A GENERAL PROCEDURE FOR OBTAINING VELOCITY VECTOR FROM A SYSTEM--ETC(U)

JUL 79  D ADLER, R P SHREEVE

UNCLASSIFIED  NPS67-79-007
A General Procedure for Obtaining Velocity Vector from a System of High Response Impact Pressure Probes

D. Adler and R. P. Shreeve

July 1979

Approved for public release; distribution unlimited

Prepared for:
Naval Air Systems Command
Washington, DC

and

Office of Naval Research
Arlington, VA
The work reported herein was supported by the Naval Air Systems Command, Washington, DC., and Office of Naval Research, Arlington, VA.

Reproduction of all or part of this report is authorized.

This report was prepared by:

D. Adler, Visiting Professor of Aeronautics

R. P. Shreeve, Associate Professor of Aeronautics

Reviewed by:

Released by:

M. F. PLATZER, Chairman
Department of Aeronautics

W. M. TOLLES
Dean of Research
DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
A General Procedure for Obtaining Velocity Vector from a System of High Response Impact Pressure Probes

D. Adler and R. P. Shreeve

Naval Postgraduate School
Monterey, CA 93940

Naval Air Systems Command
Washington, DC 20361

Approved for Public Release; distribution unlimited.

Flow Measurements

A technique to measure a high frequency, repetitively pulsating flow field is presented. Two impact tube probes and a Kiel probe, all with pressure transducers mounted in their tips are used. Five readings are required to identify a velocity vector at a point.

In this report the technique, the numerical procedure and the computer program used are described.
A General Procedure for Obtaining Velocity Vector From a System of High Response Impact Pressure Probes

by

D. Adler and R. P. Shreeve
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notation</td>
<td>1</td>
</tr>
<tr>
<td>Notation of PENPTS</td>
<td>2</td>
</tr>
<tr>
<td>Notation of INTSCS</td>
<td>3</td>
</tr>
<tr>
<td>Notation of main program VDR</td>
<td>6</td>
</tr>
<tr>
<td>Introduction</td>
<td>10</td>
</tr>
<tr>
<td>Mathematical model</td>
<td>15</td>
</tr>
<tr>
<td>Evaluation of $\alpha$ and $\phi$ (method)</td>
<td>21</td>
</tr>
<tr>
<td>Evaluation of the intersection point coordinates (numerical procedures)</td>
<td>26</td>
</tr>
<tr>
<td>Evaluation of the velocity vector and pressures from the probe signals</td>
<td>43</td>
</tr>
<tr>
<td>Convergence and accuracy</td>
<td>48</td>
</tr>
<tr>
<td>Evaluation of the computation time</td>
<td>50</td>
</tr>
<tr>
<td>Input</td>
<td>51</td>
</tr>
<tr>
<td>Output</td>
<td>52</td>
</tr>
<tr>
<td>Conclusions</td>
<td>53</td>
</tr>
<tr>
<td>Appendix 1 (VDR)</td>
<td>56</td>
</tr>
<tr>
<td>Appendix 2 (SWVDR)</td>
<td>64</td>
</tr>
<tr>
<td>Appendix 3 (WVDR)</td>
<td>72</td>
</tr>
</tbody>
</table>
Notation

\( C_p \) - Pressure coefficient defined in eq. (3)

\( \alpha \) - Yaw angle relative to laboratory space

\( \phi \) - Pitch angle relative to laboratory space

\( P_t \) - Total pressure

\( P_s \) - Static Pressure

\( \alpha_{rp} \) - Yaw angle relative to probe axis

\( \phi_{rp} \) - Pitch angle relative to probe axis

\( P_p \) - Pressure indicated by the probe

\( \alpha_p \) - Yaw setting of the probe

\( \phi_p \) - Pitch setting of the probe

Subscripts

I \( \) relating to probe positions shown

II \( \) in figure 3

IV \( \)

A relating to A type probe

B relating to B type probe

Lo lowest value in an array

rp relative to the probe axis

Up highest value in an array
Notation of PENPTS

DD - denominator in the subsequent line of the program

HT - penetration height (Cp of a calibration surface)

IR - a flag with a value of either 1 or 2 to identify first or second penetration point. This flag also controls return of values to calling program or subroutine

IB - a flag indicating the location of the last rectangle in the band checked for a penetration point. It prevents repetition of the scanning in the 103 Do loop

NX - number of X values in the calibration surface grid (≈ values of the calibration surface)

NY - number of Y values in the calibration surface grid (≈ values of the calibration surface)

X(I) - X values in the grid

XM - value of X at point B (fig. 5)

XR(I) - two values of X returned by the subroutine as results

XS - currently calculated value of X which is the present result and is later stored in XR(I)

Y(J) - Y values in the calibration grid

YC - intermediate value used in the subsequent line of the program

YG - the Y position of the penetrating line*

Z(I,J) - Values of the Z = Cp calibration surface above grid points

ZX - value of Z either at point A or point C (see figs. 5, 6 and 7)

ZM - value of Z at point B

* It should be pointed out here that the penetrating line is parallel to the X axis at a height HT = Cp above the X,Y plane and at a distance YG from the X axis on the X,Y plane
Notation of INTSCS

AA - constant defined in eq. 8

AB - constant defined in eq. 9

ARIN - ordinate at which scanning starts

ARLO - lowest ordinate of calibration surface (as a condition it must be identical for all calibration surfaces used)

ARM1 - ordinate measured relative to probe 1 (the probe belonging to the first curve $C_p = \text{const}$.)

ARM2 - ordinate measured relative to probe 2 (the probe belonging to the second curve $C_p = \text{const}$)

ARP1 - setting ordinate of probe 1

ARP2 - setting ordinate of probe 2

ARRES (1) - abscissa value returned to INTSCS from PENPTS

ARUP - highest ordinate of calibration surface (as a condition it must be identical for all calibration surfaces used)

AR2R - ordinate relative to laboratory space

AR2RJ - ordinate relative to laboratory space of previous scan

BA - constant defined in eq. 8

BB - constant defined in eq. 9

DAR - ordinate step for scanning

HT1 - penetration height of calibration surface of probe 1 ($C_p$ of probe 1)

HT2 - penetration height of calibration surface of probe 2 ($C_p$ of probe 2)

IEPS - a flag which is equal to 2 when the calculation is carried out as shown in fig. 11, otherwise it is equal to 1. (This information is required in the main program and is returned to it)
ISL - a flag which is equal to 10 if one of the left hand side results returned from PENPTS is 1000.0 (i.e., is not a penetration point). Otherwise it is equal to 1

ISR - a flag which is equal to 10 if one of the right hand side results returned from PENPTS is 1000.0 (i.e., is not a penetration point). Otherwise it is equal to 1

K - a flag indicating whether the first or the second intersection is evaluated

N1 - number of abscissas in calibration surface grid (fig. 4)

N2 - number of Ordinates in calibration surface grid (fig. 4)

RES1(K) - evaluated and returned abscissa of intersection

RES2(K) - evaluated and returned ordinate of intersection

R1 - radius in fig. 11 for probe 1

R2 - radius in fig. 11 for probe 2

RAIL - abscissa of point AIL in fig. 9 or 10

RAIR - abscissa of point AIR in fig. 9 or 10

RAJL - abscissa of point AJL in fig. 9 or 10

RAJR - abscissa of point AJR in fig. 9 or 10

RBIL - abscissa of point BIL in fig. 9 or 10

RBIR - abscissa of point BIR in fig. 9 or 10

RBJL - abscissa of point BJL in fig. 9 or 10

RBJR - abscissa of point BJR in fig. 9 or 10

X1(I) - abscissas of calibration surface of probe 1 or its transformation as determined by calling statement in main program

X2(I) - abscissas of calibration surface of probe 2 or its transformation as determined by calling statement in main program
X3(I) - abscissas of calibration surface of probe 1 or its transformation as determined by calling statement in main program

X4(I) - abscissas of calibration surface of probe 2 or its transformation as determined by calling statement in main program

Y1(I) - ordinates matching X1(I)

Y2(I) - ordinates matching X2(I)

Y3(I) - ordinates matching X3(I)

Y4(I) - ordinates matching X4(I)

Z1(I,J) - values of $C_{Pr}$ for X1(I) and Y1(I)

Z2(I,J) - values of $C_{Pr}$ for X2(I) and Y2(I)

Z3(I,J) - values of $C_{Pr}$ for X3(I) and Y3(I)

Z4(I,J) - values of $C_{Pr}$ for X4(I) and Y4(I)
Notation of Main Program VDR

A - variables in library routine IXCLOK used to evaluate computation times
ALF - values of $\alpha$ returned to VDR from INTSCS
ALFA - $\alpha$ input when experiment simulation is carried out
ALFC - $\alpha_{rpIII}$
ALFCN - new value of $\alpha_{rpIII}$ for next iteration
ALFD - $\alpha_{rpIV}$
ALFDN - new value of $\alpha_{rpIV}$ for next iteration
ALF1 - $\alpha_I$
ALF2 - $\alpha_{II}$
ALF3 - $\alpha_I$
ALF4 - $\alpha_{III}$
AM - mach number
CPA - $C_{PrpI}$
CPB - $C_{PrpII}$
CPC - $C_{PrpIII}$
CPD - $C_{PrpIV}$
EPSPS - relative difference between present static pressure and its value in the previous iteration
EPSPSG - convergence criterion (on the static pressure)
FALF - final $\alpha$ value (at convergence)
FPHI - final $\phi$ value (at convergence)
IA50 - a flag in IXCLOK
IA60 - a flag in IXCLOK
ICOPS - number of PS corrections carried out to ensure the Cp's are in calibration range
IEPS - a flag governing the convergence criterion returned from INTSCS
IEPS1 - a flag governing the convergence criterion for probes I and II

IEPS2 - a flag governing the convergence criterion for probes I and III

IIT - number of iterations on $p_s$

ISCAN - number of static pressure scans from initial static pressure guess upwards

IXCLOK - system subroutine for computation time evaluation

NOCOPS - a flag which when equal to 1 causes static pressure corrections to be carried out such that Cp's are always inside calibration range, and when equal to 2 cause these corrections to be skipped

NOSIM - a flag which when equal to 1 causes the program to simulate velocity measurement experiments and when equal to 2 causes the program to reduce measured data

NX - number of calibration surface matrix abscissa's, $\alpha$ values (must be identical for all calibration surface matrices used)

NY - number of calibration surface matrix ordinates, $\phi$ values (must be identical for all calibration surface matrices used)

PDYN - dynamic pressure

PHI - value of $\phi$ returned to VDR from INTSCS

PHIB - $\phi_{pII}$

PHII - $\phi$ input when experiment simulation is carried out

PHI1 - $\phi_{pI}$

PHI2 - $\phi_{pII}$

PHI3 - $\phi_{pI}$

PHI4 - $\phi_{pIII}$

PPA - signal of probe in position I

PPB - signal of probe in position II
PPC - signal of probe in position III
PPC - signal of probe in position IV
PS - static pressure
PSIN - initial guess of static pressure
PSN - new value of static pressure for next iteration
PSC - corrected static pressure
PST - memorized corrected static pressure
PT - total pressure measured
RELXPS - relaxation factor for convergence on the static pressure
VIRTIM - virtual computation time
XA - abscissas of calibration surface matrix of probe A
XAX - transformed abscissas of calibration surface matrix of probe A
XB - abscissas of calibration surface matrix of probe B
XBX - transformed abscissas of calibration surface matrix of probe B
XPC - yaw setting of position III
XPD - yaw setting of position IV
XRIN - $\alpha$ value at which scanning starts
XRLU - lowest $\alpha$ value of calibration surface (as a condition it must be identical for probes A and B)
XRUP - highest $\alpha$ value of calibration surface (as a condition it must be identical for probes A and B)
YA - ordinates of calibration surface matrix of probe A
YAX - transformed ordinates of calibration surface matrix of probe A
YB - ordinates of calibration surface matrix of probe B
YBX - transformed ordinates of calibration surface matrix of probe B
YP A - pitch setting of probe A
YP B - pitch setting of probe B
YRIN - φ value at which scanning starts
YRLO - lowest φ value of calibration surface (as a condition it must be identical for probes A and B)
YRUP - highest φ value of calibration surface (as a condition it must be identical for probes A and B)
ZA - \( C_{rPa} \) values of probe A
ZAX - transformed \( C_{rPa} \) values of probe A
ZAMAX - maximum of ZA array
ZAMIN - minimum of ZA array
ZB - \( C_{rPb} \) values of probe B
ZBX - transformed \( C_{rPb} \) values of probe B
ZBMAX - maximum of ZB array
ZBMIN - minimum of ZB array
Introduction

Experimental knowledge of the flow field generated by rotating turbo impellers is of prime importance in the research and development of turbomachinery. It is essential for the refinement of design methods, for the development of new flow models which include secondary flow and tip clearance effects, and particularly for the verification of new computer codes developed to calculate the flow through rotating blade rows.

In recent years laser velocimeter techniques have been applied successfully to measure the flow both inside and downstream of rotors. (Ref. 1 for example). It has become clear however, that the laser techniques are only reliable in the hands of experienced investigators. A window which remains clean is essential, and seeding is usually required. Laser techniques do not measure the pressure field and usually can only measure two components of the velocity unless the axis of the laser is tilted. Difficulty is also encountered when measuring close to walls. Hence there are reasons to consider alternative techniques, particularly if they are simpler to apply routinely in stationary turbomachinery passages. Furthermore, the achievement of redundancy in measuring the flow field behind the impeller is itself a worthwhile goal. The present
work deals with the application of a particular system of small high response pressures probes at the exit of an impeller.

Measurements behind an impeller, in the stationary bladeless gap, are simpler to make than measurements within the rotating passages. Transducer probes can be installed through the stationary machine casing and the data transmitted without resort to slip rings or rotary transformers. The sensor is not subjected to the centrifugal field or to the vibration of the rotor. However, the flow to be measured is then fluctuating at blade-passing frequency and any system of sensors must be calibrated for a wide range of possible Mach number, pitch angle, yaw angle and pressure variation - and yet must be capable of the necessary frequency response.

In Ref. 2, a method was described for using two semiconductor pressure probes together with the technique of synchronized sampling, to obtain the distribution of the velocity vector downstream of a rotor. The geometries of the two probes, designated Type A and Type B, and their installation in the compressor annulus are shown in Fig 1. It was argued that, in principle, by rotating the probes in yaw about their tips and controlling the sampling of the data from each probe to be at the same position in the rotor frame, the system of two separate probes could be used to acquire data at a point in the periodic flow
FIGURE 1. TYPE A AND TYPE B PROBES AND COMPRESSOR INSTALLATION.
from the rotor, corresponding to data normally obtained from the multiple sensors of four- or five hole pneumatic probes when measuring velocity in steady flows. The technique promised the use of probes having the simplest geometry and thus avoided the large size, expense and unreliability of multiple sensor probes which incorporate multiple semi-conductor transducers (Ref. 3). Because of the simple sensor tip geometry (that of a cylinder at incidence to the flow) the unsteady response was likely to be as good as could be expected of any single physical sensor.

The two-probe technique is strictly applicable only to periodic flows. However, data obtained on successive rotations of the rotor can be averaged to eliminate fluctuations which are not periodic. This was shown to be effective in tests reported in Ref. 2 in which a single Type A probe was used to establish the peripheral blade-to-blade distribution of flow yaw angle.

In order to obtain velocity from the pressure measurements which can be obtained from the two probes, the steady response characteristics must first be established in calibration tests carried out in a known, controlled, uniform flow. Second, a method of applying the calibration to measurements made in an unknown flow must be devised. In the present method two different approaches have been followed. In Ref. 4, a technique is given for representing
and applying the probe calibration analytically. When first applied the method gave surprisingly good accuracy (1 - 2%) since the method required that the probes had characteristics which could be well represented analytically. This in turn required that the probe tips be geometrically precise, a feature which was not present in the first generation of probes.

A second, more general approach is reported here, wherein the calibration of each probe is represented by a two-dimensional array of pressure coefficients. The application of the calibration given in this form, in an unknown flow required the development of special numerical procedures. The purpose of the present report is to document the analysis and the Fortran program developed to apply the method.

In its present form, the method does not require that the calibration "Surfaces" be symmetrical about any axis or be expressed in analytical form, but does require that the pressure coefficient be independent of Mach number. The latter restriction could undoubtedly be removed by introducing additional iterative steps. Further, in the present method only five measurements have to be taken to determine uniquely a velocity vector at a point. Throughout the report, the Fortran program notation has been used to describe the physics and equations involved in the solution.
Mathematical model

Assume the A and B type probes of ref. [1] (see also fig. 1) to be immersed in a three-dimensional steady flow field.* The pressure response of each of these probes in given gas is functionally described by four variables as:

\[ P_p = P_{p}(\alpha, \phi, P_T, P_S) \]  \hspace{1cm} (1)

If a pressure coefficient is defined as

\[ C_{p_{rp}} = \frac{P_p - P_S}{P_T - P_S} \]  \hspace{1cm} (2)

The calibration surface of each probe is given in the general case in form of a matrix of values of \( C_p \), where

\[ C_{p_{rp}} = C_{p_{rp}}(\alpha_{rp}, \phi_{rp}) \]  \hspace{1cm} (3)

The pressure coefficient defined in this way has only a second order dependence on Mach and Reynolds numbers in the range of

* For our purpose, using the synchronized sampling, the flow field behind the impeller is steady, although the probes require a high speed response because of fluctuations.
$0 < \text{Ma} < 0.7$ in turbulent flows (Ref. 2), so that, to first order their influence is neglected in writing eq. (3).

If the type A probe is rotated about its axis into three different positions ($i = I, III, IV$) readings are taken, and the type B probe is fixed in position II and a single reading is taken, the following four equations can be written.

\[
\begin{align*}
CP_{Ai} &= \frac{P_{pAi} - PS}{PT - PS} = CP_{Ai}(\alpha, \phi) \\
CP_{BII} &= \frac{P_{pBII} - PS}{PT - PS} = CP_{BII}(\alpha, \phi)
\end{align*}
\]  

(4)

it should be pointed out here, to avoid misunderstanding that $\alpha$ and $\phi$ are defined in a coordinate system relative to the machine axis and not relative to the probe axis. In the set of four equations (4) there are four unknown quantities, namely: $\alpha, \phi, PT$ and $PS$. These are the quantities to be evaluated using the measured data. Together with the stagnation temperature they define the flow field uniquely. The four equations, resulting from the four measurements, should be sufficient to determine the four unknown quantities.

But the problem is complicated by three facts:

1) The calibration surfaces are not generally known in analytical form.
2) The calibration surfaces are double valued in $\alpha$ and $\phi$ i.e., for a given $Cp$ and $\alpha$ there exist two $\phi$ values, or for a given $Cp$ and $\phi$ there are two $\alpha$ values which satisfy eq. 3.
3) As a result of the measurement the value of $P_p$ rather than $C_p$ is determined.

Since the calibration surfaces are not given in a simple analytical form the solution has to be numerical. An iterative procedure is required because $P_p$ and not $C_p$ values are measured. First $P_s$ and $P_T$ have to be guessed to yield $C_p$ values, knowing the four measured values of $P_p$, and using eq. 4. The guess is then iteratively corrected to converge on the solution. However, convergence of the iterative procedure is complicated because of the double valued nature of the calibration surfaces.

This method of iteration shown in Fig. 2 was attempted initially for the evaluation of a measured point. In practice the initially suggested procedure converged in some cases and diverged in others, depending on the values, and signs of $\alpha$ and $\phi$. This was not surprising as convergence on two variables is not likely to be a simple matter. However, in order that the measurement technique be useful, convergence on the correct solution for a general set of measurements, is absolutely necessary. In practice, this can certainly be achieved if one of the two iteration variables is obtained by measurement. Since the static pressure is a difficult quantity to measure even in a steady flow field, only the stagnation pressure measurement need be considered. It is possible that the time-varying stagnation pressure could be measured with a suitably designed Kiel probe. Data would then be taken by synchronized sampling from the fixed Kiel probe, from the type A probe rotated into two positions, and from the
GUESS 
$P_T$ & $P_S$

COMPUTE 
FOUR $C_P$ VALUES 
FROM FOUR $P_P$ VALUES

EVALUATE 
$\alpha$ & $\phi$

CALCULATE 
$P_T$ & $P_S$

CHECK 
FOR CONVERGENCE 
OF $P_T$ & $P_S$ 
SIMULTANEOUSLY

CONVERGED? 
NO YES 
EXIT

Fig. 2: Unstable iterative solution using measurement of A type and B type probes only.
fixed type B probe. The method of solution is then as shown in Fig. 3. The method shown in Fig. 3 proved to converge under all conditions. It is described in detail in the following pages.
Fig. 3: Stable iterative solution using measurements of A type and B type probes as well as measurements of a Kiel probe.
Evaluation of $\alpha$ and $\phi$ (method)

The best way to understand the evaluation of yaw and pitch is to look at it from a topographical point of view. Each of the type A and type B probes has a unique calibration surface $C_{\text{Prp}} = C_{\text{Prp}}(\alpha_{\text{rp}}, \phi_{\text{rp}})$, where $\alpha_{\text{rp}}$ and $\phi_{\text{rp}}$ are measured relative to the axis of symmetry through the sensor at the probe tip. The calibration surface in this representation is invariant to yaw and pitch of the probe axis relative to the laboratory space. The same calibration surface if represented in the form $C_p = C_p(\alpha, \phi)$ can be written as

$$C_p = C_p[(\alpha_p + \alpha_{\text{rp}}), (\phi_p + \phi_{\text{rp}})]$$

where $\alpha_p$ and $\phi_p$ are the probe tip axis angular settings relative to the laboratory space. It is clear from eq. (5) that $C_p$ can be derived from $C_{\text{Prp}}$ by a constant translation: $\alpha_p$, $\phi_p$ on the $\alpha, \phi$ plane. As each $C_{\text{Prp}}$ can be viewed as a hill with its peak at $\alpha_{\text{rp}} = 0$ and $\phi_{\text{rp}} = 0$. The $C_p$ surfaces are the same hills with their peaks translated to $\alpha = \alpha_p$ and $\phi = \phi_p$.

In the present method probe A is used in three different angular settings namely:

1) $\alpha_p = 0$, $\phi_p = 0$
III) $\alpha_p = \alpha_{pIII} \quad \phi_p = 0$

IV) $\alpha_p = \alpha_{pIV} \quad \phi_p = 0$

and the B probe is used in a fixed position, namely:

II) $\alpha_p = 0 \quad \phi_p = \phi_{pII}$

For this case the topography of the calibration surfaces will appear as four hills. The three with their peaks at points $(0,0); (+\alpha_{pIII}, 0)$ and $(-\alpha_{pIV}, 0)$ are the translated hills $Cp_{pA}$ and the fourth with its peak at $(0, \phi_{pII})$ is the translated hill $Cp_{pB}$. Their contours of constant $Cp$ are then as shown in figure 4.

Assume now that a velocity vector with yaw $\alpha$ and pitch $\phi$ is to be measured. These values of $\alpha$ and $\phi$ will be sensed uniquely by the probes at their four angular settings. Were equation (4) single valued, the values of $\alpha$ and $\phi$ could be uniquely evaluated, as the single intersection point between the projections of appropriate lines of constant $Cp$ on the $\alpha, \phi$ plane*.

In the present case of the double valued functions, the lines of constant $Cp$ are closed curves and more than a single intersection point do exist.

* $Cp$ has generally a different value for each probe in each of its settings. Thus the solution involves solving for the intersection points between projections of contours of specified (but different) $Cp$ values on the different hills.
Fig. 4: Projections of $C_p = \text{const.}$ lines of the four calibration surfaces (the center hill is for probe A at $\alpha_p = 0$, $\phi_p = 0$, the top hill is for probe B at $\alpha_p = 0$, $\phi_p = \phi_{\Pi}$, the right hill is for probe A at $\alpha_p = +\alpha_{\Pi}$, $\phi_p = 0$ and the left hill is for probe A at $\alpha_p = \alpha_{\Pi}$, $\phi_p = 0$.}
The situation is shown in Fig. 5 for an example of a number of such intersection points. The correct intersection point, however, is uniquely identified as the only point through which all four $C_p = \text{const.}$ curves pass.

From an examination of Fig. 5 it is clear that the intersection points of three closed $C_p$ curves projections only are sufficient to identify $\alpha$ and $\phi$ uniquely. However one of these three curves must be that belonging to the type B probe.

The details of the numerical procedure used to obtain the correct intersection point in the evaluation of $\alpha$ and $\phi$, are given in the following paragraph.
Fig. 5: An example showing a number of intersection points. The real one is in the second quadrant.
Evaluation of the intersection point coordinates (numerical Procedure)

The calibration surfaces \( C_{rp} \) are represented in form of a linear string of values ordered into an array, as shown schematically in figure 6. The numbers in Fig. 6 indicate the position of a \( C_{rp} \) value in the string. The string starts with the value of \( C_{rp} \) at a point \((-\alpha_{rpLo}; -\phi_{rpLo})\) and ends with the value of \( C_{rp} \) at a point \((+\alpha_{rpUp}; +\phi_{rpUp})\). This sequence must be kept and can not be changed.

Each of the closed \( C_p = \text{const.} \) curves projections in figs. 2 and 3 is the projection of the line of intersection between a plane parallel to the \( \alpha,\phi \) plane at a height equal to \( C_p \) and the calibration surface. These closed curves can be determined as the locus of the projections of all the penetration points of arbitrary lines parallel to the \( \alpha,\phi \) plane at a height \( C_p \) and the calibration surface. Such penetration points are calculated in subroutine "PENPTS", Fig. 10.

PENPTS calculates the first two penetration points of a surface \( Z \) (in the present case \( Z = C_p \)) by a straight line piercing that surface. If the surface is double valued these two points are the only roots. The surface is given as a table of numbers on a cartesian basis \( Z = Z(X,Y) \) (or in the present case \( C_p = C_p(\alpha,\phi) \).

The subroutine has the following limitations:

1) No roots can be found on the lower \( Y = \text{const.} \) boundary
Fig. 6: The order of a calibration surface array.
2) The roots can be found only along a piercing line which is parallel to the X axis (but at any height above the X,Y plane)

3) The table \( Z = Z(X,Y) \) defining the surface must be based on a grid comprising lines \( X = \text{const.} \) and \( Y = \text{const.} \). The spacing of these lines must not be equal. In other words the surface is defined by a rectangular grid in the X,Y plane, from \( X_{\text{min}} \) to \( X_{\text{max}} \) and from \( Y_{\text{min}} \) to \( Y_{\text{max}} \).

4) Only a single root can be evaluated in a surface element located above a defining rectangle on the X,Y plane.

5) Not more than the first two roots will be evaluated for any piercing line.

6) The surface must be monotonic over each rectangle (this is a result of limitation 4).

7) All X and Y arguments must be given in increasing order.

These limitations do not restrict the application of PENPTS in the present problem as long as the calibration surfaces are smooth within the element located above a grid rectangle. However, the elemental grid rectangles can be reduced arbitrarily in size. If a calibration surface is more than double valued PENPTS will fail. However in this case the probe yielding such a calibration surface can not be considered a useful instrument unless used only in parts of the domain where it is double valued.
Z is defined from $X_{\text{min}}$ to $X_{\text{max}}$ and from $Y_{\text{min}}$ to $Y_{\text{max}}$.

PENPTS is given the following initial information: $X$ values and $Y$ values defining grid points, corresponding $Z$ values, the $Y$ location ($YG$) and the height above the $X,Y$ plane ($HT$) of the piercing line. With this information PENPTS searches for the band of rectangles which includes $YG$ or of which $YG$ is the lower boundary (see fig. 7) and then scans this band from left to right in search for penetration points. The scanning is based on the geometry given in fig. 7 which represents a particular element in the band, approximated by two plane triangles.

Initially a coarse scan is carried out just to detect, but not to evaluate, an intersection point. This is done checking for each sub-domain whether \[
\frac{(ZX(I-1) - HT)}{(ZX(I) - HT)} > 0.
\]
If this condition is true a root is detected and control is transferred to its exact evaluation. The value of the root is calculated after its location, either in the first (left) or second (right) triangle is determined (each grid rectangle is composed of two triangles). Equation 6 which is based on fig. 8 (for a left triangle) or eq. 7 which is based on fig. 9 (for a right triangle) is used to evaluate the penetration point. These equations express the linear interpolation of $C_p$ in the Fortran rotation used in this program.

\[
XS = X(I-1) + \frac{\text{ABS}(ZX(I-1)-HT) \times (XM-X(I-1))}{\text{ABS}(HT-ZM) + \text{ABS}(ZX(I-1)-HT))}
\]

(6)
Fig. 7: The geometry of a linearized calibration surface element comprising two plane triangles and its intersection with a plane normal to \( X,Y \) along \( Y = Y_G \).
Fig. 8: The geometry for a penetration through a left hand (first) triangle.
Fig. 9: The geometry for a penetration through a right hand (second) triangle.
\[ XS = XM + (X(I) - XM) \times \frac{ABS(ZM - HT)}{ABS(ZM - HT) + ABS(HT - Z(I))} \] (7)

Equations 6 and 7 are invariant to the slope of the calibration surface i.e., the slope of the straight lines AB and BC.

Finally it should be pointed out that when two penetration points are determined the values of their abscissas, \( X(I) \), are returned. When only a single penetration point is detected the abscissa \( X(2) \) will be returned with a value of 1000.0. When no penetration point is determined both \( X(I) \) values returned will have the value of 1000.0. The program logic is designed to recognize these messages.

It was stated earlier that PENPTS is used to determine the closed intersection curves projections on the X,Y plane. In fact not the curves but just the intersection points between each two of them are required (see figure 3).

To compute the coordinates of these points the subroutine "INTSCS" is used. It uses PENPTS as a subroutine.

In INTSCS a scanning procedure is carried out from a minimal value of \( Y \) (or \( \phi \)) YRIN to an upper value of YRUP, or a
prescribed 10,000 times* which ever comes earlier. The subroutine scans through any two arbitrarily chosen closed curve projections to find their intersection points. In each scan (I) up to four penetration points can be determined, while the penetration points of the previous scan (J) are memorized. Together, eight penetration points can be involved. When no intersection point exists the geometrical situation is as shown in figure 11, while the existence of an intersection point is characterized in figure 12. Subroutine INTSCS can distinguish between the two situations. In figures 11 and 12 the case of four penetration points found in each scan are shown. The subroutine, however will handle any possible number of such points, from zero to four. A "no penetration points" is assigned an abscissa value of 1000.0 by PENPTS as explained earlier.

When an intersection point is detected its evaluation is based on the geometry shown in fig. 12. The eight penetration points have the following coordinates:

\[
\begin{align*}
AJL(RAJL, YRJ) & \quad BJL(RBJL, YRJ) \\
BJR(RBJR, YRJ) & \quad AJR(RAJR, YRJ) \\
BIL(RBIL, YR) & \quad AIL(RAIL, YR) \\
AIR(RAIR, YR) & \quad BIR(RBIR, YR)
\end{align*}
\]

* The 10,000 scans are governed by a program constant in the line "DO 140 ...." and can be arbitrarily varied.
Fig. 11: The geometry for two successive scans when no intersection point exists.
Fig. 12: The geometry for two successive scans when an intersection point does exist.
The left intersection point illustrated in fig. 12 is the intersection point of the lines aa and bb, each described by its equation:

for aa \( Y = (AA)X + BA \) \( (8) \)

for bb \( Y = (AB)X + BB \) \( (9) \)

The constants AA, BA, AB and BB are given in equations 10 to 13.

\[ AA = \frac{YR - YR}{RAJL - RAIL} \] \( (10) \)

\[ BA = YR - AA \times RAIL \] \( (11) \)

\[ AB = \frac{YR - YR}{RBIL - RBIL} \] \( (12) \)

\[ BB = YR - AB \times RBIL \] \( (13) \)

The coordinates of the intersection point to be calculated are

\[ X = \frac{BB - BA}{AA - AB} \] \( (14) \)

\[ Y = X \times AA + BA \] \( (15) \)

These relations are true for a left hand intersection point. Analogous equations are true for a right hand intersection point. In this algorithm the straight lines aa and bb approximate the curved lines connecting AJL with AIL and BJL with BIL or similar lines on the right hand side of figure 12. The error introduced through this approximation is reduced as \( DAR = \Delta \phi \) is reduced.

It is possible that an intersection point is identical with AIL and BIL or AIR and BIR. This case is defined as "direct hit". The program is designed to detect such a direct hit and evaluate the corresponding intersection point accordingly.

The above algorithm works perfectly as long as the two closed \( Cp = \text{const. curves} \) are far from being tangent. But in
practice a situation of almost tangent curves can arise when $X$ (or $a$) is very small. In this case the preceding algorithm will fail and must be replaced. The geometry of this situation is described in figure 13. This situation is identified by INTSCS and the intersection points are then evaluated assuming that they are intersections of two circular arcs. When the curves $C_p = \text{const.}$ are almost circular this approximation does not lead to unacceptable errors.

It was stated earlier that INTSCS scans from a minimum value of $\phi$ to a maximum value of $\phi$ with prescribed steps $\Delta \phi$, as shown in figs. 11 and 12. This direction of the scanning is used when the intersection points of the curves $C_{pI}$ and $C_{pII}$ of fig. 5 are evaluated.

However, in course of the reduction of the measured data, scans in the direction of $a$ in steps of $\Delta a$ are also necessary to evaluate the intersections of the curves $C_{pI}$ and $C_{pIII}$ or $C_{pI}$ and $C_{pIV}$. INTSCS is designed to carry out this task as well. To do this the calling statement for INTSCS is appropriately changed as will be explained in the next section. To be general enough INTSCS is not written in terms of $a$ and $\phi$ or $X$ and $Y$ but rather in terms of general arguments. The best way to understand INTSCS is to compare its general arguments to physical quantities by means of the calling statements. In fig. 14 the flow diagram in INTSCS is given in terms of the general arguments.

INTSCS returns to the main program (fig. 15) the $(a, \phi)$ coordinates of two intersection points between the $C_p = \text{const.}$ curves specified in the calling statement. Let us now follow the way in which the main program is designed to utilize INTSCS.
Fig. 13: The geometry when two closed $C_p = \text{const.}$ curves are almost tangent.
FIGURE 14. FLOW DIAGRAM OF INTSCS
for the evaluation of $\alpha$ and $\phi$ of the velocity vector, as well as static and total pressures.
Evaluation of the Velocity Vector and Pressures from the Probe Signals

Fortran program VDR was written to evaluate the velocity vector from probe measurements of pressures. The program is shown in Fig. 15.

At line 1410 INTSCS is called to scan curves I and II for possible intersection points. Scanning can be carried out in the direction of the ordinate only, with the calibration curve matrices compiled exactly as shown in fig. 6. This limitation is imposed by the way PENPTS is constructed. In the case of the intersection points between I and II ZA and ZB are scanned without difficulties in the direction of the ordinate which is $\phi$ as shown in fig. 12 and returns with the coordinates of the two first intersection points, points 1 and 2 of Fig. 16. They are ALFI, PHI1 and ALF2, PHI2. In line 1500 INTSCS is called again to scan curves I and III. Now scanning has to be carried out in the direction of the abscissa, a task for which INTSCS was not designed. To overcome this problem the calibration curve matrices are used in a transformed form such that the previous abscissas are now ordinates, ordinates are abscissas and the internal structures of the $C_{p_r}$ arrays are accordingly modified. This transformation is carried in VDR in the section between lines 320 and 450. Comparison of the calling line 1500 to the previous calling line 1410 shows very clearly how the various arrays: original and transformed, are used. The coordinates of points 3 and 4 of fig. 16
Fig. 16: Selection of the proper intersection point in VDR.
are now returned to the main program of VDR, they are ALF3, PHI3 and ALF4, PHI4.

We are seeking a single intersection point; the one representing yaw and pitch as sensed by probe A in positions I and III and probe B in position II. But, as the calibration surfaces are double valued we are now, unfortunately, in the possession of four points. The solution, however, is physically unique. Only a single velocity vector exists in reality and its yaw and pitch are included in the four intersection points evaluated. Were both the measurements and the numerical procedure absolutely accurate, two of the four points would have been identical. But this is not the case in reality, instead of a single point, two points close to each other will be detected. Therefore the average coordinates of the two of the four intersection points returned to VDR from INTSCS which are closest to each other are selected as the measured yaw and pitch angles. In the example shown in fig. 16 this will be the point dividing the distance between points 2 and 3.

The calculation is now at the point at which $\alpha$ and $\phi$ are temporarily known (see fig. 3). Using $\alpha$ and $\phi$ a new $C_{p_{PIV}}$ (for probe A in position IV) is computed by linear interpolation using subroutine INTPLT. With this new value of $C_{p_{PIV}}$ and with $P_{PIV}$ a new $P_s$ is computed. If this new value of $P_s$ is close enough as determined by EPSPSG to the guessed value, or to the value of $P_s$ in the previous iteration, the data reduction for the particular point in question is terminated.
The relative difference between present and previous $P_s$ is compared to the convergence criterion EPSPSG. This criterion is evaluated by an empirical function determined to give best compromise between accuracy of results and computer time required until convergence is achieved. When the calculation follows normal routine the convergence criterion is given in line 1710 as function of $P_T$. When the routine for almost tangent curves is used during data reduction either on the right or the left side a different empirical function (lines 1720 or 1730) is used.

If the convergence criterion is satisfied results are printed out and data for a new measurement point is read for reduction. If, however convergence is not reached a new static pressure for a next iteration is evaluated (line 1780).
Convergence and accuracy

The convergence of the present iterative procedure is not ensured in all possible cases of data sets. The program, however, is adjusted to converge in most of the cases. Similarly to other iterative computation method some experience is required to achieve convergence when the calculation does not converge. In this paragraph the principal factors affecting convergence, computation time and accuracy, which are obviously coupled, are pointed out.

The convergence and accuracy of reduction of a data set: $P_T$, PPA, PPB, PPC, PPD, PSIN (or analogous set in the experiment simulation mode) depends on the following factors, which can be varied by the user.

1) Coarseness of calibration arrays $Z_A$, $Z_B$ and their linearity. With coarser arrays convergence problem will increase and accuracy of results decrease.

2) Probe settings $Y_PB$, $X_PC$, $X_PD$ (with the following probe settings being fixed and not variable $Y_PA = Y_PC = Y_PD = 0.0$; $X_PA = X_PB = 0.0$). The optimum setting is about $+20^\circ$ to $25^\circ$. Too small values reduce accuracy, too big values cause convergence problems and probe tip flow separation.

3) RELXPS, the static pressure relaxation factor. The smaller this factor the safer will convergence be achieved. Computation time, however, will increase.

4) The constant 5.0 in line 1050 of the program. The dimension of this constant is kg/m$^2$. If too big results can be lost and if too small computer time will be growing. The variation of this constant should be coupled with an appropriate modification of the constant in line 1010 of VDR (item 8 in this list).
5) The constants in the evaluation of EPSPSG (lines 1710 to 1730). The smaller EPSPSG the more accurate the results in expense of increased computation time and reduced convergence safety.

6) The constant scanning step DAR. The smaller DAR the more accurate the results, but too small values can cause complete loss of results. Computation time increases with reduced DAR.

7) The constant 10 in line 1360. This constant governs the number of $P_s$ corrections.

8) The constant 1000 in line 1010. This constant governs the number of scans.

If convergence is not achieved in a particular case variation of each or of a combination of the above values will always enable convergence.
Evaluation of the Computation Time

By deleting the letters CT from column 1 and 2 of lines 20 to 70, 1930 to 1960, 1980, 2000, 2010 the actual computation time as well as CPU time are evaluated and printed out. The following statement prior to execution is required in this case:

GLOBAL T SYSLIB SSPLIB

This option is useful for adjusting the constants affecting convergence as to optimal compromise between accuracy, ease of convergence and calculation costs.

* When the program is run on Naval Postgraduate School IBM 360 system.
Input

1) Calibration arrays have to be input in the following manner: first values of NX and NNY, second values of X in rising order, third values of Y in rising order and fourth values of Z in the order shown in Fig. 6. Two calibration arrays are read: first the array of type A probe and second the array of type B probe.

2) If the program is run in experiment simulation mode PT and PS are read in Kg/m² and subsequently o and φ.

3) If the program is run in data reduction mode the measured values of PT, PPA, PPB, PPC and PPD are read and then the guess of static pressure PSIN. All are read in Kg/m². PSIN has to be lower than the actually existing value to ensure convergence of the calculation. Too low a value will cause waste of computer time.

Calibration arrays are read from a disk space on which they are stored. The following statement prior to execution is required for successful reading:

    FILEDEF 02 DSK NAME XX

here NAME is the name of the file on which the calibration data of probes A and B is stored in proper order and format. XX is a two digit number.

Simulation or measurements reduction input is read in the normal way using the terminal keyboard or punched cards.
Output

\( \alpha, \phi, P_s, P_T \) and \( Ma \) are printed out according to lines 1820 to 1920. This output, however, is not sufficient when the logic of the computation is to be followed either to examine the execution of the program or for debugging. Two programs with additional output are given in appendices 2 and 3 and can be used for this purpose. The first, SWVDR gives short additional output, namely:

1) When \( PSIN \) is successively increased automatically by the program to ensure convergence, the values of \( PSIN \) are printed out.

2) \( IIT, FALF, FPHI \) and \( PSN \) are printed out at the end of each iteration prior to the convergence test.

The second program WVDR prints more detailed information. All additional WRITE statements in this program are numbered with three digit numbers starting with 9.
Conclusions

The following conclusions are drawn on the basis of experience gained in running the program with various data sets and various types of calibration surfaces.

1) Optimal probe settings for ease of convergence and high accuracy are: YPB = 250°, XPC = 250°, XPD = -250°. It is therefore suggested that a B type probe with 250° pitch will be used, and that the A type probe be rotated to + 250°.

2) Convergence and accuracy, as well as computer time efficiency are improved when the calibration surfaces of both probes are not flat at their peaks but are rather rounded. It is therefore suggested that a new probe tip geometry be considered. A spherical tip with a central pressure tap is recommended. To prevent damage to the sensitive transducer located behind the pressure tap, and in order to improve the frequency response of the probes it is suggested that the volume between the pressure tap face and the transducer be filled with an appropriate liquid that will not affect the transducer negatively. In this case the opening of the pressure tap has to be sealed with a very thin low inertia membrane.

3) It is probably possible to modify the iterative procedure such that safe convergence can be achieved also when using the scheme of Fig. 2. If this can be achieved the Kiel probe will not be necessary. An effort in this direction is suggested.
References


FILE: VJR FORTRAN P1

WRITE(*,400) PS
READ(*,411) ALFA
WRITE(*,411) ALFA
READ(*,421) PHII
WRITE(*,421) PHII
CALL 1,IPFLT(XA,YA,ZA,ALFA,PHII,NX,NY,CRA)
ALP2=ALP-10C
CALL 1,IPFLT(XA,YA,ZA,ALFC,PHII,NX,NY,CPC)
ALF2=ALF-10C
CALL 1,IPFLT(XA,YA,ZA,ALFE,PHII,NX,NY,CPE)
PHII=PHII-YPD
CALL 1,IPFLT(XA,YA,ZA,ALFO,PHII,NX,NY,CPO)
PS=PS+10P
PPA=CPP+10P
PPB=CPP+10P
PPD=CPP10P
C READ MEASUREMENTS DATA
IF(NAXA.LE.I400) GO TO 501
READ(*,321) PS
WRITE(*,321) PS
READ(*,331) PS
WRITE(*,331) PS
READ(*,341) PS
WRITE(*,341) PS
READ(*,351) PS
WRITE(*,351) PS
IF(SCAN.EQ.0) GO TO 181
IF(SCAN.EQ.11) GO TO 181
IP=1
ICURVE=1
IF(SCAN.EQ.1) GO TO 181
IF(SCAN.EQ.11) GO TO 181
PS=PS1
CPP=CPP1
C CALCULATE PRESSURE COEFFICIENTS
POY=PT-PS
CPP=(CPP-PS)/POY
CPP=CPP+PS
CPP=CPP+PS
C CORRECTS PS ASSUMPTICA TO ENSURE CR ARE IN CALIBRATION RANGE
IF(NAXA.EQ.1) GO TO 301
ZAMAX=-10000
CALL MAX(2,4,ZA,ZAMAX,NX,NY)
IF(CP2.EQ.0.1A) PS=1.0P
IF(CP2.EQ.1.04AX) PS=1.0P
IF(CP2.EQ.1.2AX) PS=1.0P
ZAMAX=-10000
CALL MAX(2,4,ZA,ZAMAX,NX,NY)
IF(CP2.EQ.0.1A) PS=1.0P
ZAMAX=-10000
CALL MAX(2,4,ZA,ZAMAX,NX,NY)
IF(CP2.EQ.0.1A) PS=1.0P
CALL MAX(2,4,ZA,ZAMAX,NX,NY)
IF(CP2.EQ.0.1A) PS=1.0P
IF(CP2.EQ.0.1A) PS=1.0P
ZAMAX=-10000
CALL MAX(2,4,ZA,ZAMAX,NX,NY)
IF(CP2.EQ.0.1A) PS=1.0P
 consulted for input, output, and attendance.
FILE: VOR FORTRAN P1

C CALCULATIONS IF PENETRATIONS IN EITHER TRIANGLE

104 IF((X(1)-1)-HT)/(X(1)-HT) LT=0.0 GC TO 134
C CONTINUE

105 IF((Z(1)-I)*HT)/(Z(1)-HT) LT=0.0 GC TO 120
C CONTINUE

106 IF((Z(1)-I)*HT)/(Z(1)-HT) LT=0.0
C CONTINUE

C LOGICAL PENETRATIONS ACCUMULATION

107 X(11)=X(1)
I=11
GO TO 110

108 X(11)=X(1)
I=11
GO TO 110

109 X(11)=X(1)
I=11
GO TO 110

110 X(11)=X(1)
I=11
GO TO 110

111 IF(X(11)-2) GO TO 114
I=2
I=1
GO TO 114

113 IF(X(11)-2) GO TO 114
I=2
I=1
GO TO 114

114 END
C SUBROUTINE [VPL](X,Y,Z,A,Y,G,X,A,Y,ZRES)
C LINEAR INTERPOLATION BY EQUATION 1 TO EVALUATE
C CP VALUE AT POINT X,Y,Z. THE RESULT RETURNED IS ZRES
C Dimension X(Y,J),Y(J,C),Z(J,C)
C SEARCH FOR 1 CP LEFT X LINE

DO 1 J=1,NX
IF(1.JC)-C+0.0 TO 2
CONTINUE

2 CONTINUE
C SEARCH FOR 1 CP LOWEST LINE

DO 3 JC=1,NY
IF(Y(JC)-GE+YG) GOTO 130
CONTINUE

3 CONTINUE
C RESULTING 1 CP IS IN LOWER TRIANGLE

X4=X(11)
Y4=Y(1)
Z4=Z(11)
GO TO 9

C RESULTING 1 CP IS IN UPPER TRIANGLE

X5=X(11)
Y5=Y(1)
Z5=Z(11)
GO TO 9

C GENERAL CALCULATION: LLCC FOR BOTH TRIANGLES

AL=X(11)
YL=Y(1)
ZL=Z(11)
XY=XY(1)
YY=YY(1)

115 BP=(Z2-Z1)*(Y1-Y2)*(Z1-Z2)*(Y1-Y3)/(X1-X3)*(Y1-Y2)-(X1-X2)

116 BEP=BP*AS+AP*AS+AP*AS

61
FILE: VAR  FORTRAN 01  

C RESULTING Z IS J ( J+1)J+1) J-J LINE
7 RES = ( 1 (1+1), 1+1) X(1) X(1+1) X(1+1)+Z(1J)
10 C RETURN
END
C RESULTING MAXARY ( Z , 2 MAX, NX, NY)
C EVALUATION OF MAXIMA CP OF CALIBRATION SURFACE MATRIX
5 D( 1, 1), IX
DO J = 1, NY
C EVALUATION OF MINIMA (Z , 2 MIN, NX, NY)
C EVALUATION OF MINIMA CP OF CALIBRATION SURFACE MATEIX
3 D( 1, 1), IX
DO J = 1, NY
1 IF (Z(J,J) > MAX) MAX = Z(J,J)
RETURN
END
C EVALUATION CP VER

END
FILE: SWDMF
FCRATAN
PI
READ(5,4,1) FT
WRITE(6,4,1) PT
IF(ACS.PC,1,0) UC TO 500
READ(5,4,1) P3
WRITE(6,4,1) PS
READ(5,4,1) ALFA
WRITE(6,4,1) ALF,
READ(5,4,1) PHI
WRITE(6,4,1) PHI
CALL INFLT(X,Y,2A,ALFA,PHII,NX,NY,CPS)
ALFC=ALFA-PS
CALL INFLT(X,Y,2A,ALFC,PHII,NX,NY,CPS)
ALFC=ALFA-PC
CALL INFLT(X,Y,2A,ALFC,PHII,NX,NY,CPS)
PHIB=PHII-YP
CALL INFLT(X,Y,F,28,ALFA,PHIB,NX,NY,CPS)

C READ MEASUREMENTS DATA
500 REAL(5,135) PDA
WRITE(6,136) PDA
READ(5,135) PPA
WRITE(6,136) PPA
READ(5,135) PPD
WRITE(6,136) PPD
READ(5,135) PPC
WRITE(6,136) PPC
501 REAL(5,135) PSI
WRITE(6,136) PSI
READ(5,135) PSI
WRITE(6,136) PSI
180 IF(ISCAN.EQ.1) 10 TO 501
111 IF(ISCVR.EQ.1) 90 TO 305
10 ISSCAN=ISCRN+50
WRITE(6,136) ISSCAN
11 C CALCULATES PRESSURE COEFFICIENTS
300 CALLMKE
POW=PI-PS
CPA=(PP-PS)/PO
CPF=PP/PO
CPC=(PP-PS)/DPS
PL=(PP-PS)/DPS
C CORRECTS PRESSURE DATA TO ENSURE DATA ARE IN CALIBRATION RANGE
IF(MDCPS.EQ.2) DO TO 301
220 IF(POW.EQ.1.0) DO TO 301
230 CALL MAXY(X,Y,AX,AY)
IF(ACSL.RT.1) DSC=1.0,1.0
IF(AFSL.RT.1) DSC=0.5,0.5
IF(PCS.LT.1) DSC=4.0,4.0
IF(PSC.LT.1) DSC=4.0,4.0
IF(PSC.LT.1) UC TO 301

65
**FILE: SWVCPR FORTRAN PL**

```
FILE: SWVCPR FORTRAN PL

ICCP = 1, ICPS = 1
IF (ICPS = GT. 4) GO TO 180
PS+P = 1
PS+P = 1
GO TO 342
C CALCULATES ALF & PHI ANGLES FOR '1' & '11'
C \( 11 \) is ALF & PHI ANGLES FOR '11'
C \( 1 \) is ALF & PHI ANGLES FOR '1'
ALF1 = ALF(1)
ALF1 = ALF(2)
ALF1 = ALF(3)
ALF1 = ALF(4)
C SELECTS THE PER ALF & PHI ANGLES OUT OF THE FOUR VALUES
ALF = ALF1(4)
ALF = ALF1(1)
ALF = ALF1(2)
ALF = ALF1(3)
C CALCULATES THE DATA OF POSITION D AND CORRECTS PT
ALF = ALF1(4)
C LINES:
```

---

**Explanation:**

The provided text seems to be a FORTRAN program snippet. It appears to calculate angles for different configurations and then selects the appropriate angles out of the four possible values for further calculations. The program involves various mathematical operations and angle calculations, typical of numerical analysis or simulation programs. The names and variables such as ALF and PHI suggest that these calculations could be related to fluid dynamics or similar fields where such angle calculations are common.
FILE: shwr fijtran p1

C CHECK FOR DIRECT HIT TO RIGHT INTERSECTIONS, EVALUATE IF REQUIRED
105 IF(150.D.0) GT 161
161 IF(FRA1H,0,REH) GO TO 162
RES(I)=-1
IF(ERL.EQ.0.) GC TO 163
GO TO 163
C CHECK FOR INTERMEDIATE RIGHT INTERSECTIONS, EVALUATE IF REQUIRED
162 IF(ERH.EQ.100) GC TO 162
RES(1)=-1
IF(ERL.EQ.100.) GC TO 163
C RE-INITIATE
163 RAA0=0.0
RAJ0=0.0
RAK0=0.0
140 C RTJ@NE
210 FORMAT(1,L00)"1 POINTS FOUND"
191 IF(RES(1),EQ.,CCORL) GO TO 191
IF(RES(1).EQ.,CCORL) GO TO 191
IF(RES(1).EQ.,CCORL) GO TO 191
GO TO 191
C EVALUATION OF INTERSECTIONS WHEN CURVES ARE ALMOST TANGENT
191 CALL FA55S(X1,Y1,X2,Y2,X3,Y3,L3,1-TY,Y1,ARRES)
CALL PE55S(X1,Y1,X2,Y2,X3,Y3,L3,1-TY,Y1,ARRES)
RES(1)=((Y3-Y1)*(X3-X2)-(2.0*Y4))-((Y3-Y2)*(X3-X4)-(2.0*Y3))
RES(1)=((Y3-Y2)*(X3-X2)-(2.0*Y1))-((Y3-Y1)*(X3-X1))
155 RETURN
END
SUBROUTINE PE55S(AX,AY,AX,AY,AX,AY,AX,AY)
PREPARE (4.0,4.0,4.0,4.0,4.0,4.0) AT 110
C SEARCH FOR J OF LOWER Y LINE
100 IF (Y(J).LT.Y(J)) TT TO 102
102 CONTINUE
C IJASE SEAF FOR ZERO PASS
C NEXT + LINE IF EXECUTED IN FIRST SEARCH ONLY
X(J)=Z(J)+1-Z(J)+1
IF(X(J).LT.0) GC TO 107
GO TO 112
112 GC L J=1,10
IF(X(J)+1-LT.0) GC TO L12
CONTINUE
GO TO 103
FILE: SWVJR FORTRAN P1

C CALCULATE SF PEVETATIONS IN EITHER TRIANGLE
104 A=X(I+1)-X(I)+X(I+1)-X(I-1)+Y(Y(I+1)-Y(I))
105 IF(I.EQ.2) GO TO 105
106 DO 105 K=1,N
107 X(K)=X(K)+A(K)+ABS(X(I+1)-X(I-1))
108 GO TO 110
109 C LOGICAL FEASIBILITY ACUMULATION
110 IF(K.EQ.1) GO TO 113
111 IF(I.EQ.2) GO TO 114
112 A=1
113 GO TO 112
114 RETURN
115 CONTINUE
116 ALNT=4
117 X(I)=X(I)
118 GO TO 115
119 C SEARCH FOR 1 CP LEFT X LINE
120 IF(X(I).GT.XG) GO TO 2
121 C CONTINUE
122 J=1
123 C SEARCH FOR 1 CP LEFT Y LINE
124 IF(Y(I).LT.YG) GO TO 4
125 C CONTINUE
126 X(I)=X(I)
127 Y(I)=Y(I)
128 IF(Y(I).EQ.YG) GO TO 5
129 C CONTINUE
130 X(I)=X(I)
131 Y(I)=Y(I)
132 C RESULTING Z IS IN LOWER TRIANGLE
133 X(I)=X(I)
134 Y(I)=Y(I)
135 RETURN
136 C RESULTING Z IS IN UPPER TRIANGLE
137 X(I)=X(I)
138 Y(I)=Y(I)
139 RETURN
140 C GENERAL CALCULATION: GCD FOR BOTH TRIANGLES
141 X(I)=X(I)
142 Y(I)=Y(I)
143 RETURN
144 C RESULTING Z IS ON DIVIDING LINE

69
FILE: SWOR  FORTRAN PL M P S

7    ZRES=Z(L+1,J+1)-Z(L,J)*2*(XU-XM)/(X(1+1)-X(1))*Z(1,J)
100   RETURN
END

SUBROUTINE ZMAXZ(1,2MAXZ,2NX,NY)
C EVALUATION OF "MAX" OF CALIBRATION SURFACE MATRIX
DIMENSION Z(40,40)
DO 1 J=1,40
    IF(1,J .LE. 2MAXZ) ZMAX=Z(1,J)
1     RETURN
END

SUBROUTINE ZMINZ(1,2MINZ,2NX,NY)
C EVALUATION OF "MIN" OF CALIBRATION SURFACE MATRIX
DIMENSION Z(40,40)
DO 1 J=1,40
    IF(1,J .LE. 2MINZ) ZMIN=Z(1,J)
1     RETURN
END

C END OF SWOR
APPENDIX 3

LISTING OF WVDR
FILE: 4VGJR  FORTRAN PL  N P S  

READ(L,121) PT
WRITE(6,L2) PT
IF(NEXT(I,10)) UC TO 500
READ(2,130) S
WRITE(6,L3) S
READ(3,140) ALFA
WRITE(6,L4) ALFA
READ(4,150) PHII
WRITE(6,L5) PHII
CALL L4PL1(AA1,AA2,ALFA,PHII,IX,XY,CPA)
ALFA=ALFA+AF1
CALL L4PL1(AA1,AA2,ALFC,PHII,IX,XY,CPA)
ALFC=ALFC+AF1
CALL L4PL1(AX,AY,AZ,ALFO,PHII,IX,XY,CPD)
PHII=PHII+IP
CALL L4PL1(AX,AY,AZ,ALFO,PHII,IX,XY,CPD)
WRITE(5,L6) CP,C FC,FD
PY=EP-PS
PP=CF>PPMPS
PP=CP>PPMPS
PP=LP>PPMPS
PP=UP>PPMPS
WRITE(6,L7) IPA,FP,EPP,PPC,PPD
402  F3M457S  CALL PT,PP,PS,PPC,PPD
403  C READ MEASUREMENTS: DATA
500  READ(L,121) PS
WRITE(6,L3) PS
READ(5,130) PPC
WRITE(6,L4) PPC
READ(6,130) PPD
WRITE(6,L5) PPD
READ(7,130) PSI
WRITE(5,L6) PSI
18C  IF(ISCAY=2) 30 TO 181
181  TT=1
ISCAY=ISCAY+1
PS=PS+PS
PPC=PPC+PS
CONTINUE
C CALCULATES PRESSURE COEFFICIENTS
900  C CONTINUE
WRITE(5,L6) IT
94C  FORML1(C,16-AFLC NUN, 1,15)
PP=PT>PS
CPA=(PP-1) / DT
CPD=(PP-2) / DT
CFC=(CPA-CPD) / CPD
CP=CP+(CPA-CPD) / DT
908  #IT=IT+1 #Pl+CP,CFC,CPA,CPD
C CORRECTS PRESSURE COEFFICIENT TO ENSURE CF ARE IN CALIBRATION RANGE
IF(CFC>1) 70    C C C "0"
ZFC*0.150)
CALL "YAY(A,AA2,ALFA,PHII,IX,XY,CPA)
IF(CFAC>Y) "STADY,STDY,STDY"
IF(CFAC<Y) "STADY,STDY,STDY"
IF(CFAC=Y) "STADY,STDY,STDY"
73
FILE: WVDP FURTRAN P

GO TO 163

C CHECK FOR INTERMEDIATE LEFT INTERSECTION, EVALUATE IF REQUIRED
160 IF(ISLEN=1.0) GO TO 160

GO TO 163

164 IF((IJL-EJL)/(IJL-RJL),GT,0.0) GC TO 163

AA=AP(1,JL-2)/IJL-RJL

AB=(ABJL-RJL)/IJL-RJL

RES1(N)=1.0-FV(1,1-A+B)

RES2(N)=AA-AP(1,JL-1)

IF(KC>2) GT 155

K=2

C CHECK FOR DIRECT HIT AT RIGHT INTERSECTION, EVALUATE IF REQUIRED
183 IF(ISN=1.0) GO TO 181

IS=1

GO TO 163

164 IF((IJL=ABJL-RJL),GT,0.0) GC TO 163

AA=AP(1,JL-2)/IJL-RJL

AB=(ABJL-RJL)/IJL-RJL

RES1(K)=AA-AP(1,JL-1)

RES2(K)=AB-AP(1,JL-2)

IF(K=2) GO 155

C RE-INITIATE
163 RRJL=FAIL

RAJL=FAIL

RES1=2.2

RES2=2.2

C EVALUATE IF INTERSECTIONS WHEN CURVES ARE ALMOST TANGENT
191 IF(ISR=1) GO TO 191

CALL PEPI3(Y,INTL,YJ,APRES)

RES1=APRES

RES2=APRES

RES1=0.75*(RES1+RES2)

RES2=RES1

C SEARCH FOR JLF, LOWER Y LINE
210 IF(YJ=1) GO TO 210

GO TO 191
FILE: WUR FORTRAN P1  N P S

101 CONTINUE

102 DO 1 SCAN FOR ZEROD PASS

103 C NEAT 6 LINES ARE EXECUTED IN FIRST SEARCH ONLY

112 GO TO 116

113 IF (X(I)-HT)/ (ZX(I)-HT) LT 0.7) GO TO 105

105 ABS((X(I)-HT) (YJ)-(1) ) GT 104
go TO 104

104 CALCULATE IF PERMUTATIONS IN EITHER TRIANGLE

112 IF (X(I)-HT)/ Y(I)-HT) LT 0.7) GO TO 103

103 ABS((X(I)-HT) (YJ)-(1) ) GT 103
go TO 103

102 CONTINUE

END SUBROUTINE LDPBT ((X,Y,Z,X1,Y1,X2,Y2) ZSBS)

C LINEAR INTERPOLATION ON CALIBRATION SURFACE TO EVALUATE

C CP VALLEY AT X JOXO111111 X5 Y5 THE RESULT RETURNED IS ZSBS

C SEARCH FOR I LEFT X LINE

DO 1 IF = 1,Y

1 CONTINUE

C SEARCH FOR J CF LOWEF Y LINE

DO 3 JF = 1,Y

3 CONTINUE

C RESULTING 2 IS 11 LOWER TRIANGLE

X3=X(I)

33=Y(J)

J=J+1

C RESULTING 2 IS 14 UPPER TRIANGLE

X3=X(I)

33=Y(J)

J=J+1

77
FILE: WVR

FORTRAN PL

N P S

C GENERAL CALCULATION: GCJU FOR BOTH TRIANGLES

6

XI=X(1)

YI=Y(1)

ZI=Z(1,1)

YZ=Y(Z+1)

Z2=Z(1+1,1)

Z3=Z(1,1+1)

AP=1(Z1-Z2)*((Y1*Y3)-(Z1-Z2)*(Y1*Y3))+(X1-X3)*(Y1-Y2)-(X1-X2)

10

PL=(Z1-Z2)*((X1-X2)/(Y1-Y2))

C=Z2-Z1

C=PL

ZRES=AP*G+MP*G

GT TO TJ

C RESULTING Z IS CALIBRATING LINE

7

ZRES=(Z1+1+1)+Z1,Z1+1+1)/X1+1+1-X1+1+1+Z1+1+1)

100 RETURN

END

SUBROUTINE MAXXY(ZMAX,AX,AY)

C EVALUATION OF MAXIMUM OF CP CALIBRATION SURFACE MARIAX

DIMENSION Z(40,40)

CZ 1=1,AX

CZ 1=1,A

IF(Z1,1)+GE.ZMAX) ZMAX=Z1,1)

RETURN

END

SUBROUTINE MINXY(ZMIN,AX,AY)

C EVALUATION OF MINIMUM OF CP CALIBRATION SURFACE MARIAX

DIMENSION Z(40,40)

DO 1 I=1,AX

GO TO 1

1 IF(Z1,1)+LE.ZMIN) ZMIN=Z1,1)

RETURN

END

C END OF WVR

78
<table>
<thead>
<tr>
<th>No.</th>
<th>DISTRIBUTION LIST</th>
<th>No. of Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Defense Documentation Center</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Cameron Station</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alexandria, Virginia 22314</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Library</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Code 0212</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93940</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Office of Research Administration</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Code 012A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93940</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Chairman</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Code 67</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Department of Aeronautics</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93940</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Director, Turbo-Propulsion Laboratory</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Department of Aeronautics</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Postgraduate School</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Monterey, California 93940</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Dr. H. J. Mueller</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Research Administrator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code 310A</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Air Systems Command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Navy Department</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Washington, D. C. 20360</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>Mr. Karl H. Guttmann</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Code 330C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Naval Air Systems Command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Navy Department</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Washington, D. C. 20360</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>Mr. James R. Patton, Jr.</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Power Program, Code 473</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Office of Naval Research</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arlington, Virginia 22218</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>Commanding Officer</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Naval Air Propulsion Test Center</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Attn: Mr. Vernon Lubosky</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trenton, New Jersey 08620</td>
<td></td>
</tr>
</tbody>
</table>
10. National Aeronautics and Space Administration
    Lewis Research Center (Library)
    2100 Brookpark Road
    Cleveland, Ohio 44135

11. CAG Library
    The Boeing Company
    Seattle, Washington 98124

12. Library
    General Electric Company
    Aircraft Engine Technology Division
    DTO Mail Drop H43
    Cincinnati, Ohio 45215

13. Library
    Pratt and Whitney Aircraft
    Post Office Box 2691
    West Palm Beach, Florida 33402

14. Library
    Pratt and Whitney Aircraft
    East Hartford, Connecticut 06108

15. Chief, Fan and Compressor Branch
    Mail Stop 5-9
    NASA Lewis Research Center
    21000 Brookpark Road
    Cleveland, Ohio 44135

16. Director, Whittle Laboratory
    Department of Engineering
    Cambridge University
    ENGLAND

17. Prof. D. Adler
    Technion Israel Institute of Technology
    Department of Mechanical Engineering
    Haifa 32000
    Israel

18. Prof. A. E. Breugelmans
    Institut von Karman de la Dynamique des Fluides
    72 Chausée de Waterloo
    1640 Rhode-St. Genese
    Belgium
<table>
<thead>
<tr>
<th></th>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>Mr. Robert O. Bullock</td>
<td>Air Research Mfg. Corporation Division of Garrett Corporation 402 South 36th Street Phoenix, Arizona 85034</td>
</tr>
<tr>
<td>20</td>
<td>Dr. F. O. Carta</td>
<td>United Technologies Research Labs 400 Main Street Hartford, Connecticut 06108</td>
</tr>
<tr>
<td>21</td>
<td>Prof. Jacques Chauvin</td>
<td>Universite D'Aix-Marseille 1 Rue Honnorat Marseille, France</td>
</tr>
<tr>
<td>22</td>
<td>Mr. James V. Davis</td>
<td>Teledyne CAE 1330 Laskey Road Toledo, Ohio 43601</td>
</tr>
<tr>
<td>23</td>
<td>Mr. Jean Fabri</td>
<td>ONERA 25, Ave. de la Division Leclerc 92 Chatillon France</td>
</tr>
<tr>
<td>24</td>
<td>Prof. Dr. Ing Heinz E. Gallus</td>
<td>Lehrstuhl und Institut fur Strahlantiebe und Turbourbeitsmaschinen Rhein-Westf. Techn. Hochschule Aachen Templergraben 55 5100 Aachen, Germany</td>
</tr>
<tr>
<td>25</td>
<td>Professor J. P. Gostelow</td>
<td>School of Mechanical Engineering The New South Wales Institute of Technology Australia</td>
</tr>
<tr>
<td>26</td>
<td>DR. Ing. Hans-J. Heineman</td>
<td>DFVLR-AVA Bunsenstasse 10 3400 Gottingen, W. Germany</td>
</tr>
<tr>
<td>27</td>
<td>Prof. Ch. Hirsch</td>
<td>Vrije Universiteit Brussel Pleinlaan 2 1050 Brusselis, Belgium</td>
</tr>
</tbody>
</table>
28. Prof. J. P. Johnston  
Stanford University  
Department of Mechanical Engineering  
Stanford, California 94305

29. Prof. Jack L. Kerrebrock, Chairman  
Aeronautics and Astronautics Department  
31-265 Massachusetts Institute of Technology  
Cambridge, Massachusetts 02139

30. Dr. B. Lukshinarayana  
Professor of Aerospace Engineering  
The Pennsylvania State University  
233 Hammond Building  
University Park, Pennsylvania 16802

31. Mr. R. A. Langworthy  
Army Aviation Materials Laboratories  
Department of the Army  
Fort Eustis, Virginia 23604

32. Dr. A. A. Mikolajczak  
Pratt and Whitney Aircraft Engineering 2H  
East Hartford, Connecticut 06108

33. Prof. Dr. L. G. Napolitano  
Director  
Institute of Aerodynamics  
University of Naples  
Viale C. Augusto 60125 Napoli  
Italy

34. Prof. Erik Nilsson  
Institutionen for Stromningsmaskinteknik  
Chalmers Tekniska Hogskola  
Fack, 402 20 Goteborg 5  
Sweden

35. Prof. Gordon C. Oates  
Department of Aeronautics and Astronautics  
University of Washington  
Seattle, Washington 98105

36. Prof. Walter F. O'Brien  
Mechanical Engineering Department  
Virginia Polytechnic Institute and State University  
Blacksburg, VA 24061
37. Prof. Dr. K. Oswatitsch  
Technische Hochschule  
Karlsplatz 13  
Vienna, Austria

38. Dr. P. A. Paranjee  
Head, Propulsion Division  
National Aeronautical Laboratory  
Post Bag 1799  
Bangalore - 17  
India

39. R. E. Peacock  
School of Mechanical Engineering  
Cranfield Institute of Technology  
Cranfield, Bedford MK43 QAL  
ENGLAND

40. Dr. Bruce A. Reese  
Director, Jet Propulsion Center  
School of Mechanical Engineering  
Purdue University  
Lafayette, Indiana 47907

41. Dr. W. Schlachter  
Brown, Boveri-Sulzer Turbomachinery Ltd  
Dept. TDE  
Escher Wyss Platz  
CH-8023 Zurich  
Switzerland

42. Dr. George K. Serovy  
Professor of Mechanical Engineering  
208 Mechanical Engineering Building  
Iowa State University  
Ames, Iowa 50010

43. Dr. Fernando Sisto  
Professor and Head of Mechanical Engineering Department  
Stevens Institute of Technology  
Castle Point, Hoboken, New Jersey 07030

44. Dr. Leroy H. Smith, Jr.  
Manager, Compressor and Fan Technology Operation  
General Electric Company  
Aircraft Engine Technology Division  
DTO Mail Drop HA3  
Cincinnati, Ohio 45215
45. Dr. W. Tabakoff  
Professor, Department of Aerospace Engineering  
University of Cincinnati  
Cincinnati, Ohio 45221

46. Mr. P. Tramm  
Manager, Research Labs  
Detroit Diesel Allison Division  
General Motors  
P. O. Box 894  
Indianapolis, Indiana 46206

47. Prof. Dr. W. Trauelp  
institut fur Thermische Turbomaschinen  
Eidg. Technische Hochschule

48. Dr. Arthur J. Wennerstrom  
ARL/LF  
Wright-Patterson AFB  
Dayton, Ohio 45433

49. Dr. H. Weyer  
DFVLR  
Linder Hohe  
505 Porz-Wahn  
Germany

50. Mr. P. F. Yaggy  
Director  
U. S. Army Aeronautical Research Laboratory  
AMES Research Center  
Moffett Field, California 94035