
0S028990
0S029000
0S029010
0S029020
0S029030
0S029040
0S029050
0S029060
0S029070
0S029080
0S029090
0S029100
0S029110
0S029120
0S029130
0S029140
0S029150
0S029160
0S029170
0S029180
0S029190
0S029200
0S029210
0S029220
0S029230
0S029240
0S029250
0S029260
0S029270
0S029280
0S029290
0S029300
0S029310
0S029320
0S029330
0S029340
0S029350
0S029360
0S029370
0S029380
0S029390
0S029400
0S029410
0S029420
0S029430
0S029440
0S029450
0S029460
0S029470
0S029480
0S029490
0S029500
0S02951
```

```

C      DO (ADDITION OF BLADE SURFACE ON CLEAN FLATPLATE MESH)
C      RETURN
C
C      DIMENSION ET(100),ZET(100),XBOD(200),YBOD(200)
C      REAL K,LNK,M1,LAM,LAMDAO,M
C      INTEGER ZNO,ENO,ZN2M1,ZN2,END,ENOPLS,CAMBER,ZNP1,ENOPI
C      DOUBLE PRECISION MMDELK
C      LOGICAL PRECISION CAPK,CAPKP,DM,DM1
C      LOGICAL DOSURF,SMOOTH
C      COMMON/ENTIRE/ X(100,30),Y(100,30)
C      COMMON/GEOM/ZNO,END,RLE,RTE,A,CHORB,XUPS,XDNS,STAG,CC,CAPK
C      1,CAPK,PI
C      1,CAPK,M1,KM,END,M,DOSURF,R,SOLID,ENOPLS
C      1,STABAC,SLP1,SLP2,SLP3,SLP4,SMOOTH,DUMBO(5),CAMBER
C      XUPS=-1.E+01
C      XDNS=1.E+01
C      PI=3.14159265
C      DELTA=0.
C      ZNP1=ZNO+1
C      LAMDAO=STAG
C      555 FORMAT(1H,2E12.5)
C      ENOPLS=ENO+ENO/CAMBER
C      CHORD=CHORB-RLE/2.-RTE/2.
C      XDNS=-25.*CHORD
C      XUPS=-25.*CHORD
C      SOLID=CHORB/S
C      SOLFLT=CHORD/S
C      SOLID0=SOLFLT
C      IF(STAG.EQ.0.)GOTO3002
C
C      DO (INTERMEDIATE TRANSFORM SETUP, I.E. ZERO STAG, S=2PI)
C
C      PROCEDURE (INTERMEDIATE TRANSFORM SETUP, I.E. ZERO STAG, S=2PI)
C      ASTAG=ABS(STAG)
C      CALL ALTERP(SOLFLT,ASTAG,ALFA,R)
C      CHORD=ALOG((R*R+2.*R+1.)/(R*R-2.*R+1.))*2.
C      SOLID0=CHORD/PI/2.
C      S=2.*PI
C      LAMDAO=0.
C      END
C
C      3002 CONTINUE

```

```

C      DO (CALCULATION OF GEOMETRY CONSTANTS)
C      PROCEDURE (CALCULATION OF GEOMETRY CONSTANTS)
A= COS(LAMDAO)/2./SOLIDO
K= EXP(-PI/2./A)
LNK = -ALOG(K)
M=K*K
DM=M
DM1=1,DO-DM
M1=DM1
CAPK=MMDELK(1,DM,IER)
CAPKP=MMDELK(1,DM1,IER)
END
C
C      DO (KNOWN GRID COORDINATES, ET,ZET)
C      PROCEDURE (KNOWN GRID COORDINATES, ET,ZET)
DELZ=CAPKP/(ZNO-1)
DELE=CAPK*STABAC/(ENO-1)
ZN2M1=2*ZNO-1
ZET(1)=+CAPKP
DO 217 NZ=2,ZN2M1
ZET(NZ)=ZET(NZ-1)-DELZ
ET(1)=0.
ZN2 = 2*ZNO
DO 221 NE=2,ENOPLS
ET(NE)=ET(NE-1)+DELE
ET(ENO)=STABAC*CAPK
ZET(ZNO)=0.
IF(SLP1.GE.0.)GOTO205
217
221
C      IF(SLP1.LT.0.) DO (CONCENTRATION OF GRID LINES IN KEY AREAS)
C      PROCEDURE (CONCENTRATION OF GRID LINES IN KEY AREAS)
XTES=-1.E19
ITS=1
SLP1=-SLP1
C      IF SLP2.LT.0 SETUP THE PROPER STAGGERED FLATPLATE MESH TO FIND WH
C      ICH LINE INTERSECTS T. E.
C      IF(SLP2.GE.0.;OR.S.NE.2.*PI) GOTO226
DELTA=ABS(SLP2)

```

Q5025930
 Q5025940
 Q5025950
 Q5025960
 Q5025970
 Q5025980
 Q5025990
 Q5026000
 Q5026010
 Q5026020
 Q5026030
 Q5026040
 Q5026050
 Q5026060
 Q5026070
 Q5026080
 Q5026090
 Q5026100
 Q5026110
 Q5026120
 Q5026130
 Q5026140
 Q5026150
 Q5026160
 Q5026170
 Q5026180
 Q5026190
 Q5026200
 Q5026210
 Q5026220
 Q5026230
 Q5026240
 Q5026250
 Q5026260
 Q5026270
 Q5026280
 Q5026290
 Q5026300
 Q5026310
 Q5026320
 Q5026330
 Q5026340
 Q5026350
 Q5026360
 Q5026370
 Q5026380
 Q5026390
 Q5026400

```

C C DO (FLATPLATE-ZERO STAGGER TRANSFM, E,Z-X,Y UPPER SURFACE)
C C PROCEDURE (FLATPLATE-ZERO STAGGER TRANSFM, E,Z-X,Y UPPER SURFACE)
C C DO 2060 I=1,ZNO
C C DO 2060 J=1,ENOPLS
C C IF(I.EQ.1.OR.I.EQ.ZNO).AND.(J.EQ.1) GOTO2060
C C ETER=ET(J)
C C
C C DO (YOUR BASIC TRANSFORMATION )
C C PROCEDURE (YOUR BASIC TRANSFORMATION )
C C ZED=ZET(I)
C C IF(ETER.GT.CAPK) ZED=-ZED
C C IF(ETER.GT.CAPK) ETER=2.*CAPK-ETER
C C IF(I.EQ.ZNO) ZED=0.
C C ALLFNC=FUDS(ETER, ZED,M1)
C C LAM = BADZAC(M1)
C C TH = ONOW(M1)
C C X(I,J)= CHORD*A/PI*(LNK-2.*ALOG(ALLFNC/LAM))
C C IF(J.GT.END.AND.I.EQ.ZN2M1)X(I,J)=-CHORD-X(I,J)
C C IF(J.GT.END.AND.I.EQ.ZNO)X(I,J)=CHORD-X(I,J)
C C Y(I,J)= 2.*A*CHORD/PI*TH
C C IF(X(I,J).GT.XDNS)X(I,J)=XDNS
C C IF(X(I,J).LT.XUPS)X(I,J)=XUPS
C C END
C 2060 CONTINUE
C C END
C C DO (FLATPLATE-ZERO STAGGER TRANSFM, E,Z-X,Y LOWER SURFACE)
C C PROCEDURE (FLATPLATE-ZERO STAGGER TRANSFM, E,Z-X,Y LOWER SURFACE)
C C DO 2071 I=ZN1,ZN2M1
C C IDIF=I-ZNO
C C X(I,I)=X(ZNO-IDIF,I)-S*SIN(LAMDAO)
C C Y(I,I)= -A*CHORD
C C DO 2071 J=1,ENOPLS
C C IF(I.EQ.ZN2M1.AND.(J.EQ.1) GOTO2071
C C ETER=ET(J)
C C
C C DO (YOUR BASIC TRANSFORMATION )
C C PROCEDURE (YOUR BASIC TRANSFORMATION )
C C ZED=ZET(I)
C C IF(ETER.GT.CAPK) ZED=-ZED
C C IF(ETER.GT.CAPK) ETER=2.*CAPK-ETER
C C IF(I.EQ.ZNO) ZED=0.
C C ALLFNC=FUDS(ETER, ZED,M1)
C C LAM = BADZAC(M1)
C C TH = ONOW(M1)
C C X(I,J)= CHORD*A/PI*(LNK-2.*ALOG(ALLFNC/LAM))

```

```

C
2071
C
4445
234
226
C
C
C
C
237

IF(J.GT.ENO.AND.I.EQ.ZN2M1)X(I,J)=~CHORD-X(I,J)
IF(J.GT.ENO.AND.I.EQ.ZNO)X(I,J)=CHORD-X(I,J)
Y(I,J)=2.*A*CHORD/PI*TH
IF(X(I,J).GT.XDONS)X(I,J)=XDONS
IF(X(I,J).LT.XUPS)X(I,J)=XUPS
END
CONTINUE
END
CALL TOSTAG(S,ZET,CHORD)
DO 234 I3=1,ZN2M1
IF(X(I3,ENO).LE.XIES) GOTO234
WRITE(6,4445) XIES,I3,X(I3,ENO)
FORMAT(E13.5,I6,E13.5)
XTFS=X(I3,ENO)
ITS=I3
CONTINUE
EF=ZET(ITS)/CAPKP
IF(SLP1.EQ.10.) EF=1.-EF
SLP1=(1.-DELTA/EF/EF*(.6667-.2/EF/EF))/
1(1.-1./EF/EF*(.6667-.2/EF/EF**4))-DELTA/2.5/EF**4
SLP2=3.-SLP1*(2.-1./2.5/EF**4)
CONTINUE
CONCENTRATE THE RADIATING LINES (COMPUTATIONAL COORD ZET)
DO (CONCENTRATION POLYNOMIAL)
PROCEDURE (CONCENTRATION POLYNOMIAL)
CON1=2.5*(1.-SLP1)/CAPKP**2
CON2=(SLP2-SLP1)/(2.*CAPKP**2)
CON1=CON1-CON2
CON3=(SLP2-SLP1)/(2.*CAPKP**4)
CON4=1.5*(1.-SLP1)/(CAPKP**4)
CON3=CON3-CON4
END
DO 237 I3=1,ZN2M1
CAPFAC=0
ZETER1=ZET(I3)
ZET(I3)=SLP1*ZETER1+CON1*ZETER1**3+CON3*ZETER1**5-CAPFAC
IF(DELTA.EQ.0.) GOTO241
EF=1.-EF
SLP1=(1.-DELTA/EF/EF*(.6667-.2/EF/EF))/
1(1.-1./EF/EF*(.6667-.2/EF/EF**4))-DELTA/2.5/EF**4
SLP2=3.-SLP1*(2.-1./2.5/EF**4)

```

```

C
C
DO (CONCENTRATION POLYNOMIAL)
PROCEDURE (CONCENTRATION POLYNOMIAL)
CON1=2.5*(1.-SLP1)/CAPK**2
CON2=(SLP2-SLP1)/(2.*CAPK**2)
CON1=CON1-CON2
CON3=(SLP2-SLP1)/(2.*CAPK**4)
CON4=1.5*(1.-SLP1)/(CAPK**4)
CON3=CON3-CON4
END
C
CONCENTRATE THE SURFACE CONTOURS (COMPUTATIONAL COORD ET)
DO 244 I3=1,ZN2M1
CAPFAC=0
ZETER1=ZET(I3)
ZET(I3)=SLP1*ZETER1+CON1*ZETER1**3+CON3*ZETER1**5-CAPFAC
CON5=2.5*(1.-SLP3)/CAPK**2
CON6=(SLP4-SLP3)/(2.*CAPK**2)
CON5=CON5-CON6
CON7=(SLP4-SLP3)/2/CAPK**4
CON8=1.5*(1.-SLP3)/CAPK**4
CON7=CON7-CON8
DO 248 J3=1,ENOPLS
IF(J3.GT.ENO)ET(J3)=2.*CAPK-ET(J3)
ET(J3)=SLP3*ET(J3)+CON5*ET(J3)**3+CON7*ET(J3)**5
IF(J3.GT.ENO)ET(J3)=2.*CAPK-ET(J3)
CONTINUE
SLP1=-SLP1
IF(DELTA.NE.0.)SLP2=-DELTA
FORMAT(6E13.5)
END
C
END OF CONCENTRATION PROCEDURE
C
205
X(1,1)=XUPS
X(ZNO,1)=XONS
X(ZN2M1,1)=XUPS
ZNZNOM1=ZNO-1
C
DO (FLATPLATE-ZERO STAGGER TRANSFM. E,Z-X,Y UPPER SURFACE)

```

```

C      PROCEDURE (FLATPLATE-ZERO STAGGER TRANSFM. E,Z-X,Y UPPER SURFACE)
DO 260 I=1,ZNO
DO 260 J=1,ENOPLS
IF(I.EQ.1.OR.I.EQ.ZNO).AND.J.EQ.1) GOTO260
ETER=ET(J)
QSO27850
QSO27860
QSO27870
QSO27880
QSO27890
QSO27900
QSO27910
QSO27920
QSO27930
QSO27940
QSO27950
QSO27960
QSO27970
QSO27980
QSO27990
QSO28000
QSO28010
QSO28020
QSO28030
QSO28040
QSO28050
QSO28060
QSO28070
QSO28080
QSO28090
QSO28100
QSO28110
QSO28120
QSO28130
QSO28140
QSO28150
QSO28160
QSO28170
QSO28180
QSO28190
QSO28200
QSO28210
QSO28220
QSO28230
QSO28240
QSO28250
QSO28260
QSO28270
QSO28280
QSO28290
QSO28300
QSO28310
QSO28320

C      DO (YOUR BASIC TRANSFORMATION )
C      PROCEDURE (YOUR BASIC TRANSFORMATION )
ZED=ZET(I)
IF(ETER.GT.CAPK) ZED=-ZED
IF(ETER.GT.CAPK) ETER=2.*CAPK-ETER
IF(I.EQ.ZNO)ZED=0.
ALLFNC=FUDS(ETER,ZED,M1)
LAM = BADZAC(M1)
TH = OWOW(M1)
X(I,J)= CHORD*A/PI*(LNK-2.*ALOG(ALLFNC/LAM))
IF(J.GT.ENO.AND.I.EQ.ZN2M1)X(I,J)=-CHORD-X(I,J)
IF(J.GT.ENO.AND.I.EQ.ZNO)X(I,J)=CHORD-X(I,J)
Y(I,J)= 2.*A*CHORD/PI*TH
IF(X(I,J).GT.XDNS)X(I,J)=XDNS
IF(X(I,J).LT.XUPS)X(I,J)=XUPS
END
C
260 CONTINUE
C      END
ZNP1=ZNO+1
FK=1./SQRT(K)
NZN2M2=ZN2M1-1

C      DO (FLATPLATE-ZERO STAGGER TRANSFM. E,Z-X,Y LOWER SURFACE)
C      PROCEDURE (FLATPLATE-ZERO STAGGER TRANSFM. E,Z-X,Y LOWER SURFACE)
DO 271 I=ZNP1,ZN2M1
IDIF=1-ZNO
X(I,1)=X(ZNO-IDIF,1)-S*SIN(LAMDAO)
Y(I,1)=-A*CHORD
DO 271 J=1,ENOPLS
IF(I.EQ.ZN2M1.AND.J.EQ.1) GOTO271
ETER=ET(J)
QSO27850
QSO27860
QSO27870
QSO27880
QSO27890
QSO27900
QSO27910
QSO27920
QSO27930
QSO27940
QSO27950
QSO27960
QSO27970
QSO27980
QSO27990
QSO28000
QSO28010
QSO28020
QSO28030
QSO28040
QSO28050
QSO28060
QSO28070
QSO28080
QSO28090
QSO28100
QSO28110
QSO28120
QSO28130
QSO28140
QSO28150
QSO28160
QSO28170
QSO28180
QSO28190
QSO28200
QSO28210
QSO28220
QSO28230
QSO28240
QSO28250
QSO28260
QSO28270
QSO28280
QSO28290
QSO28300
QSO28310
QSO28320

C      DO (YOUR BASIC TRANSFORMATION )
C      PROCEDURE (YOUR BASIC TRANSFORMATION )
ZED=ZET(I)
IF(ETER.GT.CAPK) ZED=-ZED
IF(ETER.GT.CAPK) ETER=2.*CAPK-ETER

```

Q5028330
Q5028340
Q5028350
Q5028360
Q5028370
Q5028380
Q5028390
Q5028400
Q5028410
Q5028420
Q5028430
Q5028440
Q5028450
Q5028460
Q5028470
Q5028480
Q5028490
Q5028500
Q5028510
Q5028520
Q5028530
Q5028540
Q5028550
Q5028560
Q5028570
Q5028580
Q5028590
Q5028600
Q5028610
Q5028620
Q5028630
Q5028640
Q5028650
Q5028660
Q5028670
Q5028680
Q5028690
Q5028700
Q5028710
Q5028720
Q5028730
Q5028740
Q5028750
Q5028760
Q5028770
Q5028780
Q5028790
Q5028800

```

IF(I.EQ.ZNO)ZED=0.
ALLFNC=FUDS(ETER, ZED,M1)
LAM = BADZAC(M1)
TH = OWOW(M1)
X(I,J)= CHORD*A/PI*(LNK-2.*ALOG(ALLFNC/LAM))
IF(J.GT.ENO.AND.I.EQ.ZN2M1)X(I,J)=-CHORD-X(I,J)
IF(J.GT.ENO.AND.I.EQ.ZNO)X(I,J)=CHORD-X(I,J)
Y(I,J)= 2.*A*CHORD/PI*TH
IF(X(I,J).GT.XDONS)X(I,J)=XDONS
IF(X(I,J).LT.XUPS)X(I,J)=XUPS
END
C
271 CONTINUE
C END
C DO (SPECIAL POINTS)
C
C PROCEDURE (SPECIAL POINTS)
DO 278 J=1, ENOPLS
X(I,J)=X(ZN2M1,J)
Y(ZN2M1,J)=0.
Y(ZNO,J)=0.
Y(I,J)=0.
END
278 C
IF(STAG.EQ.0.) GOTO 209
C DO (CIRCLE TRANSFM. FROM ZERO STAGGER TO INPUT VALUES)
C
C PROCEDURE (CIRCLE TRANSFM. FROM ZERO STAGGER TO INPUT VALUES)
CALL TOSTAG(SIZE,CHORD)
CHORD=CHORB-R/E/2.-RLE/2.
LAMDAO=STAG
SOLIDO=SOLFLT
S=CHORD/SOLFLT
OLDK=CAPK
OLDKP=CAPKP
C
C DO (CALCULATION OF GEOMETRY CONSTANTS)
C
C PROCEDURE (CALCULATION OF GEOMETRY CONSTANTS)
A= COS(LAMDAO)/2./SOLIDO
K= EXP(-PI/2./A)
LNK = -ALOG(K)
M=K*K
DM=M

```

```

C
283
287
C
209
C
C
291
C
C
C
295
C
C
DO (ADDITION OF BLADE SURFACE ON CLEAN MESH)
IF (DOSURF) CALL SURFU(ET,ZET,XBOD,YBOD,EF)

```

```

QSD28810
QSD28820
QSD28830
QSD28840
QSD28850
QSD28860
QSD28870
QSD28880
QSD28890
QSD28900
QSD28910
QSD28920
QSD28930
QSD28940
QSD28950
QSD28960
QSD28970
QSD28980
QSD28990
QSD29000
QSD29010
QSD29020
QSD29030
QSD29040
QSD29050
QSD29060
QSD29070
QSD29080
QSD29090
QSD29100
QSD29110
QSD29120
QSD29130
QSD29140
QSD29150
QSD29160
QSD29170
QSD29180
QSD29190
QSD29200
QSD29210
QSD29220
QSD29230
QSD29240
QSD29250
QSD29260
QSD29270
QSD29280

```

7777 FORMAT(14E9.3)

RETURN

```

C-----C
END DOUBLE PRECISION FUNCTION MMDELK(ILPT,ARG,IER)
CALCS COMPLETE ELLIPTIC INTEGRAL OF FIRST KIND FOR MODULUS SQUARED
      = ARG
DOUBLE PRECISION ARG,PI,A,B,C,A0,B0
DATA PI/3.141592653589793D0/
IER = 130
IF(ILPT.NE.1) STOP 105
IF(ARG.LT.0.D0.OR.ARG.GE.1.D0) GOTO40
IER = 0
A = 1.C0
B = DSQRT(1.D0 - ARG)
DO 10 I = 1,15
C = .5D0 * (A - B)
IF(DABS(C).LT.1.D-12) GOTO15
A0 = A
B0 = B
A = .5D0 * (A0 + B0)
B = DSQRT(A0 * B0)
CONTINUE
IF(I.GT.15) STOP 110
MMDELK = .5D0 * PI / A
RETURN
END
SUBROUTINE SURFUP(ET,ZET,XBOD,YBOD,EF)
10
15
40
```


QSO30250
QSO30260
QSO30270
QSO30280
QSO30290
QSO30300
QSO30310
QSO30320
QSO30330
QSO30340
QSO30350
QSO30360
QSO30370
QSO30380
QSO30390
QSO30400
QSO30410
QSO30420
QSO30430
QSO30440
QSO30450
QSO30460
QSO30470
QSO30480
QSO30490
QSO30500
QSO30510
QSO30520
QSO30530
QSO30540
QSO30550
QSO30560
QSO30570
QSO30580
QSO30590
QSO30600
QSO30610
QSO30620
QSO30630
QSO30640
QSO30650
QSO30660
QSO30670
QSO30680
QSO30690
QSO30700
QSO30710
QSO30720

```
DOUBLE PRECISION CAPK,CAPKP
COMMON/GEOM/ZNO,ENO,RLE,RTE,A,CHORD,XUPS,XDNS,LAMDAO,CC,CAPK
1,CAPKP,PI
1,LNK,M1,KM,END,M,DOSURF,AARR,SOL,ENOPLS
1,SDUM(5),SMOOTH,CHOP,THET,THETL,TDUM,LEONLY
COMMON/ENTIRE/XX(100,30),YY(100,30)
COMMON/SPGENC/DA,DSMAX,RMAX,THIN,DB,TMIN,DSEND
COMMON/FUNCDS/S,C,D,SN,CN,DN
COMMON/MAN/DELSMX,PIO2,DELS1,IHUB
DIMENSION XBDD(200),YBDD(200),SU(400),SD(400),ET(100),ZET(100),
1,ISAVE(25),SEDT(400),SYETA(400),SDXETA(400),SDYETA(400)
1,INTEGER ZNO,ENO,END,ZN2M1,ENOPLS,ENDM1,ENPOLS,ENDMM1,ENMI
LOGICAL DUM,RZTRP,IZTRP,THIN,DONE,SMOOTH,DOSURF,LEONLY
COMPLEX FZTRP,RELAXD,BOTTOM,FZ,Z(400),XOETA(400),YOETA(400)
1,EDGL(400),EDGT(400),XYETA(400),YYETA(400)
1,REAL M,M1,LNK,LAMDAO
NAMELIST/SPGENS/DA,DSMAX,RMAX,THIN,DB,TMIN,DSEND
NAMELIST/TESY/I,IS,ETA,Y,X,SPX,SPY,SPE,ITS1,DF1DN,DF2DN,DF3DX,
1F1,F2,F3,DELN,DELX,DELY,FZ,SP
NAMELIST/TESX/XYETA,XYETA,SYETA,SYETA
DATA Z/400*(1.E20,0.)/
DATA Z2/400*(1.E20,0.)/
DATA EDGT,EDGL,XOETA,YOETA,XYETA,YYETA/2400*(1.E20,0.)/

ET(ENO)=CAPK
DELSMX=.06667
PIO2=1.57079
THET=THETL*PI/180.
THETL=THETL*PI/180.
ISA=1
PLATE=(CHORD-RLE/2.-RTE/2.)/2.
```

C DO (BODY COORDINATE LOADING)

C PROCEDURE (BODY COORDINATE LOADING)

IF(.NOT.SMOOTH) GOT0222

```

IF(LEONLY) GOTO1008
KMH=(KM+1)/2
IF(MOD(KM,2).NE.0)KMH=KMH-1
DO 234 I1=1,KMH
I5=I1+KMH
Z(I1)=CMPLX(XBOD(I5),YBOD(I5))
IF(MOD(KM,2).NE.0)
1 Z(KMH+1)=CMPLX(XBOD(I5+1),YBOD(I5+1))
IF(MOD(KM,2).NE.0)KMH=KMH+1
KWOW=KMH+1+END-KM-2-(KM+1)/2+KM/2
KMH1=KMH+1
DO 240 I1=KMH1,KWOW
I5=END-I1+KMH
Z(I1)=CMPLX(XBOD(I5),YBOD(I5))
KWOWP1=KWOW+1
ENDMM1=END-1
DO 244 I1=KWOWP1,ENDMM1
I5=I1-END+(KM-1)/2+3-MOD(KM,2)
Z(I1)=CMPLX(XBOD(I5),YBOD(I5))
GOTO225
1008 KMH=END-KM
DO 226 I1=1,KMH
I5=END-I1+1
Z(I1)=CMPLX(XBOD(I5),YBOD(I5))
KWOW=END-1
KMH1=KMH+1
DO 230 I1=KMH1,KWOW
I5=I1-KMH+1
Z(I1)=CMPLX(XBOD(I5),YBOD(I5))
FORMAT(6E13.5)
ENDMM1=END-1
WRITE(6,1234) (Z(I1),I1=1,ENDMM1),
1(XBOD(I1),YBOD(I1),I1=1,END)
FORMAT(10E13.6)
DA=.17
WRITE(6,1235)
FORMAT(55H)THE FOLLOWING IS A PRINT OF THE Z MATRIX FROM SURFUP)
WRITE(6,*12
CALL XYCALC(1,ENDMM1,Z,SU,SD)
WRITE(6,1236)
FORMAT(69H)THE FOLLOWING IS A PRINT OF Z,SU,SD MATRICES AFTER TH
+E CALL XYCALC)
WRITE(6,*12
WRITE(6,*1SU
WRITE(6,*1SD

```

```

END=ENDM1
ITS=0.
XTS=0.
DO 248 I1=1,END
IF(SU(I1).GT.XTS) ITS=I1
IF(SU(I1).GT.XTS) XTS=SU(I1)
CONTINUE
KMP=ITS
ITS=0
XTS=100
DO 254 I1=1,END
IF(SU(I1).LT.XTS) IIS=I1
IF(SU(I1).LT.XTS) XTS=SU(I1)
CONTINUE
KIP=ITS
KM=END-(KIP-KMP)
KMI=KM+1
END=END+1
DO 260 I1=1,KM
I5=KIP+I1-1
IF(I5.GE.END-1) I5=I1-END+KIP+1
X800(I1)=SU(I5)
Y800(I1)=SD(I5)
FORMA(I1)=4E13.5,317)
260
4448
C
C
LOAD SMOOTHED DATA AROUND L.E. REGION INTO EDGL FOR USE IN *EDGE
REGION* TECHNIQUE
DO 264 I1=KMI,END
I5=END+KMP-I1
X800(I1)=SU(I5)
Y800(I1)=SD(I5)
DO 268 I1=KMP,KIP
IF(SU(I1).LT.-CHOP*.50*PLATE) GOTO271
CONTINUE
I1=KIP
I2L=0
DO 274 I3=1,END
IF(ABS(SU(I3)).LT.CHOP*.50*PLATE) GOTO277
YEE=SU(I3)*SIN(THETL)+SD(I3)*COS(THETL)
I2L=I2L+1
EDGL(I2L)=CMPLX(YEE,SU(I3))
NOTO=SGEN(SEDGL,EDGL,I2L)
274
277
C
C
IF (.NOT.LEONLY) LOAD SMOOTHED DATA AROUND T.E. INTO EDGT FOR USE
IN *EDGE REGION*
IF(LEONLY) GOTO280

```

```

282 DO 282 I1=1,KMP
285 IF(SU(I1).GT.CHOP*.50*PLATE) GOTO285
CONTINUE
I2T=0
DO 288 I3=1,K1P
IF(SU(I3).LT.CHOP*.50*PLATE) GOTO291
YEE=SU(I3)*SIN(THET)+SD(I3)*COS(THET)
I2T=I2T+1
EDGT(I2T)=CMPLX(YEE,SU(I3))
NOTO=SGEN(SEDGT,EDGT,I2T)

288
291 DO 298 I1=1,400
Z(I1)=1.E20
IF(I1.LE.400) SU(I1)=0.
IF(I1.LE.400) SD(I1)=0.
CONTINUE
DO 2100 I1=1,KM
Z(I1)=CMPLX(XBOD(I1),YBOD(I1))
NOTO=SGEN(SU,Z,KM)
KM1=KM+1
IF(NOTO.NE.0) WRITE(6,2) NOTO
FORMAT(10,SGEN FAILURE,16)
KENDO=END-KM
DO 2104 I1=KM1,END
I2=I1-KM
Z2(I2)=CMPLX(XBOD(I1),YBOD(I1))
NOTO=SGEN(SD,Z2,KENDO)
END OF BODY COORDINATE LOADING
ZN2M1=2*ZN0-1
NZ2M2=ZN2M1-1

DO 202 I=1,NZ2M2
ENPOL=ENOPLS

C DO (ITERATION SETUP)
C PROCEDURE (ITERATION SETUP)
ENPOL=ENOPLS
RELAXD=.FALSE.
FAILD=.FALSE.
EDGES=.FALSE.
BOTTOM=.FALSE.

```

```

QS031690
QS031700
QS031710
QS031720
QS031730
QS031740
QS031750
QS031760
QS031770
QS031780
QS031790
QS031800
QS031810
QS031820
QS031830
QS031840
QS031850
QS031860
QS031870
QS031880
QS031890
QS031900
QS031910
QS031920
QS031930
QS031940
QS031950
QS031960
QS031970
QS031980
QS031990
QS032000
QS032010
QS032020
QS032030
QS032040
QS032050
QS032060
QS032070
QS032080
QS032090
QS032100
QS032110
QS032120
QS032130
QS032140
QS032150
QS032160

```

Q5032170
Q5032180
Q5032190
Q5032200
Q5032210
Q5032220
Q5032230
Q5032240
Q5032250
Q5032260
Q5032270
Q5032280
Q5032290
Q5032300
Q5032310
Q5032320
Q5032330
Q5032340
Q5032350
Q5032360
Q5032370
Q5032380
Q5032390
Q5032400
Q5032410
Q5032420
Q5032430
Q5032440
Q5032450
Q5032460
Q5032470
Q5032480
Q5032490
Q5032500
Q5032510
Q5032520
Q5032530
Q5032540
Q5032550
Q5032560
Q5032570
Q5032580
Q5032590
Q5032600
Q5032610
Q5032620
Q5032630
Q5032640

```

C      DO (GRID-LINE COORDINATE LOADING)
C      PROCEDURE (GRID-LINE COORDINATE LOADING)
ITST=0
JTST=0
DO 20109 I1=1,400
  XOETA(I1)=1.E20
  YOETA(I1)=1.E20
  IF(I1.LE.400) SOXETA(I1)=0.
  IF(I1.LE.400) SDYETA(I1)=0.
20109 CONTINUE
DO 20115 I1=1,ENPOLS
  IF(ABS(X(I1,1)).GT.20.)XX(I1,1)=SIGN(7.*ABS(XX(I,2)),XX(I,1))
  XOETA(I1)=CMPLX(ET(I1),XX(I1,1))
  YOETA(I1)=CMPLX(ET(I1),YY(I1,1))
20115 NOTO=SGEN(SDYETA,YOETA,ENPOLS)
  IF(NOTO.NE.0) WRITE(6,2) NOTO
  NOTO=SGEN(SOXETA,XOETA,ENPOLS)
  IF(NOTO.NE.0) WRITE(6,2) NOTO
C
Y=YY(I,ENO)
IF(Y.EQ.0.)Y=SIGN(1.E-04,YY(I,ENO-1))
X=XX(I,ENO-1)
XOO=X
ETA=CAPK*(ENO-2)/(ENO-1)
IF(Y.LE.0.)AND(I.NE.1)BOTTOM=.TRUE.
IF(I.EQ.ZNO.AND.LAMDAO.GT.0.)BOTTOM=.TRUE.
DF1DN=(XX(I,ENO)-XX(I,ENO-1))/(ET(ENO-1)-ET(ENO))
DF2DN=(YY(I,ENO)-YY(I,ENO-1))/(ET(ENO-1)-ET(ENO))
DF3DX=0.
IF(DF2DN+DF1DN*DF3DX.EQ.0.)DF3DX=1.
DELN=0.
DELY=0.
SP=0.
END
C
C      DO (ITERATION FOR ETA LINE-BODY INTERSECTION)
C      PROCEDURE (ITERATION FOR ETA LINE-BODY INTERSECTION)
IF(ABS(XX(I,ENO)).GT.CHOP*PLATE.AND.
1((.NOT.LEONLY.OR.I.LE.ZNO/2.OR.I.GE.3*ZNO/2))) GOTO2119
C      2122 ITST=ITST+1
X=X+DELX

```

```

Y=Y+DELY
SPY=0.
SPX=0.
SPE=0.
ETA=ETA+DELN
IF(ETA.LT.0.*CAPK)ETA=ET(ENOPLS)
IF(ETA.GT.2.*CAPK)GOTO2124
IF(.NOT.(ET.EDGES))GOTO2124
IF(ETA.GT.90.*CAPK.AND.1TST.GT.20)ETA=.90*CAPK
DUM=RZTRP(SXVETA,XVETA,ETA,12S,SPX,12S)
GETA=AIMAG(FZTRP(SYVETA,YVETA,ETA,12Y,SPY,12Y))
FETA=AIMAG(FZTRP(SYVETA,YVETA,ETA,12Y,SPY,12Y))
GOTO2125
2124 DUM=RZTRP(SDXETA,XOETA,ETA,ENPOLS,SPX)
      GETA=AIMAG(FZTRP(SDXETA,XOETA,SPX,ENPOLS))
      DUM=RZTRP(SDYETA,YOETA,ETA,ENPOLS,SPY)
      FETA=AIMAG(FZTRP(SDYETA,YOETA,SPY,ENPOLS))
      F1=X-GETA
      F2=Y-FETA
      IF(.NOT.(BOTTOM))GOTO2126
      DUM=RZTRP(SD,ZZ,X,KENDQ,SP)
      FX=AIMAG(FZTRP(SD,ZZ,SP,KENDQ))
      GOTO2127
2126 DUM=RZTRP(SU,Z,X,KM,SP)
      FX=AIMAG(FZTRP(SU,Z,SP,KM))
      F3=Y-FX
2127 RELAXD=(ABS(F1)+ABS(F2)+ABS(F3)).LT..0001
      IF FIRST TRY IS UNSUCCESSFUL, GET SMOOTHED SLOPE DATA AND TRY AGAIN
      IN
      IF(1TST.LE.20.AND.ABS(DELN/ETA).LE.1000.)OR.EDGES)GOTO2128
      DO 1 ITERATION WITH BETTER SLOPES
      PROCEDURE(ITERATION WITH BETTER SLOPES)
      WRITE(6,5)I
      FORMAT(10ITERATION PROCEEDS WITH SMOOTHER SLOPE DATA FOR LINE ',I4',
1)DO 2139 I1=1,400
      XYETA(I1)=1.E20
      YVETA(I1)=1.E20
      IF(I1.LE.400)SXVETA(I1)=0.
      IF(I1.LE.400)SYVETA(I1)=0.
      CONTINUE
2139
      DO 1 ITERATION SETUP AGAIN)

```

QSO322650
 QSO322660
 QSO322670
 QSO322680
 QSO322690
 QSO322700
 QSO322710
 QSO322720
 QSO322730
 QSO322740
 QSO322750
 QSO322760
 QSO322770
 QSO322780
 QSO322790
 QSO322800
 QSO322810
 QSO322820
 QSO322830
 QSO322840
 QSO322850
 QSO322860
 QSO322870
 QSO322880
 QSO322890
 QSO322900
 QSO322910
 QSO322920
 QSO322930
 QSO322940
 QSO322950
 QSO322960
 QSO322970
 QSO322980
 QSO322990
 QSO330000
 QSO330010
 QSO330020
 QSO330030
 QSO330040
 QSO330050
 QSO330060
 QSO330070
 QSO330080
 QSO330090
 QSO330100
 QSO330110
 QSO330120

Q5033130
Q5033140
Q5033150
Q5033160
Q5033170
Q5033180
Q5033190
Q5033200
Q5033210
Q5033220
Q5033230
Q5033240
Q5033250
Q5033260
Q5033270
Q5033280
Q5033290
Q5033300
Q5033310
Q5033320
Q5033330
Q5033340
Q5033350
Q5033360
Q5033370
Q5033380
Q5033390
Q5033400
Q5033410
Q5033420
Q5033430
Q5033440
Q5033450
Q5033460
Q5033470
Q5033480
Q5033490
Q5033500
Q5033510
Q5033520
Q5033530
Q5033540
Q5033550
Q5033560
Q5033570
Q5033580
Q5033590
Q5033600

```

C      PROCEDURE (ITERATION SETUP)
      ENPOLS=ENOPLS
      RELAXO=.FALSE.
      FAILED=.FALSE.
      EDGES=.FALSE.
      BOTTOM=.FALSE.

C      DO (GRID-LINE COORDINATE LOADING AGAIN)
      PROCEDURE (GRID-LINE COORDINATE LOADING)
      ITST=0
      DO 2109 I1=1,400
      XOETA(I1)=1.E20
      YOETA(I1)=1.E20
      IF(I1.LE.400) SXETA(I1)=0.
      IF(I1.LE.400) SYETA(I1)=0.
      CONTINUE
      DO 2115 I1=1,ENPOLS
      IF(ABS(X(I1),GT.20.)XX(I1,1)=SIGN(7.*ABS(XX(I,2)),XX(I,1))
      XOETA(I1)=CMPLX(ET(I1),XX(I,1,1))
      YOETA(I1)=CMPLX(ET(I1),YY(I,1,1))
      NOTO=SGEN(SDYETA,YOETA,ENPOLS)
      IF(NOTO.NE.0) WRITE(6,2) NOTO
      NOTO=SGEN(SXETA,XOETA,ENPOLS)
      IF(NOTO.NE.0) WRITE(6,2) NOTO
      END

C      Y=YY(I,ENO)
      IF(Y.EQ.0.)Y=SIGN(1.E-04,YY(I,ENO-1))
      X=XX(I,ENO-1)
      X00=X
      ETA=CAPK*(ENO-2)/(ENO-1)
      IF(Y.LE.0.)AND(I.NE.1)BOTOM=.TRUE.
      IF(I.EQ.ZNO.AND.LAMDA.GT.0.)BOTTOM=.TRUE
      DF1DN=(XX(I,ENO)-XX(I,ENO-1))/(ET(ENO-1)-ET(ENO))
      DF2DN=(YY(I,ENO)-YY(I,ENO-1))/(ET(ENO-1)-ET(ENO))
      DF3DX=0.
      IF(DF2DN+DF1DN*DF3DX.EQ.0.)DF3DX=1.
      DELN=0.
      DELX=0.
      DELY=0.
      SP=0.
      END

C      EDGES=.TRUE.
      DO 2146 I6=1,ENO
      DO 2146 I6=1,ENOPLS
      XYETA(I6)=XOETA(I6)

```

```

2146 YYETA(I6)=YOETA(I6)
      DA=.02
      DEL$MX=.06667
      C 12S=ENOPLS
      C SMOOTH THE SLOPE DATA
      CALL XYCALC(I,I2S,XYETA,XYETA,SYETA)
2150 DO 2150 I6=1,I2S
      XYETA(I6)=CMPLX(SYETA(I6),SYETA(I6))
      DO 2154 I4=1,400
      SYETA(I4)=0.
      C 12Y=ENOPLS
      C 12Y=ENOPLS
      CALL XYCALC(I,I2Y,YYETA,XYETA,SYETA)
      DEL$MX=.06667
      DO 2158 I6=1,I2Y
      YYETA(I6)=CMPLX(SYETA(I6),SYETA(I6))
2158 NOTO=SGEN(XYETA,XYETA,I2S)
      NOTO=SGEN(SYETA,YYETA,I2Y)
      C WRITE(6,TEST)
      C END OF BETTER SLOPE ACQUISITION, EDGES NOW = .TRUE.
2128 IF(IITST.GT.50.OR.ABS(DELN/ETA).GT.1000.) FAILED=.TRUE.
      IF(RELAXD) GOT02129
      DELN = (F3-F2-DF3DX*F1)/(DF2DN+DF1DN*DF3DX)
      DELX = -F1-DF1DN*DELN
      DELY = -F3-DF3DX*DELX
      C DO (SEARCH FOR BODY SLOPE)
      C PROCEDURE (SEARCH FOR BODY SLOPE)
      K4=1
      K3=KM
      K2=1
      IF(.NOT.BOTTOM) GOT02162
      K2=KM+1
      K3=END
      K4=1
2162 DO 2164 K1=K2,K3,K4
      IF(XBOD(K1).GE.X+DELX) GOT02167
2164 CONTINUE
      K1=K3
2167 DF3DX=(YBOD(K1-1)-YBOD(K1))/
      C * (XBOD(K1)-XBOD(K1-1))
      C END
      C DO (SEARCH FOR GRID-LINE SLOPE)
      C PROCEDURE (SEARCH FOR GRID-LINE SLOPE)
      IF(.NOT.EDGES) GOT02170
      DO 2172 J=1,I2S
      IF(REAL(XYETA(J)).GE.ETA) GOT02175
      C CONTINUE
2172

```

QS0333610
 QS0333620
 QS0333630
 QS0333640
 QS0333650
 QS0333660
 QS0333670
 QS0333680
 QS0333690
 QS0333700
 QS0333710
 QS0333720
 QS0333730
 QS0333740
 QS0333750
 QS0333760
 QS0333770
 QS0333780
 QS0333790
 QS0333800
 QS0333810
 QS0333820
 QS0333830
 QS0333840
 QS0333850
 QS0333860
 QS0333870
 QS0333880
 QS0333890
 QS0333900
 QS0333910
 QS0333920
 QS0333930
 QS0333940
 QS0333950
 QS0333960
 QS0333970
 QS0333980
 QS0333990
 QS034000
 QS034010
 QS034020
 QS034030
 QS034040
 QS034050
 QS034060
 QS034070
 QS034080

```

2175 IF(J.EQ.I2S+1) J=I2S
      DO 2180 JY=1,I2Y
      IF(REAL(XYETA(JY)).GE.ETA) GOTO2183
      CONTINUE
2180 IF(JY.EQ.I2Y+1) JY=I2Y
      DF1DN=(AIMAG(XYETA(J)))-AIMAG(XYETA(J-1)))/
      1(REAL(XYETA(J-1))-REAL(XYETA(J)))
      DF2DN=(AIMAG(YYETA(JY))-AIMAG(YYETA(JY-1)))/
      1(REAL(YYETA(JY-1))-REAL(YYETA(JY)))
      GOTO2171
2170 DO 2188 J=1,ENOPLS
      IF(ET(J).GE.ETA) GOTO2191
      CONTINUE
2188 IF(J.EQ.ENOPLS+1) J=ENOPLS
2191 DF1DN=(XX(I,J)-XX(I,J-1))/(ET(J-1)-ET(J))
      DF2DN=(YY(I,J)-YY(I,J-1))/(ET(J-1)-ET(J))
      CONTINUE
2171 C
2129 IF(.NOT.(DF2DN+DF3DX.EQ.0..OR.
      1((I.EQ.1.OR.I.EQ.ZNO):AND.LAMDAO.EQ.0..OR.
      1(.NOT.SMOOTH.AND.ABS(XX(I,ENO)).GT..98*PLATE.AND.FAILED))) GOTO2121
      RELAXD=.TRUE.
      FAILD=.FALSE.
      Y=YY(I,ENO)
      X=X80D(KM)
      IF(I.EQ.1) X=X80D(1)
      IF(SMOOTH) GOTO2135
      Y=0.
      X=X80D(1)
      IF(I.GT.ZNO/2.AND.I.LT.3*ZNO/2)X=X80D(KM)
      NUM=IZTRP(SOXETA,X,ENO,SPE)
      ETA=REAL(FZTRP(SOXETA,XOETA,SPE,ENO))
2135
2121 IF(.NOT.(RELAXD.OR.FAILD))GOTO2122
      C
2119 END OF ITERATION FOR ETA LINE-BODY INTERSECTION
      IF(RELAXD.OR.ABS(XX(I,ENO)).LE.CHOP*PLATE.AND..NOT.FAILD) GOTO3004
      IF(FAILD)WRITE(6,3) I
      IF(.NOT.FAILD)WRITE(6,4) I,XX(I,ENO),PLATE
      FORMAT(10ITERATION FAILED FOR LINE ,I4)
      3 , PROCEEDING TO EDGE REGION TECHNIQUE.
      4 , FORMAT(10ITERATION NOT ATTEMPTED FOR LINE ,I4,
      1 , , BODY X TOO CLOSE TO LE OR TE: ',2E15.6,' PROCEED TO EDGE TECH.')
```

```

QSO34090
QSO34100
QSO34110
QSO34120
QSO34130
QSO34140
QSO34150
QSO34160
QSO34170
QSO34180
QSO34190
QSO34200
QSO34210
QSO34220
QSO34230
QSO34240
QSO34250
QSO34260
QSO34270
QSO34280
QSO34290
QSO34300
QSO34310
QSO34320
QSO34330
QSO34340
QSO34350
QSO34360
QSO34370
QSO34380
QSO34390
QSO34400
QSO34410
QSO34420
QSO34430
QSO34440
QSO34450
QSO34460
QSO34470
QSO34480
QSO34490
QSO34500
QSO34510
QSO34520
QSO34530
QSO34540
QSO34550
QSO34560
```

```

208 IF(ET(J).GE.ETA) GOTO211
211 CONTINUE
IF(J.EQ.ENOPLS+1) ETA=ET(ENOPLS)
IF(.NOT.RELAXD) GOTO202
XX(I,ENO)=X
YY(I,ENO)=Y
ET(I)=ETA
DO (ETA VALUE STRETCH)
PROCEDURE (ETA VALUE STRETCH)
ENM1=ENO-1
DO 20196 J=2,ENM1
ETANEW=ET(J)/CAPK*ETA
SPX=SDXETA(I)
SPY=SDYETA(I)
DUM=RZTRP(SDXETA,XOETA,ETANEW,ENPOL,S PX)
XX(I,J)=AIMAG(FZTRP(SDXETA,XOETA,SPX,ENPOL))
DUM=RZTRP(SDYETA,YOETA,ETANEW,ENPOL,SPY)
YY(I,J)=AIMAG(FZTRP(SDYETA,YOETA,SPY,ENPOL))
20196 ENDO
C
202 CONTINUE (EDGE REGIONS)
C DO (EDGE REGIONS)
C PROCEDURE (EDGE REGIONS)
C
C *EDGE REGION TECHNIQUE* = INTERPOLATE ON CURVE OF INTERSECTION POINTS
C DEFINED BY SUCCESSFUL NEIGHBORING LINES.
C
NOZ02=ZN0/2
IF(ISAVE(I).GE.NOZ02) GOTO2020
DO 2202 I3=I,ISA
IF(ISAVE(I3).GT.NOZ02) GOTO2205
CONTINUE
I3=ISA
I3=I3-1
ISAP=ISA
DO 2208 I4=1,I3
ISAVE(ISAP)=ISAVE(I4)
ISAP=ISAP+1
ISAM1=ISA-1
DO 2212 I4=1,ISAM1
I4I3=I4+I3
ISAVE(I4)=ISAVE(I4I3)
DO 2216 I1=1,400
XYETA(I1)=1.E20
IF(I1.GT.400) GOTO2216
SXETA(I1)=0.

```

```

2216 SYVETA(I1)=0.
      SU(I1)=0.
      SD(I1)=0.
      CONTINUE
      I2=0
      IS=1
      NZ2M11=ZN2M1-1
C      LOAD VALUES OF BODY Y AND ZET FROM ALL SUCCESSFUL LINE ITERATION
C      S INTO ARRAY XYETA AND SMOOTH IT.
      DO 2222 I1=NOZ02,NZ2M11
      IF(I1S.GT.I1SA) GOT02222
      IF(I1SAVE(I1S).EQ.I1) GOT02228
      I2=I2+1
      THE=XX(I1,I1,ENO)*((THE1L-THE1I)/(-1.6*CHOP*PLATE)
      IF(XX(I1,I1,ENO).LT.-CHOP*.80*PLATE)THE=THE1L
      IF(XX(I1,I1,ENO).GT.-CHOP*.80*PLATE)THE=THE1I
      YEF=Y(I1,I1,ENO)*COS(THI)+XX(I1,ENO)*SIN(THI)
      YEE=YEF*10.
      XYETA(I2)=CMPLX(ZET(I1),YEF)
      YVETA(I2)=CMPLX(ZET(I1),YEE)
      IF(I1SAVE(I1S).EQ.I1)IS=IS+1
      CONTINUE
      DO 2230 I1=1,NOZ02
      IF(I1SAVE(I1S).EQ.I1) GOT02234
      I2=I2+1
      ZED=ZET(I1,I1-2,*CAPKP
      THE=XX(I1,I1,ENO)*((THE1L-THE1I)/(-1.6*PLATE)
      IF(XX(I1,I1,ENO).LT.-CHOP*.80*PLATE)THE=THE1L
      IF(XX(I1,I1,ENO).GT.-CHOP*.80*PLATE)THE=THE1I
      YEF=Y(I1,I1,ENO)*COS(THI)+XX(I1,ENO)*SIN(THI)
      YEE=YEF*10.
      XYETA(I2)=CMPLX(ZED,YEF)
      YVETA(I2)=CMPLX(ZED,YEE)
      IF(I1SAVE(I1S).EQ.I1)IS=IS+1
      CONTINUE
      I2S=I2
      DELSMX=.5
      DA=.05
      CALL XYCALC(I1,I2S,XYETA,SU,SD)
      DELSMX=.06667
      DO 2236 I6=1,I2S
      XYETA(I6)=CMPLX(SU(I6),SD(I6))
      NOT0=SGEN(SXYETA,XYETA,I2S)
      NOT0=SGEN(SYVETA,YVETA,I2)
      WRITE(6,TEXS)
      SPT=0.
C
2228
2222
2234
2230
2236
C

```

```

SPL=0.
ISAM1=ISA-1
DO 2240 IS=1, ISAM1
SPX=0.
SPE=0.
TE=FALSE
I1=ISAVE(IS)
I=I1
IMM1=I-1
IF (IMM1.EQ.0) IMM1=ZN2M1-1
IF (I.GT.ZN2M1/4.AND.I.LT.3*ZN2M1/4) TE=.TRUE.

IF (TE) SPL=0.
ZED=ZET(I1)

C INTERPOLATE ON THE TABLE OF Y VALUES IN XYETA USING THE KNOWN VAL
C UE OF ZETA
C FOR EACH UNSUCCESSFUL LINE.

IF (.NOT. TE.AND.ZED.GT..85*CAPKP) ZED=ZED-2.*CAPKP
DUM=RZTRP(SXYETA,XYETA,ZED,I2S,SPX)
Y=AIMAG(FZTRP(SXYETA,XYETA,SPX,I2S))/10.

C AFTER Y IS FOUND, OBTAIN X FROM TABLE OF L.E. AND T.E. BODY SHAPE
C IN EDGL AND EDGT.

IF (.NOT. SMOOTH) GOTO2244
IF (.NOT. TE) GOTO2246
DUM=RZTRP(SEDGT,EDGT,Y,I2I,SP1)
X=AIMAG(FZTRP(SEDGT,EDGT,SP1,I2T))
IF (ABS(X-XX(IMM1,END)).LE.1.5/FLOAT(ZNO)) GOTO22461
DUM=RZTRP(SEDGT,EDGT,Y,I2I,SP1)
X=AIMAG(FZTRP(SEDGT,EDGT,SP1,I2T))
GOTO2245
DUM=RZTRP(SEDGL,EDGL,Y,I2L,SP1)
X=AIMAG(FZTRP(SEDGL,EDGL,SP1,I2L))
IF (ABS(X-XX(IMM1,END)).LE.1.5/FLOAT(ZNO)) GOTO22441
DUM=RZTRP(SEDGL,EDGL,Y,I2L,SP1)
X=AIMAG(FZTRP(SEDGL,EDGL,SP1,I2L))
GOTO2245
IF (Y.GT.0.OR.I.EQ.1) GOTO2252
DUM=IZTRP(SD,Z2,Y,KENDO,SP)

```

QSO35530
 QSO35540
 QSO35550
 QSO35560
 QSO35570
 QSO35580
 QSO35590
 QSO35600
 QSO35610
 QSO35620
 QSO35630
 QSO35640
 QSO35650
 QSO35660
 QSO35670
 QSO35680
 QSO35690
 QSO35700
 QSO35710
 QSO35720
 QSO35730
 QSO35740
 QSO35750
 QSO35760
 QSO35770
 QSO35780
 QSO35790
 QSO35800
 QSO35810
 QSO35820
 QSO35830
 QSO35840
 QSO35850
 QSO35860
 QSO35870
 QSO35880
 QSO35890
 QSO35900
 QSO35910
 QSO35920
 QSO35930
 QSO35940
 QSO35950
 QSO35960
 QSO35970
 QSO35980
 QSO35990
 QSO36000

```

2252 X=REAL(FZTRP(SD,Z2,SP,KENDO))
      GOTO 2245
      DUM=IZTRP(SU,Z,Y,KM,SP)
      X=REAL(FZTRP(SU,Z,SP,KM))

2245 IF(.NOT. TE) Y=(Y-X*SIN(THETL))/COS(THETL)
      IF(TE) Y=(Y-X*SIN(THET))/COS(THET)
      DO (GRID-LINE COORDINATE LOADING)
      C PROCEDURE (GRID-LINE COORDINATE LOADING)
      C ITST=0
      C JTST=0
      DO 21109 I1=1,400
      C XOEFTA(I1)=1.E20
      C YOEFTA(I1)=1.E20
      IF(I1.LE.400) SDXETA(I1)=0.
      IF(I1.LE.400) SDYETA(I1)=0.
      C CONTINUE
21109 DO 21115 I1=1,ENPOLS
      C IF(ABS(XX(I1)).GT.20.) XX(I1,1)=SIGN(7.*ACS(XX(I,2)),XX(I,1))
      C XOEFTA(I1)=CMPLX(ET(I1),XX(I,1,1))
      C YOEFTA(I1)=CMPLX(ET(I1),YY(I,1,1))
      C NOTO=SGEN(SDYETA,YOEFTA,ENPOLS)
      IF(.NOT. NE.O) WRITE(6,2) NOTO
      C NOTO=SGEN(SDXETA,XOEFTA,ENPOLS)
      IF(.NOT. NE.O) WRITE(6,2) NOTO
      C END
      DUM=IZTRP(SDXETA,XOEFTA,X,ENPOLS,SPE)
      ETA=REAL(FZTRP(SDXETA,XOEFTA,SPE,ENPOLS))

      C DO (ETA VALUE STRETCH FOR EDGE REGION POINTS)
      C PROCEDURE (ETA VALUE STRETCH)
      C ENM1=ENO-1
      DO 2196 J=2,ENM1
      C ETANEW=ET(J)/CAPK*ETA
      C SPX=0.
      C SPY=0.
      DUM=RZTRP(SDXETA,XOEFTA,ETANEW,ENPOLS,SPX)
      C XX(I,J)=AIMAG(FZTRP(SDXETA,XOEFTA,SPX,ENPOLS))
      C YY(I,J)=AIMAG(FZTRP(SDYETA,YOEFTA,ETANEW,ENPOLS,SPY))
      C YY(I,J)=AIMAG(FZTRP(SDYETA,YOEFTA,SPY,ENPOLS))
      C END
2196 IF(XX(I,1).EQ.XX(I,ENO-1).AND. YY(I,1).EQ.YY(I,ENO-1)) WRITE (6,7) I
      C FORMAT(1,1) OGRID LINE NO. ,14, HAS DEGENERATED TO POINT.,
      C 1, TRY THETL, THET = LE,TE CAMBER ANGLES.
      C XX(I,ENO)=X

```



```

C
      ZGRID=(XGRID(I,J))+AI*YGRID(I,J)*SIG
      AHATAN=CEXP((ZGRID
      )/2.)*CEXP(-(ZGRID
      )/2.)
      AHYPT=AHATAN/AHTAN2
      ZCMX=ALAMDA*AHYPT
      WCMX=ZCMX+CSQRT(ZCMX-1.)*CSQRT(ZCMX+1.)
      IF(R.LT.1.1.AND.YGRID(I,J).EQ.0.)WCMX=CMPLX(REAL(WCMX),0.)
      IF(ABS(ABS(REAL(WCMX))-R).LT.1.E-07)
      1AQT1=CMLPX(SIGN(25.,XGRID(I,J)),0.)
      IF(ABS(ABS(REAL(WCMX))-R).GE.1.E-07)
      1AQT1=(CLOG(READK*WCMX+1.))-CLOG(READK*WCMX-1.)
      IF(ZET(I).LT.0.)AQT1=AQT1-2.*AI*PI
      ADD STAGGER TERM
      AQT1=AQT1*CEXP(-AI*STAG)
      AQT2=(CLOG(WCMX+READK)-CLOG(WCMX-READK))*CEXP(AI*STAG)
      CORRECT FOR PROPER SOLIDITY COEFFICIENT AND ASSEMBLE X+IY
      ZCMX=(AQT1+AQT2)/(2.*PI*SOLID2)*CHORD
      XGRID(I,J)= REAL(ZCMX)+SIN(STAG)/2./SOLID2 *CHORD
      YGRID(I,J)= SIG*(AIMAG(ZCMX)+COS(STAG)/2./SOLID2 *CHORD)
      YGRID(ZN2M1,J)=YGRID(I,J)
      XGRID(ZN2M1,J)=XGRID(I,J)
      RETURN
      END
      SUBROUTINE ALTERP(COVERS,STAG,ALFA,R)
      PERFORMS DOUBLE LINEAR INTERPOLATION ON STAGGER ANGLE AND SOLIDITY
      Y TO OBTAIN
      POSITIONS OF VORTEX SINGULARITIES IN THE CONFORMAL PLANE.
      SOURCE, HAWTHORNE (LISTED AS REF. 4 OF REF. 2. OF USERS MANUAL)
      DIMENSION CSTAB(25,25),GAMTAB(25),RTAB(25),ISTAR(2)
      TABULATED VALUES FROM THE RANGE OF FIGS B,5C AND B,5D (PP.30,31)
      OF HAWTHORNE
      DATA RTAB,GAMTAB/1.001,
      11.5,1.6,1.7,1.8,1.9,1.2,2.5,2.5,1.005,1.01,1.02,1.03,1.04,1.05,1.1,1.2,1.3,1.4,
      115.,20.,25.,30.,35.,40.,45.,50.,55.,60.,65.,70.,75.,80.,85.,91.,
      195.,100.,4*150./
      PI=3.14159265
      DO 1 J=1,19
      G=GAMTAB(J)*PI/180.

```

```

C      EQNS. FROM FIGS B,5C AND D
1      AL=ATAN(TAN(G)*(RD*RD-1.)/(RD*RD+1.))
      CSTAB(I,J)=1./PI*(COS(G)*ALOG((RD*RD+2.*RD*COS(AL)+1.)/(RD*RD-2.
      1*RD*COS(AL)+1.))+2.*SIN(G)*ATAN(2.*RD*SIN(AL)/(RD*RD-1.)))
      STAGG=STAG*180./PI
20     DO 20 J=1,25
      IF(STAGG.LE.GAMTAB(J))GOTO21
      CONTINUE
      J=25
21     JSTAR=J
      JSM1=J-1
      DO 40 J=JSM1,JSTAR
      J1=J-JSM1+1
      DO 30 I=1,25
      IF(COVERS.GE.CSTAB(I,J))GOTO31
      CONTINUE
      I=25
31     ISTAR(J1)=I
      CONTINUE
      JSM1=ISTAR(2)-1
      ISSM1=ISTAR(1)-1
      ISS=ISTAR(2)
      ISS=ISTAR(1)
      R1=RTAB(ISSM1)+(RTAB(ISSM1)-RTAB(ISS))*(COVERS-CSTAB(ISS,JSM1))/
      1(CSTAB(ISSM1,JSM1)-CSTAB(ISS,JSM1))
      R2=RTAB(ISS)+(RTAB(ISSM1)-RTAB(ISS))*(COVERS-CSTAB(ISS,JSTAR))/
      1(CSTAB(ISSM1,JSTAR)-CSTAB(ISS,JSTAR))
      R=R1+(R2-R1)*(STAGG-GAMTAB(JSM1))/(GAMTAB(JSTAR)-GAMTAB(JSM1))
      IF(COVERS.GT.5)R=1.001
      ALFA=ATAN(TAN(STAG)*(R*-1.)/(R*R+1.))
      RETURN
      END
      FUNCTION FUDS(E,Z,SCK)
C      CALCULATES FUNCTION 'T' IN EQN. 1., REF. 2. OF USER MANUAL
      COMMON/GEOM/ DUMPI(19),G
      COMMON/FUNCOS/S,C,D,SN,CN,DN
      CALL JELF(S,C,D,E,SCK)
      CALL JELF(SN,CN,DN,Z,G)
      B=S*DN*S*DN+(C*D*SN*CN)*#2

```

```

      FUDES = SQRT(B)
      RETURN
    END
    FUNCTION OMOW(SCK)
      C      CALCULATES FUNCTION 'OMEGA' IN EQN. 1., REF. 2 OF USERS MANUAL
      COMMON/FUNCOS/S,C,D,SN,CN,DN
      IF(S.EQ.0.)S=1.E-20
      B= C*D*SN*CN/S/DN
      OMOW = ATAN(B)
      RETURN
    END
    FUNCTION BADZAC(SCK)
      C      CALCULATES FUNCTION 'GAMMA' IN EQN. 1., REF. 2 OF USERS MANUAL
      COMMON/FUNCOS/S,C,D,SN,CN,DN
      BADZAC = CN*CN+(1.-SCK)*S*S*SN*SN
      RETURN
    END
    SUBROUTINE JELF(SN,CN,DN,X,SCK)
      C      COMPUTES JACOBIAN ELLIPTIC FCNS SN,CN,DN
      IF X IS GIVEN AS THE ELLIPTIC INTEGRAL OF THE FIRST KIND WITH MOD
      ULUS K
      FROM ZERO TO SIN(PHI) WITH 0<PHI<PI/2, THEN
        SCK = 1-K*K
        SN(X,K)=SIN(PHI)
        CN(X,K)=COS(PHI)
        DN(X,K)=SQRT(1.-K*K*SIN(PHI)**2)
      AND
      DIMENSION ARI(12),GEO(12)
      CM=SCK
      Y=X
      IF(SCK) 3,1,4
      D=EXP(X)
      A=1./D
      B=A+D
      CN=2./B
      DN=CN
      SN=TANH(X)
      RETURN
    1
    2

```

```

3      D=1.-SCK
      CM=-SCK/D
      D=SQRT(D)
      Y=D*X
      A=1.
      DN=1.
      DO 6 I=1,12
      L=I
      ARI(I)=A
      CM=SQRT(CM)
      GEO(I)=CM
      C=(A+CM)*.5
      IF(ABS(A-CM)-1.E-4*A)7,7,5
      CM=A*CM
      A=C
      Y=C*Y
      SN=SIN(Y)
      CN=COS(Y)
      IF(SN)8,13,8
      A=CN/SN
      C=A*C
      DO 9 I=1,L
      K=L-I+1
      B=ARI(K)
      A=C*A
      C=DN*C
      DN=(GEO(K)+A)/(B+A)
      A=C/B
      A=1./SQRT(C*C+1.)
      IF(SN)10,11,11
      SN=-A
      GOTO 12
11     SN=A
12     CN=C*SN
13     IF(SCK)14,2,2
14     A=DN
      DN=CN
      CN=A
      SN=SN/D
      RETURN
      END
      SUBROUTINE XYCALC(KSTART,K2,Z,X,Y)
      --XYCALC --
      C.....GENERATES DATA FILES FOR ON-BODY POINTS.
      INTEGER SGEN
      REAL
      COMPLEX Z(400),DZ(400),DZ2,ZZ,FZTRP,DZTRP
      C(400),S(400),SP(400)

```

```

QSQ38410
QSQ38420
QSQ38430
QSQ38440
QSQ38450
QSQ38460
QSQ38470
QSQ38480
QSQ38490
QSQ38500
QSQ38510
QSQ38520
QSQ38530
QSQ38540
QSQ38550
QSQ38560
QSQ38570
QSQ38580
QSQ38590
QSQ38600
QSQ38610
QSQ38620
QSQ38630
QSQ38640
QSQ38650
QSQ38660
QSQ38670
QSQ38680
QSQ38690
QSQ38700
QSQ38710
QSQ38720
QSQ38730
QSQ38740
QSQ38750
QSQ38760
QSQ38770
QSQ38780
QSQ38790
QSQ38800
QSQ38810
QSQ38820
QSQ38830
QSQ38840
QSQ38850
QSQ38860
QSQ38870
QSQ38880

```

```

LOGICAL THIN,EVEN,SPGEN,DONE
COMMON /MAN/ DELSMX,PI02,DELS1,IHUB
DIMENSION X(400),Y(400)
COMMON /SPGEN/ A,DSMAX,RMAX,THIN,B,TMIN,DSEND
DATA NSMAX,EMPTY/400,1.0E20/

C..... INITIALIZE PROGRAM.
SI=0.0
NMAX=400
DSMAX=DELSMX
DELS1=DELSMX
THIN=.FALSE.
RMAX=1.2
DSEND=DSMAX
DONE=.FALSE.
EVEN=.FALSE.
B=0.3
THIN=0.1
NFIN=0
NSP=0

C..... INPUT BODY POINTS AND BODY TYPE.
DO 25 I=1,NSMAX
  IF (REAL(Z(I)).EQ.EMPTY) GO TO 30
  NS=I
  S(I)=SI
  IBAD=SGEN(S,Z,NS)
  WRITE(6,140)
  FORMAT(31H)THE FOLLOWING IS S FROM XYCALC(///)
  WRITE(6,*)S
  WRITE(6,145)
  FORMAT(31H)THE FOLLOWING IS Z FROM XYCALC(///)
  WRITE(6,*)Z
  WRITE(6,150)
  FORMAT(32H)THE FOLLOWING IS NS FROM XYCALC(///)
  WRITE(6,*)NS
  IF (IBAD.NE.0) WRITE (6,125)IBAD

C..... SET UP DERIVATIVES + CURVATURES.
DO 35 I=1,NS
  DZ(I)=DZTRP(S,Z,S(I),NS)
  DO 40 I=1,NS
    DZZ=DZTRP(S,DZ,S(I),NS)
    C(I)=AIMAG(CONJG(DZ(I))*DZZ)/CABS(DZ(I))*3

C..... INPUT AUXILIARY (CONTROL) DATA.
IF (NFIN.EQ.0) NFIN=NS
SFIN=S(NFIN)

```

```

Q5038890
Q5038900
Q5038910
Q5038920
Q5038930
Q5038940
Q5038950
Q5038960
Q5038970
Q5038980
Q5038990
Q5039000
Q5039010
Q5039020
Q5039030
Q5039040
Q5039050
Q5039060
Q5039070
Q5039080
Q5039090
Q5039100
Q5039110
Q5039120
Q5039130
Q5039140
Q5039150
Q5039160
Q5039170
Q5039180
Q5039190
Q5039200
Q5039210
Q5039220
Q5039230
Q5039240
Q5039250
Q5039260
Q5039270
Q5039280
Q5039290
Q5039300
Q5039310
Q5039320
Q5039330
Q5039340
Q5039350
Q5039360

```

```

DSMAX=AMAX1(DSMAX,DSEND)
C.....GENERATE BODY POINTS ON A SEGMENT
IF (.NOT.SPGEN(S,Z,C,NS,SP,NSP,SFIN,NMAX)) GO TO 130

C.....OUTPUT RESULTING ON-BODY POINTS.
85 DO 90 I=1,NSP
   IBY=KSTART+I-1
   ZZ=FZTRP(S,Z,SP(I),NS)
   C   DZ(I)=DZTRP(S,Z,SP(I),NS)
   C   DZZ=DZTRP(S,DZ,SP(I),NS)
   C   C(I)=AIMAG(CONJG(DZ(I))*DZZ)/CABS(DZ(I))**3
90   X(I*IBY)=REAL(ZZ)
   Y(I*IBY)=AIMAG(ZZ)
   K1=KSTART
   K2=NSP+KSTART-1
120 RETURN
C.....ERROR MESSAGES
125 FORMAT(21H SGEN FAILED. IBAD=,I3)
130 WRITE(6,135)
135 FORMAT(34H SPGEN UNABLE TO COMPLETE SEGMENT )
STOP
END
INTEGER FUNCTION SGEN(S,F,NS)
REAL S(400)
COMPLEX F(400),DZTRP
DATA MAX,N,FN,TEST/4,10,10.0,0.01/
DO 10 I=2,NS
   S(I)=S(I-1)+CABS(F(I)-F(I-1))
DO 30 K=1,MAX
   SGEN=0
DO 25 I=2,NS
   DS=S(I)-S(I-1)
   DARG=DS/FN
   ARG=DS(I-1)-DARG/2.0
   SUM=0.0
DO 15 J=1,N
   ARG=ARGO+FLOAT(J)*DARG
   TVQQ=CABS(DZTRP(S,F,ARG,NS))-1.0
   SUM=SUM+TVQQ
   SUM=SUM/FN
   ERROR=ABS(SUM)
WRITE(6,19)
19 FORMAT(42H THE FOLLOWING IS TVQQ,SUM,ERROR FROM SGEN)
C   WRITE(6,*)TVQQ,SUM,ERROR
DS=DS*SUM
DO 20 J=1,NS

```

```

QSO39370
QSO39380
QSO39390
QSO39400
QSO39410
QSO39420
QSO39430
QSO39440
QSO39450
QSO39460
QSO39470
QSO39480
QSO39490
QSO39500
QSO39510
QSO39520
QSO39530
QSO39540
QSO39550
QSO39560
QSO39570
QSO39580
QSO39590
QSO39600
QSO39610
QSO39620
QSO39630
QSO39640
QSO39650
QSO39660
QSO39670
QSO39680
QSO39690
QSO39700
QSO39710
QSO39720
QSO39730
QSO39740
QSO39750
QSO39760
QSO39770
QSO39780
QSO39790
QSO39800
QSO39810
QSO39820
QSO39830
QSO39840

```

```

20      S(J)=S(J)+DS
25      IF(ERROR.GT.TEST.AND.SGEN.EQ.0)SGEN=I
30      CONTINUE
      IF(SGEN.EQ.0)RETURN
      CONTINUE
      RETURN
      END
      LOGICAL FUNCTION SPGEN (S,Z,C,NS,SP,NSP,SFIN,NMAX)
      GENERATES TABLE SP HAVING VALUES OF PARAMETER S AS WIDELY SPACED
      AS POSSIBLE AND YET SATISFYING THE FOLLOWING CONDITIONS ON DS:
      1 NSP .LE. NMAX
      2 DS .LE. A/C(S) (C=CURVATURE)
      3 DS .LE. DSMAX
      4 A DS(1) .LE. DS(I-1)*RMAX
      4B DS(1) .GE. DS(I-1)/RMAX
      FOR THIN SECTIONS, AN ADDITIONAL CONDITION IS:
      DS .LE. B*TLOC (TLOC=LOCAL THICKNESS)
      C.....SPGEN = .TRUE. IF ALL CONDITIONS HAVE BEEN SATISFIED.

      REAL S(400),C(400),SP(400)
      COMPLEX Z(400),FZTRP
      LOGICAL THIN,FIN
      COMMON /SPGENC/ A,DSMAX,RMAX,THIN,B,TMIN,DSEND
      COMMON /MAN/ DELSMX,PI02,DELS1,IHUB
      DATA ONE,CMIN/1.0001,1.0E-67

      C..... INITIALIZATION SECTION.
      SPGEN=.FALSE.
      J1=MAX0(NSP,2)+1
      IF (NSP.GT.1) GO TO 15
      IF (NSP.LT.1) SP(1)=S(1)
      DS1=DELS1
      SP(2)=SP(1)+DS1
      WRITE(6,51)
      FORMAT(30H)THE FOLLOWING IS S FROM SPGEN////////
      10  WRITE(6,*)S
      51  WRITE(6,52)
      52  FORMAT(30H)THE FOLLOWING IS C FROM SPGEN////////
      53  WRITE(6,53)
      54  FORMAT(33H)THE FOLLOWING IS SBAR FROM SPGEN////////
      55  WRITE(6,54)
      56  FORMAT(31H)THE FOLLOWING IS NS FROM SPGEN////////
      57  WRITE(6,55)
      BEGIN MAIN LOOP.
      15  DO 45 J=J1,NMAX
          L=J

```

```

5039850
5039860
5039870
5039880
5039890
5039900
5039910
5039920
5039930
5039940
5039950
5039960
5039970
5039980
5039990
5040000
5040010
5040020
5040030
5040040
5040050
5040060
5040070
5040080
5040090
5040100
5040110
5040120
5040130
5040140
5040150
5040160
5040170
5040180
5040190
5040200
5040210
5040220
5040230
5040240
5040250
5040260
5040270
5040280
5040290
5040300
5040310
5040320

```

```

20      I=L
25      SBAR=SP(I-1)-SP(I-2)
        CA=AMAX1(CHIN,ABS(FNTRP(S,C,SBAR,NS)))
        DSLIM=AMIN1(DS1,DSLAST+RMAX)
        IF (.NOT. THIN) GO TO 30
        TLOC=CABS(FZTRP(S,Z,SBAR,NS))-FZRP(S,Z,S(NS)-SBAR,NS)
        DSLIM=AMIN1(DSLIM,B*AMAX1(TLOC,THIN))
30      DSFIN=SFIN-SP(I-1)
        NEVEN=DSFIN/DSLIM/ONE+1.0
        DSEVEN=DSFIN/FLOAT(NEVEN)
        DS=AMIN1(A/CA,DSEVEN)
        IF (I.NE.J) DS=AMIN1(DS,DSLAST/RMAX)

C.....CALCULATED VALUE OF DS SATISFIES CONDITIONS 2 THRU 4A.TEST FOR 4B.
        IF (DS.GE.DSLAST/RMAX) GO TO 40

C..... IF CONDITION 4B IS NOT SATISFIED, RE-DO EARLIER INTERVALS
C..... USING SMALLER VALUES OF DS. IF RE-DOING ALL INTERVALS WON'T
C..... WORK, START OVER USING SMALLER STARTING VALUE OF DS (DS1).
35      L=L-1
        IF (L.GE.J1) GO TO 20
        IF (NSP.GT.1) RETURN
        DS1=DS1/RMAX
        GO TO 10

C..... IF CONDITIONS 2 THRU 4B ARE SATISFIED, TEST FOR FINISH.
40      SP(I)=SP(I-1)+DS
        FIN=SFIN/SP(I).LE.ONE
        IF (FIN.AND.DS.GT.DSEND) GO TO 35
        IF (I.GE.J) GO TO 45
        I=I+1
        GO TO 25
45      CONTINUE
C..... SPGEN=.FALSE. IF CONDITION 1 CANNOT BE SATISFIED.
        RETURN

C..... IF CONDITIONS ARE SATISFIED, UPDATE NSP.
50      NSP=I
        DELSI=DS
        SPGEN=.TRUE.
        RETURN
        END

C..... COMPLEX FUNCTION DZTRP (A,F,X,NA)
        COMPLEX DERIVATIVE EVALUATION FOR DOUBLE 3-POINT INTERPOLATION.
        COMPLEX F(400)
        REAL A(400)

```

QSD40330
 QSD40340
 QSD40350
 QSD40360
 QSD40370
 QSD40380
 QSD40390
 QSD40400
 QSD40410
 QSD40420
 QSD40430
 QSD40440
 QSD40450
 QSD40460
 QSD40470
 QSD40480
 QSD40490
 QSD40500
 QSD40510
 QSD40520
 QSD40530
 QSD40540
 QSD40550
 QSD40560
 QSD40570
 QSD40580
 QSD40590
 QSD40600
 QSD40610
 QSD40620
 QSD40630
 QSD40640
 QSD40650
 QSD40660
 QSD40670
 QSD40680
 QSD40690
 QSD40700
 QSD40710
 QSD40720
 QSD40730
 QSD40740
 QSD40750
 QSD40760
 QSD40770
 QSD40780
 QSD40790
 QSD40800

```

COMMON /NTRPC3/ I1,I2,C(4)
FIRST EVALUATE FUNCTION COEFFICIENTS.
CALL FNTRPA (A,X,NA)
CALL DNTRPC
C..... THEN EVALUATE FUNCTION VALUE.
C..... GO TO 10
C..... ENTRY DZTRP1 (F)
C..... DZTRP1=0.
C..... DZTRP=0.0
C..... DO 10 I=I1,I2
C..... J=J+1
C..... DZTRP=DZTRP+C(J)*F(I)
C..... RETURN
C..... END
C..... COMPLEX FUNCTION FZTRP (A,F,X,NA)
C..... COMPLEX FUNCTION EVALUATION BY DOUBLE 3-POINT INTERPOLATION.
C..... COMPLEX F(400)
C..... REAL A(400)
COMMON /NTRPC2/ I1,I2,C(4)
FIRST EVALUATE FUNCTION COEFFICIENTS.
CALL FNTRPA (A,X,NA)
CALL DNTRPC
C..... THEN EVALUATE FUNCTION VALUE.
C..... GO TO 10
C..... ENTRY FZTRP1 (F)
C..... FZTRP1=0.
C..... FZTRP=0.0
C..... DO 10 I=I1,I2
C..... J=J+1
C..... FZTRP=FZTRP+C(J)*F(I)
C..... RETURN
C..... END
FUNCTION ENTRP (A,F,X,NA)
FUNCTION EVALUATION FOR DOUBLE 3-POINT INTERPOLATION.
REAL F(400),A(400)
COMMON /NTRPC2/ I1,I2,C(4)
FIRST EVALUATE FUNCTION COEFFICIENTS.
CALL FNTRPA (A,X,NA)
CALL DNTRPC
C..... THEN EVALUATE FUNCTION VALUE.
C..... GO TO 5
C..... ENTRY FNTRP1 (F)
C..... FNTRP1=0.
C..... FNTRP=0.0
C..... DO 10 I=I1,I2

```

```

QSO40810
QSO40820
QSO40830
QSO40840
QSO40850
QSO40860
QSO40870
QSO40880
QSO40890
QSO40900
QSO40910
QSO40920
QSO40930
QSO40940
QSO40950
QSO40960
QSO40970
QSO40980
QSO40990
QSO41000
QSO41010
QSO41020
QSO41030
QSO41040
QSO41050
QSO41060
QSO41070
QSO41080
QSO41090
QSO41100
QSO41110
QSO41120
QSO41130
QSO41140
QSO41150
QSO41160
QSO41170
QSO41180
QSO41190
QSO41200
QSO41210
QSO41220
QSO41230
QSO41240
QSO41250
QSO41260
QSO41270
QSO41280

```

```

10      J=J+1
      FNTRP=FNTRP+C(J)*F(I)
      RETURN
END
SUBROUTINE ONTRPC
CALCULATION OF C COEFFICIENTS FOR DERIVATIVES OF DOUBLE
3-POINT INTERPOLATION.
COMMON /NTRPC1/ L1,A11,A12,A13,A14,A22,A23,A24,A33,A34,A44
COMMON /NTRPC3/ I1,I2,C1,C2,C3,C4
IF (L-3) GO TO 13
IF (L-3) I1,I2,I3
FOR DOUBLE 3-POINT INTERPOLATION.
C1=-(A22+A33+A22)/A23/A12/A13
C2=-(A33+A22+A33)/A23/A22/A34
P=A23/A23
C2=-(A11+A33+A11)*A33/A12+(A33*A44+A22*A44+A22*A33)/A24/P
C3=-(A44+A22+A44)*A22/A34+(A22*A11+A33*A11+A33*A22)/A13/P
GO TO 14
FOR SIMPLE 3-POINT INTERPOLATION.
C1=-(A33+A22)/A12/A13
C2=-(A33+A11)/A12/A23
C3=-(A22+A11)/A13/A23
GO TO 14
FOR 2-POINT INTERPOLATION.
C1=1.0/A12
C2=-C1
GO TO 14
ONLY ONE TABLE VALUE GIVEN.
C1=0.0
I1=I
I2=I+L-1
RETURN
END
SUBROUTINE FNTRPA(A,X,NA)
SUBROUTINE SUBROUTINE EVALUATES A COEFFICIENTS IN DOUBLE
3-POINT INTERPOLATIONS.
L=NO. OF POINTS IN THE FIT
I=INDEX TO FIRST POINT
REAL A(400)
COMMON /NTRPC1/ L,I,A11,A12,A13,A14,A22,A23,A24,A33,A34,A44
COMMON AND L BY TABLE LOOK-UP.
GET I AND L
L=LIMIT(1,NA,3)
M=MAXO(1,NA-2)
CALL TLU(A,X,NA,J)
IF (J.EQ.LIMIT(2,J,M)) L=4
L=LIMIT(1,J-1,M)
CALCULATE A-ARRAY.
A11=A(I)

```

Q5041290
 Q5041300
 Q5041310
 Q5041320
 Q5041330
 Q5041340
 Q5041350
 Q5041360
 Q5041370
 Q5041380
 Q5041390
 Q5041400
 Q5041410
 Q5041420
 Q5041430
 Q5041440
 Q5041450
 Q5041460
 Q5041470
 Q5041480
 Q5041490
 Q5041500
 Q5041510
 Q5041520
 Q5041530
 Q5041540
 Q5041550
 Q5041560
 Q5041570
 Q5041580
 Q5041590
 Q5041600
 Q5041610
 Q5041620
 Q5041630
 Q5041640
 Q5041650
 Q5041660
 Q5041670
 Q5041680
 Q5041690
 Q5041700
 Q5041710
 Q5041720
 Q5041730
 Q5041740
 Q5041750
 Q5041760

005041770
 005041780
 005041790
 005041800
 005041810
 005041820
 005041830
 005041840
 005041850
 005041860
 005041870
 005041880
 005041890
 005041900
 005041910
 005041920
 005041930
 005041940
 005041950
 005041960
 005041970
 005041980
 005041990
 005042000
 005042010
 005042020
 005042030
 005042040
 005042050
 005042060
 005042070
 005042080
 005042090
 005042100
 005042110
 005042120
 005042130
 005042140
 005042150
 005042160
 005042170
 005042180
 005042190
 005042200
 005042210
 005042220
 005042230
 005042240

```

A22=A(I+1)
A33=A(I+2)
IF (L.NE.4) IF (L-2) 20,15,10,
A44=A(I+3)
A14=A11-A44
A24=A22-A44
A34=A23-A44
A44=X-A44
A13=A11-A33
A23=A22-A33
A12=A11-A22
A22=X-A22
A11=X-A11
RETURN
END
SUBROUTINE FNTRPC
CALCULATION OF C COEFFICIENTS FOR FUNCTION VALUES BY DOUBLE
3-POINT INTERPOLATION.
COMMON /NTRPC1/ L1,A11,A12,A13,A14,A22,A23,A24,A33,A34,A44
COMMON /NTRPC2/ I1,I2,C1,C2,C3,C4
IF (L.LE.1) GO TO 25
IF (L-3) 20,15,10
FOR DOUBLE 3-POINT INTERPOLATION.
C1=+A33/A23*A22/A12*A33/A13
C4=-A22/A23*A33/A34*A22/A24
P2=A33/A23*A11/A23
P3=A22/A23*A44/A23
C2=-A33*(P2/A12+P3/A24)
C3=+A22*(P3/A34+P2/A13)
GO TO 30
FOR SINGLE 3-POINT INTERPOLATION.
C1=+A22/A12*A33/A13
C2=-A11/A12*A33/A23
C3=+A11/A13*A22/A23
GO TO 30
FOR 2-POINT INTERPOLATION.
C1=+A22/A12
C2=-A11/A12
GO TO 30
ONLY ONE TABLE VALUE GIVEN.
C1=1.0
I1=I
I2=I+L-1
RETURN
END
SUBROUTINE TLU (TABLE,ARG,N,I)
TABLE LOOK UP: FINDS I SUCH THAT
  
```

```

C ARG-GE.TABLE(I),AND-ARG.LT.TABLE(I+1)
C IF I=0, ARG.LT.TABLE(I)
C IF I=N, ARG-GE.TABLE(N)
C REAL TABLE(400)
C I=LIMIT(I,I,N)
C IF (ARG-GE.TABLE(I)) GO TO 15
C.....DESCEND IN TABLE.
10 I=I-1
C IF (I.LE.0) RETURN
C IF (ARG-GE.TABLE(I)) RETURN
C GO TO 10
C.....ASCEND IN TABLE.
15 IF (I.GE.N) RETURN
C IF (ARG.LT.TABLE(I+1)) RETURN
C I=I+1
C GO TO 15
C.....
C FUNCTION LIMIT (I,J,K)
C INTEGER FUNCTION LIMITS J BETWEEN I AND K.
C.....LIMIT=I
C IF (J.LT.LIMIT) RETURN
C LIMIT=K
C IF (J.GT.LIMIT) RETURN
C LIMIT=J
C RETURN
C.....
C LOGICAL FUNCTION XNTRP (A,F,FO,NA,X)
C INVERSE FUNCTION FOR DOUBLE 3-POINT INTERPOLATION.
C.....
C GIVEN TABLES A AND F (EACH OF LENGTH NA), THE VALUE OF X IS
C FOUND FOR WHICH FNTRP (A,F,X,NA)=FO. ONLY SOLUTIONS LARGER
C THAN THE ENTRY VALUE OF X ARE CONSIDERED; ONLY THE SMALLEST
C OF THESE IS RETURNED. IF NO SOLUTION IS FOUND, XNTRP=.FALSE.
C AND X IS LEFT UNALTERED.
C REAL A(400),F(400),C(4),R(3)
C INTEGER I(3),REALX,RCMPLX,ICMPLX,TYPE
C LOGICAL LAST,SIZE
C COMMON /NTRPC1/ L,I,EXTRA(10)
C COMMON /NTRPC5/ M,B(4,4)
C DATA REALX,RCMPLX,ICMPLX,EPSLON/0,1,2,1.0E-5/
C COMPLEX X,FZ(400)
C TYPE=REALX
C GO TO 100
C.....ALTERNATE ENTRIES FOR COMPLEX TYPE FUNCTION TABLES:
C.....RZTRP IS USED TO GET X FOR WHICH REAL(FZTRP(A,F,X,NA))=FO.

```

```

C..... ENTRY RZTRP (A,FZ,FO,NA,X)
RZTRP=0.FZ(NA) THIS WAS MOVED UP TO COMPILE PROPERLY ON NPS 370
TYPE=RCMPLX
GO TO 100
Q5042730
Q5042740
Q5042750
Q5042760
Q5042770
Q5042780
Q5042790
Q5042800
Q5042810
Q5042820
Q5042830
Q5042840
Q5042850
Q5042860
Q5042870
Q5042880
Q5042890
Q5042900
Q5042910
Q5042920
Q5042930
Q5042940
Q5042950
Q5042960
Q5042970
Q5042980
Q5042990
Q5043000
Q5043010
Q5043020
Q5043030
Q5043040
Q5043050
Q5043060
Q5043070
Q5043080
Q5043090
Q5043100
Q5043110
Q5043120
Q5043130
Q5043140
Q5043150
Q5043160
Q5043170
Q5043180
Q5043190
Q5043200

C..... IZTRP IS USED TO GET X FOR WHICH AIMAG(FZTRP(A,F,X,NA))=FO.
ENTRY IZTRP (A,FZ,FO,NA,X)
IZTRP=0.
TYPE=ICMPLX
C..... INITIALIZATION.
100 XIN=X
XMAX=XIN
DX=0.0
Q5042730
Q5042740
Q5042750
Q5042760
Q5042770
Q5042780
Q5042790
Q5042800
Q5042810
Q5042820
Q5042830
Q5042840
Q5042850
Q5042860
Q5042870
Q5042880
Q5042890
Q5042900
Q5042910
Q5042920
Q5042930
Q5042940
Q5042950
Q5042960
Q5042970
Q5042980
Q5042990
Q5043000
Q5043010
Q5043020
Q5043030
Q5043040
Q5043050
Q5043060
Q5043070
Q5043080
Q5043090
Q5043100
Q5043110
Q5043120
Q5043130
Q5043140
Q5043150
Q5043160
Q5043170
Q5043180
Q5043190
Q5043200

C..... CALCULATE POLYNOMIAL COEFFICIENTS.
1 XMTN=AMAX1(XIN+DX,XMAX-DX)
CALL BNTRP (A,XMAX,NA)
LAST=M.GE.NA-1
IF (LAST) GO TO 2
MAX=M+L-3
XMAX=A(MAX)
DO 3 J=1,L
C(J)=0.0
DO 3 N=1,L
IN=1+N-1
IF (TYPE.EQ.REALX) C(J)=C(J)+B(N,J)*F(IN)
IF (TYPE.EQ.RCMPLX) C(J)=C(J)+B(N,J)*REAL(FZ(IN))
IF (TYPE.EQ.ICMPLX) C(J)=C(J)+B(N,J)*AIMAG(FZ(IN))
CONTINUE
C(1)=C(1)-FO
IF (L.EQ.4.AND.ABS(C(L)).LT.1.E-04) L=3
LL=L
IF (LL.LT.2) GO TO 8
IF (C(LL).NE.0.0) GO TO 5
LL=LL-1
GO TO 4
Q5042730
Q5042740
Q5042750
Q5042760
Q5042770
Q5042780
Q5042790
Q5042800
Q5042810
Q5042820
Q5042830
Q5042840
Q5042850
Q5042860
Q5042870
Q5042880
Q5042890
Q5042900
Q5042910
Q5042920
Q5042930
Q5042940
Q5042950
Q5042960
Q5042970
Q5042980
Q5042990
Q5043000
Q5043010
Q5043020
Q5043030
Q5043040
Q5043050
Q5043060
Q5043070
Q5043080
Q5043090
Q5043100
Q5043110
Q5043120
Q5043130
Q5043140
Q5043150
Q5043160
Q5043170
Q5043180
Q5043190
Q5043200

C..... GET ROOTS OF POLYNOMIAL.
5 NR=1
IF (LL.EQ.2) R(1)=-C(1)/C(2)
IF (LL.EQ.3) NR=NRRTS2(C,R)
IF (LL.EQ.4) NR=NRRTS3(C,R)
Q5042730
Q5042740
Q5042750
Q5042760
Q5042770
Q5042780
Q5042790
Q5042800
Q5042810
Q5042820
Q5042830
Q5042840
Q5042850
Q5042860
Q5042870
Q5042880
Q5042890
Q5042900
Q5042910
Q5042920
Q5042930
Q5042940
Q5042950
Q5042960
Q5042970
Q5042980
Q5042990
Q5043000
Q5043010
Q5043020
Q5043030
Q5043040
Q5043050
Q5043060
Q5043070
Q5043080
Q5043090
Q5043100
Q5043110
Q5043120
Q5043130
Q5043140
Q5043150
Q5043160
Q5043170
Q5043180
Q5043190
Q5043200

C..... TEST ROOTS OF POLYNOMIAL.
IF (NR.LE.0) GO TO 8
IF (.NOT.SIZE(R,II,NR)) CALL ORDER (R,II,NR)
Q5042730
Q5042740
Q5042750
Q5042760
Q5042770
Q5042780
Q5042790
Q5042800
Q5042810
Q5042820
Q5042830
Q5042840
Q5042850
Q5042860
Q5042870
Q5042880
Q5042890
Q5042900
Q5042910
Q5042920
Q5042930
Q5042940
Q5042950
Q5042960
Q5042970
Q5042980
Q5042990
Q5043000
Q5043010
Q5043020
Q5043030
Q5043040
Q5043050
Q5043060
Q5043070
Q5043080
Q5043090
Q5043100
Q5043110
Q5043120
Q5043130
Q5043140
Q5043150
Q5043160
Q5043170
Q5043180
Q5043190
Q5043200

```

```

DO 6 N=1,NR
XVAL=R(N)+A(M)
IF (XVAL.GT.XMIN) GO TO 7
CONTINUE
GO TO 8
IF (LAST .OR. XVAL.LE.XMAX) X=XVAL
IF (ABS(XVAL-XMAX).LE.1.E-04) X=XVAL
XNTRP=X.NE.XIN
C..... IF NO SOLUTION FOUND, KEEP SEARCHING.
9 IF (XNTRP.OR.LAST) RETURN
DX=EPSILON*(ABS(XMAX)+ABS(XMIN))
GO TO 1
END
C..... SUBROUTINE BNTRP (A,X,NA)
CALCULATION OF MATRIX B BY WHICH COEFFICIENTS OF THE POLYNOMIAL
USED IN DOUBLE 3-POINT INTERPOLATION MAY BE OBTAINED.
IF F(X) = C(1) + C(2)*(X-A(M)) + C(3)*(X-A(M))**2 +...
IS THE POLYNOMIAL USED IN THE NEIGHBORHOOD OF X, THE COEFFICIENTS
CAN BE OBTAINED FROM MATRIX B BY:
C(J) = B(1,J)*F(1) + B(2,J)*F(1+1) +...
FOR A TOTAL OF L VALUES OF C AND F (AT MOST 4) WHERE L AND I ARE
FOUND IN COMMON /NTRPC1/ AND M AND B ARE FOUND IN COMMON /NTRPC5/.
REAL A(NA)
COMMON /NTRPC1/ L,I,A11,A12,A13,A14,A22,A23,A24,A33,A34,A44
COMMON /NTRPC5/ M,B11,B21,B31,B41,B12,B22,B32,B42,B13,B23,B33,
1 B43,B14,B24,B34,B44
C..... SET UP COMMON BLOCK /NTRPC1/
CALL FNTRPA (A,X,NA)
IF (L.GT.1) GO TO 2
C..... ENTER HERE FOR ONLY ONE COEFFICIENT.
1 M=1
B11=1.0
RETURN
C..... ENTER HERE FOR 2 OR MORE COEFFICIENTS.
2 M=1+1
Q=1.0/A12
IF (L.GT.2) GO TO 3
B12=Q
B22=-Q

```

Q5043690
Q5043700
Q5043710
Q5043720
Q5043730
Q5043740
Q5043750
Q5043760
Q5043770
Q5043780
Q5043790
Q5043800
Q5043810
Q5043820
Q5043830
Q5043840
Q5043850
Q5043860
Q5043870
Q5043880
Q5043890
Q5043900
Q5043910
Q5043920
Q5043930
Q5043940
Q5043950
Q5043960
Q5043970
Q5043980
Q5043990
Q5044000
Q5044010
Q5044020
Q5044030
Q5044040
Q5044050
Q5044060
Q5044070
Q5044080
Q5044090
Q5044100
Q5044110
Q5044120
Q5044130
Q5044140
Q5044150
Q5044160

```

GO TO 24
C.....ENTER HERE FOR 3 OR MORE COEFFICIENTS.
3
P=1.0/A23
IF (L.GT.3) GO TO 4
B13=Q/A13
B23=-Q*P
B33=P/A13
GO TO 34

C.....ENTER HERE FOR 4 COEFFICIENTS.
4
P2=P*Q/A13
B14=-P2/A12-P2/A24
B34=P2/A34+P2/A13
B44=-P/A34/A24
B13=(Q+Q1)/A13
B23=-P*(Q+Q1)-P/A24
B33=P2*(A24/A34+(A23-A12)/A13)
B43=-1.0/A34/A24
B41=0.0
B42=0.0
B12=A23/A12/A13
B22=P-Q
B32=-A12/A13/A23
B31=0.0
B11=0.0
B21=1.0
RETURN
END
INTEGER FUNCTION NRRTS3 (C,R)
FINDS REAL ROOTS R1 OF A CUBIC WITH COEFFICIENTS C WHEN:
C(1) + C(2)*X + C(3)*X**2 + C(4)*X**3 = 0.0
C.....
REAL C(4),R(3),K1,K2
DATA THIRD,K1,K2/0.3333333,2.094395,4.188790/
QRT(ARG)=SIGN(ABS(ARG)**THIRD,ARG)

C.....CONVERT TO NORMAL FORM AND CALCULATE NORMAL COEFFICIENTS.
CNORM=3.0*C(4)
P=C(13)/CNORM
Q=C(2)/CNORM
B=P**3-1.5*(P*Q-C(1)/CNORM)
B2=B**2
A=Q-P**P
A3=A**3
RAD=B2+A3
IF (RAD) 3,2,1

```

```

C.....THERE IS ONE REAL ROOT.
1  NRRTS3=1
   IF (RAD.NE.0.0) RAD=SQRT(RAD)
   A=-B+RAD
   B=-B-RAD
   IF (A.NE.0.0) A=QRT(A)
   IF (B.NE.0.0) B=QRT(B)
   R(1)=A+B
   GO TO 10

C.....THERE ARE TWO REAL ROOTS.
2  NRRTS3=2
   IF (B.EQ.0.0) GO TO 1
   R(1)=QRT(B)
   R(2)=-2.0*R(1)
   GO TO 10

C.....THERE ARE THREE REAL ROOTS.
3  NRRTS3=3
   PHI=ARCCOS(SIGN(SQRT(-B2/A3),-B1)/3.0)
   CR=2.0*SQRT(-A)
   R(1)=CR*COS(PHI)
   R(2)=CR*COS(PHI+K1)
   R(3)=CR*COS(PHI+K2)

C.....CONVERT BACK TO ORIGINAL FORM.
10 DO 11 I=1,NRRTS3
11  R(I)=R(I)-P
   RETURN

C.....ENTRY NRRTS2 (C,R)
C.....FINDS REAL ROOTS R OF A QUADRATIC WITH COEFFICIENTS C, WHEN:
C       $C(1) + C(2)*X + C(3)*X^2 + = 0.0$ 

C.....CALCULATE RADICAL.
RAD=C(2)*2-4.0*C(3)*C(1)
IF (RAD) 21,22,23

C.....NO REAL ROOTS.
21 NRRTS3=0
   NRRTS2=0
   RETURN

C.....ONE REAL ROOT.
22 NRRTS3=1
   R(1)=-C(2)/2.0/C(3)
   NRRTS2=1

```

```

QSO44170
QSO44180
QSO44190
QSO44200
QSO44210
QSO44220
QSO44230
QSO44240
QSO44250
QSO44260
QSO44270
QSO44280
QSO44290
QSO44300
QSO44310
QSO44320
QSO44330
QSO44340
QSO44350
QSO44360
QSO44370
QSO44380
QSO44390
QSO44400
QSO44410
QSO44420
QSO44430
QSO44440
QSO44450
QSO44460
QSO44470
QSO44480
QSO44490
QSO44500
QSO44510
QSO44520
QSO44530
QSO44540
QSO44550
QSO44560
QSO44570
QSO44580
QSO44590
QSO44600
QSO44610
QSO44620
QSO44630
QSO44640

```

QSD44650
 QSD44660
 QSD44670
 QSD44680
 QSD44690
 QSD44700
 QSD44710
 QSD44720
 QSD44730
 QSD44740
 QSD44750
 QSD44760
 QSD44770
 QSD44780
 QSD44790
 QSD44800
 QSD44810
 QSD44820
 QSD44830
 QSD44840
 QSD44850
 QSD44860
 QSD44870
 QSD44880
 QSD44890
 QSD44900
 QSD44910
 QSD44920
 QSD44930
 QSD44940
 QSD44950
 QSD44960
 QSD44970
 QSD44980
 QSD44990
 QSD45000
 QSD45010
 QSD45020
 QSD45030
 QSD45040
 QSD45050
 QSD45060
 QSD45070
 QSD45080
 QSD45090
 QSD45100
 QSD45110
 QSD45120

```

RETURN
C.....TWO REAL ROOTS.
23 NRRTS3=2
RAD=SQR(TWOA)
TWOA=2.0*C(3)
R(1)=-((RAD+C(2))/TWOA)
R(2)=+((RAD-C(2))/TWOA)
NRRTS2=2
RETURN
END
LOGICAL FUNCTION SIZE (X,I,N)
C.....CREATES THE POINTER ARRAY I, ORDERED BY SIZE OF X.
C.....VALUE IS .TRUE. IF X ARRAY IS IN ORDER.
C.....DIMENSION X(3),I(3)
COMPLEX Z(3),ZS
SIZE=.TRUE.
DO 2 J=1,N
K=J
IF (K.LE.1) GO TO 2
L=I (K-1)
IF (X(J).GE.X(L)) GO TO 2
I(K)=L
K=K-1
SIZE=.FALSE.
GO TO 1
I(K)=J
RETURN
ENTRY ORDER (X,I,N)
ORDER=0.
C.....REARRANGES X ARRAY ACCORDING TO POINTER ARRAY I.
DO 14 J=1,N
IF (I(J).LE.0) GO TO 13
M=J
S=X(M)
L=I(M)
I(M)=-L
IF (L.EQ.J) GO TO 12
X(M)=X(L)
M=L
GO TO 11
11 X(M)=S
I(J)=IABS(I(J))
12 CONTINUE
13 RETURN
ENTRY ZORDER (Z,I,N)
ZORDER=0.
C.....REARRANGES Z (COMPLEX) ARRAY ACCORDING TO POINTER ARRAY I.
  
```

```

C      COMPLEX Z(N),ZS THIS WAS MOVED UP TO COMPILE PROPERLY ON NPS 370
DO 24 J=1,N
IF (I(J).LE.0) GO TO 23
M=J
ZS=Z(M)
L=I(M)
I(M)=-L
IF (L.EQ.J) GO TO 22
Z(M)=Z(L)
M=L
GO TO 21
Z(M)=ZS
I(J)=IABS(I(J))
CONTINUE
RETURN
END
SUBROUTINE PJGRID(ZET,ET,ZCAM,IMAXY,PITCH)

C      ELECTROSTATIC ANALOG GRID GENERATOR
C      J. J. ADAMCZYK
C      NASA LEWIS RESEARCH CENTER 1980
C      (SEE REF. 2, QSONIC USERS MANUAL)

COMMON/ENTIRE/ X(100,30),Y(100,30),ETA(100,30),ZETA(100,30)
COMMON/CROSS/XQ(4,100),YQ(4,100),EQ(4,100),ZQ(4,100),ZETB(100),
1 STREN(63),FUN(100),GUN(100),ARG(100),ZBODY(63),DTREN(63)
COMMON/GEOM/NZGRID,NEGRID,RLE,RTE,C,CHORB,XUPS,XDNS,STAG,CC,CAP(4)
1,PI
1,RNK,V8,KN,NUM
COMPLEX A,B,ZO,ZA,ZB,ZTEMP,ZTEMPQ
COMPLEX SUM3,SUM4,SUM5,SUM6,STREN
COMPLEX SUMD,HSIN
COMPLEX ZTEL,A1,HTAN,Z,SUM
COMPLEX SOUR,ZCAM,ZBODY,ZPET
COMPLEX ZST
DOUBLE PRECISION DTREN,BINF
DIMENSION ZPET(100,30),ZCAM(100),ET(100),ZET(100)
EQUIVALENCE (ZPET(1,1),ETA(1,1))
COMMON/CRHOSS/ BINF(63,63),SOUR(63,63)
LOGICAL XNTRP,TEST
NETAO=50
NZETO=50
NZET4=50

```

```

IMID=1
NUMX=31
NUMY=14
NUMZ=16
C
C
C
NZGRID MUST BE ODD
AI=CMPLX(0.0,1.0)
PI=3.1415927
CHORD=CHORD*COS(STAG)
STAG=STAG*180./PI
S=PITCH
WRITE(6,1006) NEGRID,NZGRID,IMAXY
FORMAT(5X,'NEGRID=',I3,' NZGRID,ODD= ',I3,' IMAXY= ',I3)
CONTINUE
WRITE(6,1020)
FORMAT(10X,'SURFACE COORDINATES')
WRITE(6,1030) (ZCAM(I),I=1,NUM)
FORMAT(7X,2F12.7)
WRITE(6,1040) NUM,PITCH,CHORD,STAG
FORMAT(4X,'NUMBER OF BODY POINTS',I5,10X,'PITCH',F12.7,10X,' CHO
1 F12.7,' STAGGER',F12.7)
ZST=ZCAM(1)
DO 25 I=1,NUM
ZCAM(I)=ZCAM(I)-ZST
CONTINUE
MUM=NUM-1
PSAVE=PITCH
PITCH=2.0*PITCH
XMAX=0.0
IMAX=1
DO 6000 I=1,NUM
IF(XMAX .LE. REAL(ZCAM(I))) XMAX=REAL(ZCAM(I))
IF(XMAX .LE. REAL(ZCAM(I))) IMAX=I
CONTINUE
DO 30 I=1,MUM
IF(IMID .EQ. 1) ZBODY(I)=(ZCAM(I)+ZCAM(I+1))/2.0
DO 30 J=1,MUM
A=ZCAM(J)
B=ZCAM(J+1)
ZA=(A+B)/2.0
ZB=CLOG(PI/PITCH*(ZBODY(I)-A))
ZB=CLOG(PI/PITCH*(ZBODY(I)-B))
IF(I .EQ. J) ZA=ZA
IF(I .EQ. J) ZB=ZB
IF(I .LT. J .AND. AIMAG(ZA) .LT. 0.0 .AND. I .LT. IMAXY) ZA=ZA+2.0
IF(I .LT. J .AND. AIMAG(ZB) .LT. 0.0 .AND. I .LT. IMAXY) ZB=ZB+2.0
**AI#PI
**AI#PI

```

```

IF(I .LT. J .AND. AIMAG(ZB) .LT. 0.0 .AND. I .LT. IMAXY) ZB=ZB+2.0QSD46090
*AI*PI
IF(I .GT. J) ZA=ZA
IF(I .GT. J) ZB=ZB
IF(I .GE. J .AND. J .GE. IMAX/2.0 .AND. AIMAG(ZA) .GT. PI/2.0) ZA=
*ZA-2.0*PI*AI
IF(I .GT. J .AND. J .GE. IMAX/2.0 .AND. AIMAG(ZB) .GT. PI/2.0) ZB=ZB
*-2.0*PI*AI
CONTINUE
SOUR(I,J)=(ZBODY(I)-A)*ZA-(ZBODY(I)-B)*ZB+(A-B)
ZTEL=(ZBODY(I)-Z0)*PI/PI*CH
HTAN=(CEXP(ZTEL)-CEXP(-ZTEL))/(CEXP(ZTEL)+CEXP(-ZTEL))
IF(CABS(ZTEL) .EQ. 0.0) SOUR(I,J)=SOUR(I,J)
IF(CABS(ZTEL) .NE. 0.0) SOUR(I,J)=SOUR(I,J)+CLOG(HTAN/ZTEL)*(B-A)
A1=ABS(REAL(B-A))
A2=ABS(AIMAG(B-A))
IF(A1 .GE. A2) BINF(I,J)=REAL(SOUR(I,J))+AIMAG(B-A)/REAL(B-A)*AIMAG
*G(SOUR(I,J))
IF(A1 .LT. A2) BINF(I,J)=-AIMAG(SOUR(I,J))-REAL(B-A)/AIMAG(B-A)*REAL
*AL(SOUR(I,J))
CONTINUE
30
C
C
C
SOLUTION OF LINEAR EQUATIONS FOR ELECTROSTATIC CHARGE STRENGTH DIS
TRIBUTION
IF(NUM .GT. 1) GO TO 40
STREN(I)=1.0/BINF(I,1)
GO TO 50
CONTINUE
DO 45 I=1,MUM
DTREN(I)=1.0DO
CONTINUE
40
45
50
CALL CHLSKY(BINF,DTREN,MUM)
CONTINUE
DO 51 I=1,MUM
A=ZCAM(I)
B=ZCAM(I+1)
A1=ABS(REAL(B-A))
A2=ABS(AIMAG(B-A))
STEMP=DTREN(I)
IF(A1 .GE. A2) STREN(I)=STEMP-AI*STEMP*AIMAG(B-A)/REAL(B-A)
IF(A1 .LT. A2) STREN(I)=-STEMP*REAL(B-A)/AIMAG(B-A)+AI*STEMP
CONTINUE
51

```

```

DO 55 I=1,MUM
SUM2=0.0
SUM7=0.0
DO 56 J=1,MUM
A=ZCAM(J+1)
B=ZCAM(J+1)
A1=ABS(REAL(B-A))
A2=ABS(AIMAG(B-A))
IF(A1 .GE. A2) SUM2=SUM2+BINF(I,J)*REAL(STREN(J))
IF(A1 .LT. A2) SUM2=SUM2+BINF(I,J)*AIMAG(STREN(J))
FUN1=AIMAG(SOUR(I,J))-REAL(SOUR(I,J))*AIMAG(B-A)/REAL(B-A)
SUM7=SUM7+REAL(STREN(J))*FUN1
CONTINUE
ZETB(I)=SUM7
CONTINUE
CHORX=CHORD
DELX=3.0*CHORX/(NUMX-1.0)
DO 60 I=1,1
DO 60 J=1,NUMY
X(I,J)=-2.0*PSAVE
CONTINUE
DELY=PSAVE/(NUMY-2)
DO 70 I=1,1
DO 70 J=1,NUMY
Y(I,J)=DELY*(J-2)
IF(J .EQ. 1) Y(I,J)=-0.00001
IF(J .EQ. 2) Y(I,J)=0.00001
SUM=0.0
Z=X(I,J)+AI*Y(I,J)
IF(X(I,J) .LT. 0.0 .OR. X(I,J) .GT. XMAX) GO TO 7031
DO 7060 L=1,IMAX
IF(X(I,J) .GE. REAL(ZCAM(L)) .AND. X(I,J) .LE. REAL(ZCAM(L+1))) GO
* 0 TO 7080
CONTINUE
CONTINUE
L1=L
DO 7100 L=IMAX,NUM
IF(X(I,J) .LE. REAL(ZCAM(L)) .AND. X(I,J) .GE. REAL(ZCAM(L+1))) GO
* TO 7120
CONTINUE
CONTINUE
L2=L
YA=AIMAG(ZCAM(L1))
YR=AIMAG(ZCAM(L1+1))
XA=REAL(ZCAM(L1))
XB=REAL(ZCAM(L1+1))
YI=YA+(YB-YA)/(XB-XA)*(X(I,J)-XA)
YA=AIMAG(ZCAM(L2))

```

56

55

60

7060

7080

7100

7120

7031	YB=AIMAG(ZCAM(L2+1))	Q5047050
	XA=REAL(ZCAM(L2+1))	Q5047060
	XB=REAL(ZCAM(L2+1))	Q5047070
	Y2=YA+(YB-YA)/(XB-XA)*(X(I,J)-XA)	Q5047080
	CONTINUE	Q5047090
	DO 75 IX=1,MUM	Q5047100
	A=ZCAM(IX)	Q5047110
	B=ZCAM(IX+1)	Q5047120
	ZA=LOG(PI/PITCH*(Z-A))	Q5047130
	ZB=LOG(PI/PITCH*(Z-B))	Q5047140
	IF(X(I,J).GT.0.0) GO TO 7000	Q5047150
	IFLA=1	Q5047160
	IF(Y(I,J).GE.0.0) GO TO 7010	Q5047170
	ZA=ZA	Q5047180
	ZB=ZB	Q5047190
	GO TO 7020	Q5047200
7010	CONTINUE	Q5047210
	IF(AIMAG(ZA).LT.0.0) ZA=ZA+2.0*AI*PI	Q5047220
	IF(AIMAG(ZB).LT.0.0) ZB=ZB+2.0*AI*PI	Q5047230
	GO TO 7020	Q5047240
7000	CONTINUE	Q5047250
	IF(X(I,J).GT.XMAX) GO TO 7040	Q5047260
	IF(Y(I,J).LT.Y1.AND.Y(I,J).GT.Y2) IFLA=2	Q5047270
	IF(Y(I,J).LT.Y1) GO TO 7140	Q5047280
	IF(AIMAG(ZA).LT.-PI/2.0) ZA=ZA+2.0*AI*PI	Q5047290
	IF(AIMAG(ZB).LT.-PI/2.0) ZB=ZB+2.0*AI*PI	Q5047300
	GO TO 7020	Q5047310
7140	CONTINUE	Q5047320
	IF(Y(I,J).LE.Y2) GO TO 7160	Q5047330
	IF(AIMAG(ZA).LT.0.0.AND.IX.GE.IMAX) ZA=ZA+2.0*AI*PI	Q5047340
	IF(AIMAG(ZB).LT.0.0.AND.IX.GE.IMAX) ZB=ZB+2.0*AI*PI	Q5047350
	GO TO 7020	Q5047360
7160	CONTINUE	Q5047370
	ZA=ZA	Q5047380
	ZB=ZB	Q5047390
	IF(IX.GE.IMAXY.AND.AIMAG(ZA).GT.PI/2.0) ZA=ZA-2.0*PI*AI	Q5047400
	IF(IX.GE.IMAXY.AND.AIMAG(ZB).GT.PI/2.0) ZB=ZB-2.0*PI*AI	Q5047410
	GO TO 7020	Q5047420
7040	CONTINUE	Q5047430
	IFLA=1	Q5047440
	ZA=ZA	Q5047450
	ZB=ZB	Q5047460
7020	CONTINUE	Q5047470
5000	CONTINUE	Q5047480
	SUM=SUM+STREN(IX)*(A-B+(Z-A)*ZA-(Z-B)*ZB)	Q5047490
	ZO=(A+B)/2.0	Q5047500
	ZTEL=(Z-ZO)*PI/PITCH	Q5047510
	HTAN=(CEXP(ZTEL)-CEXP(-ZTEL))/(CEXP(ZTEL)+CEXP(-ZTEL))	Q5047520

```

SUM=SUM+STREN(IX)*CLOG(HTAN/ZTEL)*(B-A)
IF(I.GT.1.AND.I.LT.NUMX) GO TO 3010
SUM3=(A-B)+(Z-A)*ZA-(Z-B)*ZB
SUM3=SUM3*STREN(IX)
SUM4=CLOG(HTAN/ZTEL)*(B-A)
SUM4=SUM4*STREN(IX)
SUM5=CLOG(1.0/ZTEL)*(B-A)
SUM5=SUM5*STREN(IX)
SUM6=CLOG(HTAN)*(B-A)
SUM6=SUM6*STREN(IX)
CONTINUE
3010
75
IF(X(I,J).LT.XMAX.AND.X(I,J).GT.0.0.AND.Y(I,J).LT.Y1.
*AND
1.Y(I,J).GT.Y2) ETA(I,J)=1.05
ZETA(I,J)=AIMAG(SUM)
CONTINUE
NUMD=NUMY-1
DO 310 ID=1,NUMD
ARG(ID)=Y(I,ID+1)
FUN(ID)=ETA(I,ID+1)
GUN1(ID)=ZETA(I,ID+1)
CONTINUE
310
FO=0.0
YSTAR2=.95*PSAVE/2.
TEST=XNTRP(ARG,FUN,FO,NUMD,YSTAR2)
ZETA2=FNTRP(ARG,GUN1,YSTAR2,NUMD)
ETEST=0.0
ITEST=1
DO 330 I=1,NUMD
IF(ETEST.LE.ETA(1,I+1)) ETEST=ETA(1,I+1)
IF(ETEST.LE.ETA(1,I+1)) ITEST=I
CONTINUE
AQ1=ZETA(1,ITEST+1)
AQ2=Y(1,ITEST+1)
AQ3=(ZETA(1,ITEST+2)-ZETA(1,ITEST))/(2.0*DELT)
*2)
BQ3=(ETA(1,ITEST+2)-ETA(1,ITEST))/(2.0*DELT)
BQ3=-BQ2/(2.0*BQ3)
YLMAX=AQ1+AQ2*YLMAX+AQ3*YLMAX**2
DELZ=ZETA(1,2)-ZETA(1,1)
DO 360 I=1,MUM
ARG(I)=ZETA(I)
FUN(I)=REAL(ZBODY(I))

```

5048010
 5048020
 5048030
 5048040
 5048050
 5048060
 5048070
 5048080
 5048090
 5048100
 5048110
 5048120
 5048130
 5048140
 5048150
 5048160
 5048170
 5048180
 5048190
 5048200
 5048210
 5048220
 5048230
 5048240
 5048250
 5048260
 5048270
 5048280
 5048290
 5048300
 5048310
 5048320
 5048330
 5048340
 5048350
 5048360
 5048370
 5048380
 5048390
 5048400
 5048410
 5048420
 5048430
 5048440
 5048450
 5048460
 5048470
 5048480

```

360 CONTINUE
    XBOD1=FNTRP(ARG,FUN,ZETA4,MUM)
    XBOD3=FNTRP(ARG,FUN,0.0,MUM)
    DO 370 I=1,MUM
      FUN(I)=AIMAG(ZBODY(I))
    CONTINUE
370 YBOD1=FNTRP(ARG,FUN,ZETA4,MUM)
    YBOD3=FNTRP(ARG,FUN,0.0,MUM)
    ILIM=MUM+2
    XQ(1,1)=XBOD1
    YQ(1,1,1)=YBOD1
    ZQ(1,1,1)=ZETA4
    EQ(1,1,1)=1.0
    DO 380 I=1,MUM
      IF(ZETB(I).GT. ZETA4) GO TO 390
    CONTINUE
380 ISTART=1
390 IEND=ISTART-1
    DO 410 I=ISTART,MUM
      IL=I-1
      XQ(I,1,1)=REAL(ZBODY(I))
      YQ(I,1,1)=AIMAG(ZBODY(I))
      ZQ(I,1,1)=ZETB(I)
      EQ(I,1,1)=1.0
    CONTINUE
410 IEND=ISTART-1
    DO 420 I=1,IEND
      IL=MUM-1
      XQ(I,1,1)=REAL(ZBODY(I))
      YQ(I,1,1)=AIMAG(ZBODY(I))
      ZQ(I,1,1)=ZETB(I)-DELZ
      EQ(I,1,1)=1.0
    CONTINUE
420

C      DETERMINE LOCATION OF PERIODIC BOUNDARY POINTS
    DO 110 IL=2,4
      IF(IL.EQ.2) YQ(2,1)=YSTAR2
      IF(IL.EQ.2) XQ(2,1)=-1.50*PSAVE
      IF(IL.EQ.2) EQ(2,1)=0.0
      IF(IL.EQ.2) ZQ(2,1)=ZETA2
      IF(IL.EQ.2) DELX=(CHORX+3.*PSAVE)/(NETA0-1.0)
      IF(IL.EQ.2) IEND=NETA0
  
```

[illegible]

```

      B=ZCAM(IX+1)
      ZA=CLOG(PI/PITCH*(Z-A))
      ZB=CLOG(PI/PITCH*(Z-B))
      IF(XQ(IL,I)) .GT. 0.0) GO TO 190
      IF(YQ(IL,I)) .GE. 0.0) GO TO 200
      ZA=ZB
      ZB=ZB
      GO TO 210
200 CONTINUE
      IF(AIMAG(ZA) .LT. 0.0) ZA=ZA+2.0*AI*PI
      IF(AIMAG(ZB) .LT. 0.0) ZB=ZB+2.0*AI*PI
      GO TO 210
190 CONTINUE
      IF(XQ(IL,I)) .GT. XMAX) GO TO 220
      IF(YQ(IL,I)) .LT. Y1) GO TO 230
      IF(AIMAG(ZA) .LT. -PI/2.0) ZA=ZA+2.0*AI*PI
      IF(AIMAG(ZB) .LT. -PI/2.0) ZB=ZB+2.0*AI*PI
      GO TO 210
230 CONTINUE
      IF(YQ(IL,I)) .LE. Y2) GO TO 240
      IF(AIMAG(ZA) .LT. 0.0 .AND. IX .GE. IMAX) ZA=ZA+2.0*AI*PI
      IF(AIMAG(ZB) .LT. 0.0 .AND. IX .GE. IMAX) ZB=ZB+2.0*AI*PI
      GO TO 210
240 CONTINUE
      ZA=ZA
      ZB=ZB
      IF(IX .GE. IMAXY .AND. AIMAG(ZA) .GT. PI/2.0) ZA=ZA-2.0*PI*AI
      IF(IX .GE. IMAXY .AND. AIMAG(ZB) .GT. PI/2.0) ZB=ZB-2.0*PI*AI
      GO TO 210
220 CONTINUE
      ZA=ZA
      ZB=ZB
      CONTINUE
      SUM=SUM+STREN(IX)*(A-B+(Z-A)*ZA-(Z-B)*ZB)
      SUMD=SUMD+STREN(IX)*(ZA-ZB)
      ZO=(A+B)/2.0
      ZTEL=(Z-ZO)*PI/PITCH
      HTAN=(CEXP(ZTEL)-CEXP(-ZTEL))/(CEXP(ZTEL)+CEXP(-ZTEL))
      HSIN=0.5*(CEXP(ZTEL*2.0)-CEXP(-ZTEL*2.0))
      SUM=SUM+STREN(IX)*CLOG(HTAN/ZTEL)*(B-A)
      SUMD=SUMD+STREN(IX)*(PI/PSAVE/HSIN-1.0/(Z-ZO))*(B-A)
      CONTINUE
      ETAF=REAL(SUM)
      ZETAF=AIMAG(SUM)
      ZETAFD=AIMAG(SUMD)
      ZETAFD=-REAL(SUMD)
      F1=ETAO-ETAF
      G1=ZETA0-ZETAF
180

```

Q5048970
 Q5048980
 Q5048990
 Q5049000
 Q5049010
 Q5049020
 Q5049030
 Q5049040
 Q5049050
 Q5049060
 Q5049070
 Q5049080
 Q5049090
 Q5049100
 Q5049110
 Q5049120
 Q5049130
 Q5049140
 Q5049150
 Q5049160
 Q5049170
 Q5049180
 Q5049190
 Q5049200
 Q5049210
 Q5049220
 Q5049230
 Q5049240
 Q5049250
 Q5049260
 Q5049270
 Q5049280
 Q5049290
 Q5049300
 Q5049310
 Q5049320
 Q5049330
 Q5049340
 Q5049350
 Q5049360
 Q5049370
 Q5049380
 Q5049390
 Q5049400
 Q5049410
 Q5049420
 Q5049430
 Q5049440

```

DELY1=-F1/ETAFD
DELY2=-G1/ZETAFD
IF(IIL:EQ:2) DELT=DELY1
IF(IIL:GT:2) DELT=DELY2
YQ(IIL:1)=YQ(IIL:1)+DELT
YQ(IIL:1)=YQ(IIL:1).LT.1.E-06) GO TO 250
CONTINUE
CONTINUE
EQ(IIL:1)=0.0
EQ(IIL:1)=ZETAF
EQ(IIL:3)=EQ(IIL:1)=ETAF
EQ(IIL:3)=EQ(IIL:1)=0.0
EQ(IIL:4)=EQ(IIL:1)=ETAF
EQ(IIL:4)=EQ(IIL:1)=ZETA4
CONTINUE

```

120
250

110

```

NEND=2*NZGRID-1
DO 500 I=1,NEND
DO 500 J=1,NEGRID
ETA(I,J)=0.0
ZETA(I,J)=0.0
X(I,J)=0.0
Y(I,J)=0.0
CONTINUE
DELT=ZQ(4,1)/(NZGRID-1.0)
DELE=1.0/(NEGRID-1.0)
DO 510 I=1,NETAO
ARG(I)=ZQ(2,1)
FUN(I)=XQ(2,1)
GUN(I)=YQ(2,1)
CONTINUE
DO 520 J=1,NZGRID
ETA(J,1)=0.0
ZETA(J,1)=ZQ(4,1)-DELT*(J-1.0)
RQ=ZETA(J,1)
X(J,1)=FNTRP(ARG,FUN,RQ,NETAO)
Y(J,1)=FNTRP(ARG,GUN,RQ,NETAO)
CONTINUE
NEND=2*NZGRID-1
ISTART=NZGRID+1
DO 530 J=ISTART,NEND
IML=2*NZGRID-J
ETA(J,1)=0.0
ZETA(J,1)=-ZETA(IML,1)
X(J,1)=X(IML,1)

```

500

510

520

0050049450
0050049460
0050049470
0050049480
0050049490
0050049500
0050049510
0050049520
0050049530
0050049540
0050049550
0050049560
0050049570
0050049580
0050049590
0050049600
0050049610
0050049620
0050049630
0050049640
0050049650
0050049660
0050049670
0050049680
0050049690
0050049700
0050049710
0050049720
0050049730
0050049740
0050049750
0050049760
0050049770
0050049780
0050049790
0050049800
0050049810
0050049820
0050049830
0050049840
0050049850
0050049860
0050049870
0050049880
0050049890
0050049900
0050049910
0050049920

```

530      Y(J,1)=Y(IML,1)-PSAVE
        CONTINUE
        ILM=MUM+2
        DO 540 I=1,ILIM
          ARG(I)=ZQ(I,I)
          FUN(I)=XQ(I,I)
          GUNI(I)=YQ(I,I)
          CONTINUE
        DO 550 J=1,NEND
          ETA(J,NEGRID)=1.0
          ZETA(J,NEGRID)=ZETA(J,1)
          RQ=ZETA(J,NEGRID)
          X(J,NEGRID)=FNTRP(ARG,FUN,RQ,ILIM)
          Y(J,NEGRID)=FNTRP(ARG,GUNI,RQ,ILIM)
          CONTINUE
        DO 560 I=1,NZET4
          IML=NZET4+1-I
          ARG(I)=EQ(4,IML)
          FUN(I)=XQ(4,IML)
          GUNI(I)=YQ(4,IML)
          CONTINUE
        DO 570 I=1,NEGRID
          ETA(I,1)=(I-1.0)*DELE
          ZETA(I,1)=ZQ(4,1)
          RQ=ETA(I,1)
          X(I,1)=FNTRP(ARG,FUN,RQ,NZET4)
          Y(I,1)=FNTRP(ARG,GUNI,RQ,NZET4)
          CONTINUE
        DO 580 I=1,NEGRID
          ETA(NEND,I)=ETA(1,I)
          ZETA(NEND,I)=-ZETA(1,I)
          X(NEND,I)=X(1,I)
          Y(NEND,I)=Y(1,I)
          CONTINUE
        DO 585 I=1,NZETO
          IML=NZETO+1-I
          ARG(I)=EQ(3,IML)
          FUN(I)=XQ(3,IML)
          GUNI(I)=YQ(3,IML)
          CONTINUE
        DO 587 I=1,NEGRID
          ETA(NZGRID,I)=(I-1.0)*DELE
          ZETA(NZGRID,I)=0.0
          RQ=ETA(NZGRID,I)
          X(NZGRID,I)=FNTRP(ARG,FUN,RQ,NZETO)
          Y(NZGRID,I)=FNTRP(ARG,GUNI,RQ,NZETO)
          CONTINUE
        X(1,1)=-4.0*PSAVE

```

```

05049930
05049940
05049950
05049960
05049970
05049980
05049990
05050000
05050010
05050020
05050030
05050040
05050050
05050060
05050070
05050080
05050090
05050100
05050110
05050120
05050130
05050140
05050150
05050160
05050170
05050180
05050190
05050200
05050210
05050220
05050230
05050240
05050250
05050260
05050270
05050280
05050290
05050300
05050310
05050320
05050330
05050340
05050350
05050360
05050370
05050380
05050390
05050400

```

```

X(NZGRID,1)=4.0*PSAVE
X(NEND,1)=-4.0*PSAVE
Y(1,1)=YQ(4,NZET4)
Y(NZGRID,1)=YQ(3,NZET0)
Y(NEND,1)=YQ(4,NZET4)
L=NEGRID
L=NEND
L=NZGRID
C
C
C
GIVEN PERIODIC BOUNDARY AND BLADE SHAPE, SOLVE THE BOUNDARY VALUE
PROBLEM
FOR THE INTERNAL GRID POINTS.
BIJ=2.0*(1.0/(DELE**2))+1.0/(DELZ**2))
BIJP=1.0/(DELE**2)
BIJM=BIJP
BIPJ=1.0/(DELZ**2)
BIMJ=BIPJ
NEG2=NEGRID-1
NZG2=NEND-1
EPP=1.E-05
DO 600 ITER=1,100
ERR=0.0
DO 610 I=2,NZG2
DO 610 J=2,NEG2
IF(I.EQ.NZGRID) GO TO 610
XHOLD=X(I,J)
YHOLD=Y(I,J)
X(I,J)=C/BIJ*(BIPJ*X(I+1,J)+BIMJ*X(I-1,J)+BIJP*X(I,J+1)+BIJM*X(I
* J-1))+(-C)*X(I,J)
Y(I,J)=C/BIPJ*(BIPJ*Y(I+1,J)+BIMJ*Y(I-1,J)+BIJP*Y(I,J+1)+BIJM*Y(I
* J-1))+(-C)*Y(I,J)
ERR=ERR+ABS(X(I,J)-XHOLD)+ABS(Y(I,J)-YHOLD)
CONTINUE
610 IF(ITER.LT.5) GO TO 600
ERR=ERR/(2.0*NEG2*NZG2)
IF(ERR.LE.EPP) GO TO 670
KPRT=ITER/200
IF(ITER.NE.200*KPRT) GO TO 600
DO 620 I=1,NEND
CONTINUE
620 CONTINUE
600 CONTINUE
670 CONTINUE

```

```

4444 WRITE(6,4444) ITER,ERR
      FORMAT('  ITER COUNT FOR INTERNAL GRID POINTS= ',I3,' ERR= ',E15.5)
DO 650 I=1,NEND
DO 650 J=1,NEGRID
  ZPET(I,J)=X(I,J)+AI*Y(I,J)
  CONTINUE
650
MM=NEGRID
NN=NEND
K=1
  ZPET(1,1)=-2.0*CHORB+AI*YQ(4,NZET4)
  ZPET(NZGRID,1)=2.0*CHORB+AI*YQ(3,NZET4)
  ZPET(NEND,1)=-2.0*CHORB+AI*YQ(4,NZET4)
  ZPET(1,1)=-2.0*CHORX+0.
  ZPET(NEND,1)=ZPET(1,1)
  ZPET(NZGRID,1)=2.0*CHORX+0.
C
C
C
DO 675 I=1,NEND
DO 675 J=1,NEGRID
  ZPET(I,J)=ZPET(I,J)*CEXP(-AI*STAG*PI/180.0)
  X(I,J)=REAL(ZPET(I,J))
  Y(I,J)=AIMAG(ZPET(I,J))
  CONTINUE
  ETADF1=-DELZ
  ETADF2=-ETADF1
  ZETDF1=DELE
  ZETDF2=-ZETDF1
  ET(1)=FLOAT(NZGRID-1)*(-DELZ)
  ZET(1)=0.
DO 4844 I1=2,NEND
  ET(I1)=DELZ+ET(I1-1)
DO 4845 I1=2,NEGRID
  ZET(I1)=DELE+ZET(I1-1)
NOZ=NEND
DO 780 I=1,NEND
DO 780 J=1,NEGRID
  ZPET(I,J)=ZPET(I,J)*CEXP(AI*STAG*PI/180.0)
  CONTINUE
DO 790 I=1,NEND
  ZTEMP=(ZPET(1,NEGRID)+ZPET(I+1,NEGRID))/2.0
  ZTEMQ=ZTEMP*CEXP(-AI*STAG*PI/180.0)
  XT=REAL(ZTEMP)
  YT=AIMAG(ZTEMP)
  XR=REAL(ZTEMQ)
  YR=AIMAG(ZTEMQ)
  CONTINUE
790

```

```

C - CHOLESKY - SOLVES AN N BY N SYSTEM OF EQUATIONS WITH COEFFICIENT
C MATRIX A(I,J), CONSTANT VECTOR C(I), AND SOLUTION VECTOR.
C CHOLESKYS METHOD IS USED. IT REQUIRES N**2 + O(N) OPERATIONS.
C
C   DIMENSION A(63,63),C(63)
C
C - CALCULATION OF SOLUTION MATRIX. NOTE BOTH THE UPPER AND LOWER
C TRIANGULAR MATRICES ARE STORED AS REPLACEMENT VALUES IN MATRIX A.
C ALSO THE SOLUTION VECTOR REPLACES THE CONSTANT VECTOR (C).
C
C   IST = 1
C   JST = 2
C   DO 50 K = 1,N
C
C - CALCULATION OF LOWER TRIANGULAR MATRIX ELEMENTS
C
C   DO 10 I = IST,N
C     J = 0
C     J = J+1
C     IF (J.GE.K) GO TO 10
C     A(I,K) = A(I,K) - A(I,J)*A(J,K)
C     GO TO 20
C   CONTINUE
C
C - CALCULATION OF UPPER TRIANGULAR MATRIX ELEMENTS
C
C   IF (K.EQ.N) GO TO 50
C   DO 40 J = JST,N
C     I = 0
C     I = I+1
C     IF (I.GE.K) GO TO 40
C     A(K,J) = A(K,J)-A(K,I)*A(I,J)
C     GO TO 30
C   CONTINUE
C
C   A(K,J) = A(K,J)/A(K,K)
C   IST = JST+1
C   JST = JST+1
C
C - CALCULATION OF CONSTANT VECTOR ELEMENTS. ENTRY POINT FOR MULTIPLE
C SOLUTION INVOLVING THE SAME COEFFICIENT MATRIX, BUT CHANGES IN
C CONSTANT MATRIX.

```

051850
 051860
 051870
 051880
 051890
 051900
 051910
 051920
 051930
 051940
 051950
 051960
 051970
 051980
 051990
 052000
 052010
 052020
 052030
 052040
 052050
 052060
 052070

```

C      ENTRY CHGENT(A,C,N)
      NPI = N+1
      DO 70 K = 1,N
      I = 0
      IF (I-GE.K) GO TO 70
      C(K) = C(K) - A(K,I)*C(I)
      GO TO 60
      C(K) = C(K)/A(K,K)
C      - EVALUATION OF SOLUTION MATRIX
C      DO 90 K = 1,N
      L = NPI-K
      M = NPI
      M = M-I
      IF (M-LE.L) GO TO 90
      C(L) = C(L)-A(L,M)*C(M)
      GO TO 80
      CONTINUE
      RETURN
      END
C      60
C      70
C      80
C      90
  
```

LIST OF REFERENCES

1. NASA Report SP-36, Aerodynamic Design of Axial-Flow Compressors, edited by Irving A. Johnsen and Robert A. Bullock, 1965.
2. NACA Report 1016, Effect of Tunnel Configuration and Testing Technique on Cascade Performance, by John R. Erwin and James C. Emery, 1951.
3. Duval, David A., Evaluation of a Subsonic Cascade Wind Tunnel for Compressor Blade Testing, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1980.
4. Rose, C. and Guttormson, D. L., Installation and Test of a Rectilinear Cascade, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1964.
5. Bartocci, J. E., An Investigation of the Flow Conditions at the Lower Measuring Plane, and in the Plenum Chamber of the Rectilinear Cascade Test Facility, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1966.
6. Moebius, Richard C., Analysis and Testing to Improve the Flow from the Plenum of a Subsonic Cascade Wind Tunnel, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1980.
7. Cina, Frank S., Subsonic Cascade Wind Tunnel Tests Using a Compressor Configuration of DCA Blades, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1981.
8. NASA Technical Note TN D-5427, Fortran Program for Calculating Transonic Velocities on a Blade-to-Blade Stream Surface of a Turbomachine, by Theodore Katsanis, 1969.
9. NASA Handout, Draft of Technical Paper, Quasi-3D Full Potential Transonic Blade-to-Blade Code, by Charles Farrell, 1981, Conference on Turbomachinery Flow Analysis Methods--A Status Report on Maturing Codes, Conference at NASA Lewis Research Center, 14-15 October 1981.

10. NASA Technical Paper 1493, Performance of Two-Stage Fan Having Low-Aspect-Ratio First-Stage Rotor Blading, by Donald C. Urasek, William T. Gorrell and Walter S. Cunnan, 1979.
11. 3052A System Library (9845A), Hewlett-Packard Company, 1978.
12. Turbopropulsion Laboratory, Naval Postgraduate School, Technical Note 80-03, Data Acquisition Programs for the Subsonic Cascade Wind Tunnel, by D. A. Duval, August 1980.
13. W. R. Church Computer Center, Naval Postgraduate School, Technical Note VM-01, User's Guide to VM/CMS at NPS, October 1981.
14. W. R. Church Computer Center, Naval Postgraduate School, Technical Note VM-05, Introduction to the XEDIT EDITOR, May 1981.
15. NASA Technical Note TN D-8431, Revised Fortran Program for Calculating Velocities and Streamlines on the Hub-Shroud Midchannel Stream Surface of an Axial, Radial, or Mixed-Flow Turbomachine or Annular Duct, by Theodore Katsanis and William D. McNally, 1977.
16. Turbopropulsion Laboratory, Naval Postgraduate School, Project Report NPS67-81-019PR, Uniform Inlet Conditions for the NPS Subsonic Cascade Wind Tunnel, by Donald M. McEligot, December 1981.
17. Pankhurst, R. C. and Holder, D. W., Wind Tunnel Technique, Pitman and Sons, 1952.
18. NASA Technical Paper TP-2030, Computer Program for Calculating Full Potential Transonic Quasi-Three-Dimensional Flow Through a Rotating Turbomachinery Blade Row, by Charles Farrell, 1982.

INITIAL DISTRIBUTION LIST

	<u>No. Copies</u>
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4. Director, Turbopropulsion Laboratory, Code 67Sf Naval Postgraduate School Monterey, California 93940	15
5. Fan and Compressor Branch (ATTN: N. Sanger) Mail Stop 5-9 NASA Lewis Research Center 21000 Brookpark Road Cleveland, Ohio 44135	2
6. LCDR W. D. Molloy Jr. VC-5, NAS Cubi Point FPO San Francisco, California 96601	2