

LEVEL # 2

NPS67-79-007

NAVAL POSTGRADUATE SCHOOL

Monterey, California

ADA 080798



DDC
RECEIVED
FEB 19 1980
A

THIS DOCUMENT IS BEST QUALITY PRACTICABLE
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

DDC FILE COPY

A General Procedure for Obtaining Velocity
Vector from a System of High Response Impact
Pressure Probes

D. Adler and R. P. Shreeve

July 1979

THIS DOCUMENT IS BEST QUALITY PRACTICABLE
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

Approved for public release; distribution
unlimited

Prepared for:
Naval Air Systems Command
Washington, DC

Office of Naval Research
Arlington, VA

and

80 19 119

NAVAL POSTGRADUATE SCHOOL

Monterey, California

Rear Admiral T. F. Dedman
Superintendent

Jack R. Borsting
Provost

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
D. Adler, Visiting Professor
of Aeronautics

R. P. Shreeve
R. P. Shreeve, Associate
Professor of Aeronautics

Reviewed by:

Released by:

M. F. Platzter
M. F. PLATZER, Chairman
Department of Aeronautics

William M. Tolles
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.**

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER NPS67-79-007	2. GOVT ACCESSION NO.	3. REPORT'S CATALOG NUMBER 9
4. TITLE (and Subtitle) A General Procedure for Obtaining Velocity Vector from a System of High Response Impact Pressure Probes		5. DATE OF REPORT & PERIOD COVERED Technical Report, July 1978 - July 1979
7. AUTHOR(s) D. Adler and R. P. Shreeve	8. CONTRACT OR GRANT NUMBER(s) 61153N; N00019-79-WR-91115	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Air Systems Command Washington, DC 20361		12. REPORT DATE July 1979
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 1289		13. NUMBER OF PAGES 28 84 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)

Flow Measurements

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

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.

251450

A General Procedure for Obtaining Velocity
Vector From a System of High
Response Impact Pressure Probes

by

D. Adler and R. P. Shreeve

Reference For	
ADP 11431	<input checked="" type="checkbox"/>
ADP 11432	<input type="checkbox"/>
ADP 11433	<input type="checkbox"/>
ADP 11434	<input type="checkbox"/>
ADP 11435	<input type="checkbox"/>
ADP 11436	<input type="checkbox"/>
ADP 11437	<input type="checkbox"/>
ADP 11438	<input type="checkbox"/>
ADP 11439	<input type="checkbox"/>
ADP 11440	<input type="checkbox"/>
ADP 11441	<input type="checkbox"/>
ADP 11442	<input type="checkbox"/>
ADP 11443	<input type="checkbox"/>
ADP 11444	<input type="checkbox"/>
ADP 11445	<input type="checkbox"/>
ADP 11446	<input type="checkbox"/>
ADP 11447	<input type="checkbox"/>
ADP 11448	<input type="checkbox"/>
ADP 11449	<input type="checkbox"/>
ADP 11450	<input type="checkbox"/>
ADP 11451	<input type="checkbox"/>
ADP 11452	<input type="checkbox"/>
ADP 11453	<input type="checkbox"/>
ADP 11454	<input type="checkbox"/>
ADP 11455	<input type="checkbox"/>
ADP 11456	<input type="checkbox"/>
ADP 11457	<input type="checkbox"/>
ADP 11458	<input type="checkbox"/>
ADP 11459	<input type="checkbox"/>
ADP 11460	<input type="checkbox"/>
ADP 11461	<input type="checkbox"/>
ADP 11462	<input type="checkbox"/>
ADP 11463	<input type="checkbox"/>
ADP 11464	<input type="checkbox"/>
ADP 11465	<input type="checkbox"/>
ADP 11466	<input type="checkbox"/>
ADP 11467	<input type="checkbox"/>
ADP 11468	<input type="checkbox"/>
ADP 11469	<input type="checkbox"/>
ADP 11470	<input type="checkbox"/>
ADP 11471	<input type="checkbox"/>
ADP 11472	<input type="checkbox"/>
ADP 11473	<input type="checkbox"/>
ADP 11474	<input type="checkbox"/>
ADP 11475	<input type="checkbox"/>
ADP 11476	<input type="checkbox"/>
ADP 11477	<input type="checkbox"/>
ADP 11478	<input type="checkbox"/>
ADP 11479	<input type="checkbox"/>
ADP 11480	<input type="checkbox"/>
ADP 11481	<input type="checkbox"/>
ADP 11482	<input type="checkbox"/>
ADP 11483	<input type="checkbox"/>
ADP 11484	<input type="checkbox"/>
ADP 11485	<input type="checkbox"/>
ADP 11486	<input type="checkbox"/>
ADP 11487	<input type="checkbox"/>
ADP 11488	<input type="checkbox"/>
ADP 11489	<input type="checkbox"/>
ADP 11490	<input type="checkbox"/>
ADP 11491	<input type="checkbox"/>
ADP 11492	<input type="checkbox"/>
ADP 11493	<input type="checkbox"/>
ADP 11494	<input type="checkbox"/>
ADP 11495	<input type="checkbox"/>
ADP 11496	<input type="checkbox"/>
ADP 11497	<input type="checkbox"/>
ADP 11498	<input type="checkbox"/>
ADP 11499	<input type="checkbox"/>
ADP 11500	<input type="checkbox"/>
ADP 11501	<input type="checkbox"/>
ADP 11502	<input type="checkbox"/>
ADP 11503	<input type="checkbox"/>
ADP 11504	<input type="checkbox"/>
ADP 11505	<input type="checkbox"/>
ADP 11506	<input type="checkbox"/>
ADP 11507	<input type="checkbox"/>
ADP 11508	<input type="checkbox"/>
ADP 11509	<input type="checkbox"/>
ADP 11510	<input type="checkbox"/>
ADP 11511	<input type="checkbox"/>
ADP 11512	<input type="checkbox"/>
ADP 11513	<input type="checkbox"/>
ADP 11514	<input type="checkbox"/>
ADP 11515	<input type="checkbox"/>
ADP 11516	<input type="checkbox"/>
ADP 11517	<input type="checkbox"/>
ADP 11518	<input type="checkbox"/>
ADP 11519	<input type="checkbox"/>
ADP 11520	<input type="checkbox"/>
ADP 11521	<input type="checkbox"/>
ADP 11522	<input type="checkbox"/>
ADP 11523	<input type="checkbox"/>
ADP 11524	<input type="checkbox"/>
ADP 11525	<input type="checkbox"/>
ADP 11526	<input type="checkbox"/>
ADP 11527	<input type="checkbox"/>
ADP 11528	<input type="checkbox"/>
ADP 11529	<input type="checkbox"/>
ADP 11530	<input type="checkbox"/>
ADP 11531	<input type="checkbox"/>
ADP 11532	<input type="checkbox"/>
ADP 11533	<input type="checkbox"/>
ADP 11534	<input type="checkbox"/>
ADP 11535	<input type="checkbox"/>
ADP 11536	<input type="checkbox"/>
ADP 11537	<input type="checkbox"/>
ADP 11538	<input type="checkbox"/>
ADP 11539	<input type="checkbox"/>
ADP 11540	<input type="checkbox"/>
ADP 11541	<input type="checkbox"/>
ADP 11542	<input type="checkbox"/>
ADP 11543	<input type="checkbox"/>
ADP 11544	<input type="checkbox"/>
ADP 11545	<input type="checkbox"/>
ADP 11546	<input type="checkbox"/>
ADP 11547	<input type="checkbox"/>
ADP 11548	<input type="checkbox"/>
ADP 11549	<input type="checkbox"/>
ADP 11550	<input type="checkbox"/>
ADP 11551	<input type="checkbox"/>
ADP 11552	<input type="checkbox"/>
ADP 11553	<input type="checkbox"/>
ADP 11554	<input type="checkbox"/>
ADP 11555	<input type="checkbox"/>
ADP 11556	<input type="checkbox"/>
ADP 11557	<input type="checkbox"/>
ADP 11558	<input type="checkbox"/>
ADP 11559	<input type="checkbox"/>
ADP 11560	<input type="checkbox"/>
ADP 11561	<input type="checkbox"/>
ADP 11562	<input type="checkbox"/>
ADP 11563	<input type="checkbox"/>
ADP 11564	<input type="checkbox"/>
ADP 11565	<input type="checkbox"/>
ADP 11566	<input type="checkbox"/>
ADP 11567	<input type="checkbox"/>
ADP 11568	<input type="checkbox"/>
ADP 11569	<input type="checkbox"/>
ADP 11570	<input type="checkbox"/>
ADP 11571	<input type="checkbox"/>
ADP 11572	<input type="checkbox"/>
ADP 11573	<input type="checkbox"/>
ADP 11574	<input type="checkbox"/>
ADP 11575	<input type="checkbox"/>
ADP 11576	<input type="checkbox"/>
ADP 11577	<input type="checkbox"/>
ADP 11578	<input type="checkbox"/>
ADP 11579	<input type="checkbox"/>
ADP 11580	<input type="checkbox"/>
ADP 11581	<input type="checkbox"/>
ADP 11582	<input type="checkbox"/>
ADP 11583	<input type="checkbox"/>
ADP 11584	<input type="checkbox"/>
ADP 11585	<input type="checkbox"/>
ADP 11586	<input type="checkbox"/>
ADP 11587	<input type="checkbox"/>
ADP 11588	<input type="checkbox"/>
ADP 11589	<input type="checkbox"/>
ADP 11590	<input type="checkbox"/>
ADP 11591	<input type="checkbox"/>
ADP 11592	<input type="checkbox"/>
ADP 11593	<input type="checkbox"/>
ADP 11594	<input type="checkbox"/>
ADP 11595	<input type="checkbox"/>
ADP 11596	<input type="checkbox"/>
ADP 11597	<input type="checkbox"/>
ADP 11598	<input type="checkbox"/>
ADP 11599	<input type="checkbox"/>
ADP 11600	<input type="checkbox"/>
ADP 11601	<input type="checkbox"/>
ADP 11602	<input type="checkbox"/>
ADP 11603	<input type="checkbox"/>
ADP 11604	<input type="checkbox"/>
ADP 11605	<input type="checkbox"/>
ADP 11606	<input type="checkbox"/>
ADP 11607	<input type="checkbox"/>
ADP 11608	<input type="checkbox"/>
ADP 11609	<input type="checkbox"/>
ADP 11610	<input type="checkbox"/>
ADP 11611	<input type="checkbox"/>
ADP 11612	<input type="checkbox"/>
ADP 11613	<input type="checkbox"/>
ADP 11614	<input type="checkbox"/>
ADP 11615	<input type="checkbox"/>
ADP 11616	<input type="checkbox"/>
ADP 11617	<input type="checkbox"/>
ADP 11618	<input type="checkbox"/>
ADP 11619	<input type="checkbox"/>
ADP 11620	<input type="checkbox"/>
ADP 11621	<input type="checkbox"/>
ADP 11622	<input type="checkbox"/>
ADP 11623	<input type="checkbox"/>
ADP 11624	<input type="checkbox"/>
ADP 11625	<input type="checkbox"/>
ADP 11626	<input type="checkbox"/>
ADP 11627	<input type="checkbox"/>
ADP 11628	<input type="checkbox"/>
ADP 11629	<input type="checkbox"/>
ADP 11630	<input type="checkbox"/>
ADP 11631	<input type="checkbox"/>
ADP 11632	<input type="checkbox"/>
ADP 11633	<input type="checkbox"/>
ADP 11634	<input type="checkbox"/>
ADP 11635	<input type="checkbox"/>
ADP 11636	<input type="checkbox"/>
ADP 11637	<input type="checkbox"/>
ADP 11638	<input type="checkbox"/>
ADP 11639	<input type="checkbox"/>
ADP 11640	<input type="checkbox"/>
ADP 11641	<input type="checkbox"/>
ADP 11642	<input type="checkbox"/>
ADP 11643	<input type="checkbox"/>
ADP 11644	<input type="checkbox"/>
ADP 11645	<input type="checkbox"/>
ADP 11646	<input type="checkbox"/>
ADP 11647	<input type="checkbox"/>
ADP 11648	<input type="checkbox"/>
ADP 11649	<input type="checkbox"/>
ADP 11650	<input type="checkbox"/>
ADP 11651	<input type="checkbox"/>
ADP 11652	<input type="checkbox"/>
ADP 11653	<input type="checkbox"/>
ADP 11654	<input type="checkbox"/>
ADP 11655	<input type="checkbox"/>
ADP 11656	<input type="checkbox"/>
ADP 11657	<input type="checkbox"/>
ADP 11658	<input type="checkbox"/>
ADP 11659	<input type="checkbox"/>
ADP 11660	<input type="checkbox"/>
ADP 11661	<input type="checkbox"/>
ADP 11662	<input type="checkbox"/>
ADP 11663	<input type="checkbox"/>
ADP 11664	<input type="checkbox"/>
ADP 11665	<input type="checkbox"/>
ADP 11666	<input type="checkbox"/>
ADP 11667	<input type="checkbox"/>
ADP 11668	<input type="checkbox"/>
ADP 11669	<input type="checkbox"/>
ADP 11670	<input type="checkbox"/>
ADP 11671	<input type="checkbox"/>
ADP 11672	<input type="checkbox"/>
ADP 11673	<input type="checkbox"/>
ADP 11674	<input type="checkbox"/>
ADP 11675	<input type="checkbox"/>
ADP 11676	<input type="checkbox"/>
ADP 11677	<input type="checkbox"/>
ADP 11678	<input type="checkbox"/>
ADP 11679	<input type="checkbox"/>
ADP 11680	<input type="checkbox"/>
ADP 11681	<input type="checkbox"/>
ADP 11682	<input type="checkbox"/>
ADP 11683	<input type="checkbox"/>
ADP 11684	<input type="checkbox"/>
ADP 11685	<input type="checkbox"/>
ADP 11686	<input type="checkbox"/>
ADP 11687	<input type="checkbox"/>
ADP 11688	<input type="checkbox"/>
ADP 11689	<input type="checkbox"/>
ADP 11690	<input type="checkbox"/>
ADP 11691	<input type="checkbox"/>
ADP 11692	<input type="checkbox"/>
ADP 11693	<input type="checkbox"/>
ADP 11694	<input type="checkbox"/>
ADP 11695	<input type="checkbox"/>
ADP 11696	<input type="checkbox"/>
ADP 11697	<input type="checkbox"/>
ADP 11698	<input type="checkbox"/>
ADP 11699	<input type="checkbox"/>
ADP 11700	<input type="checkbox"/>
ADP 11701	<input type="checkbox"/>
ADP 11702	<input type="checkbox"/>
ADP 11703	<input type="checkbox"/>
ADP 11704	<input type="checkbox"/>
ADP 11705	<input type="checkbox"/>
ADP 11706	<input type="checkbox"/>
ADP 11707	<input type="checkbox"/>
ADP 11708	<input type="checkbox"/>
ADP 11709	<input type="checkbox"/>
ADP 11710	<input type="checkbox"/>
ADP 11711	<input type="checkbox"/>
ADP 11712	<input type="checkbox"/>
ADP 11713	<input type="checkbox"/>
ADP 11714	<input type="checkbox"/>
ADP 11715	<input type="checkbox"/>
ADP 11716	<input type="checkbox"/>
ADP 11717	<input type="checkbox"/>
ADP 11718	<input type="checkbox"/>
ADP 11719	<input type="checkbox"/>
ADP 11720	<input type="checkbox"/>
ADP 11721	<input type="checkbox"/>
ADP 11722	<input type="checkbox"/>
ADP 11723	<input type="checkbox"/>
ADP 11724	<input type="checkbox"/>
ADP 11725	<input type="checkbox"/>
ADP 11726	<input type="checkbox"/>
ADP 11727	<input type="checkbox"/>
ADP 11728	<input type="checkbox"/>
ADP 11729	<input type="checkbox"/>
ADP 11730	<input type="checkbox"/>
ADP 11731	<input type="checkbox"/>
ADP 11732	<input type="checkbox"/>
ADP 11733	<input type="checkbox"/>
ADP 11734	<input type="checkbox"/>
ADP 11735	<input type="checkbox"/>
ADP 11736	<input type="checkbox"/>
ADP 11737	<input type="checkbox"/>
ADP 11738	<input type="checkbox"/>
ADP 11739	<input type="checkbox"/>
ADP 11740	<input type="checkbox"/>
ADP 11741	<input type="checkbox"/>
ADP 11742	<input type="checkbox"/>
ADP 11743	<input type="checkbox"/>
ADP 11744	<input type="checkbox"/>
ADP 11745	<input type="checkbox"/>
ADP 11746	<input type="checkbox"/>
ADP 11747	<input type="checkbox"/>
ADP 11748	<input type="checkbox"/>
ADP 11749	<input type="checkbox"/>
ADP 11750	<input type="checkbox"/>
ADP 11751	<input type="checkbox"/>
ADP 11752	<input type="checkbox"/>
ADP 11753	<input type="checkbox"/>
ADP 11754	<input type="checkbox"/>
ADP 11755	<input type="checkbox"/>
ADP 11756	<input type="checkbox"/>
ADP 11757	<input type="checkbox"/>
ADP 11758	<input type="checkbox"/>
ADP 11759	<input type="checkbox"/>
ADP 11760	<input type="checkbox"/>
ADP 11761	<input type="checkbox"/>
ADP 11762	<input type="checkbox"/>
ADP 11763	<input type="checkbox"/>
ADP 11764	<input type="checkbox"/>
ADP 11765	<input type="checkbox"/>
ADP 11766	<input type="checkbox"/>
ADP 11767	<input type="checkbox"/>
ADP 11768	<input type="checkbox"/>
ADP 11769	<input type="checkbox"/>
ADP 11770	<input type="checkbox"/>
ADP 11771	<input type="checkbox"/>
ADP 11772	<input type="checkbox"/>
ADP 11773	<input type="checkbox"/>
ADP 11774	<input type="checkbox"/>
ADP 11775	<input type="checkbox"/>
ADP 11776	<input type="checkbox"/>
ADP 11777	<input type="checkbox"/>
ADP 11778	<input type="checkbox"/>
ADP 11779	<input type="checkbox"/>
ADP 11780	<input type="checkbox"/>
ADP 11781	<input type="checkbox"/>
ADP 11782	<input type="checkbox"/>
ADP 11783	<input type="checkbox"/>
ADP 11784	<input type="checkbox"/>
ADP 11785	<input type="checkbox"/>
ADP 11786	<input type="checkbox"/>
ADP 11787	<input type="checkbox"/>
ADP 11788	<input type="checkbox"/>
ADP 11789	<input type="checkbox"/>
ADP 11790	<input type="checkbox"/>
ADP 11791	<input type="checkbox"/>
ADP 11792	<input type="checkbox"/>
ADP 11793	<input type="checkbox"/>
ADP 11794	<input type="checkbox"/>
ADP 11795	<input type="checkbox"/>
ADP 11796	<input type="checkbox"/>
ADP 11797	<input type="checkbox"/>
ADP 11798	<input type="checkbox"/>
ADP 11799	<input type="checkbox"/>
ADP 11800	<input type="checkbox"/>

Contents

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

Notation

- C_p - Pressure coefficient defined in eq. (3)
 α - Yaw angle relative to laboratory space
 ϕ - Pitch angle relative to laboratory space
 P_t - Total pressure
 P_s - Static Pressure
 α_{rp} - Yaw angle relative to probe axis
 ϕ_{rp} - Pitch angle relative to probe axis
 P_p - Pressure indicated by the probe
 α_p - Yaw setting of the probe
 ϕ_p - Pitch setting of the probe

Subscripts

- I }
II } relating to probe positions shown
III } 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 $HT = C_p$ 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 $C_{p_{rp}}$ for X1(I) and Y1(I)
- Z2(I,J) - values of $C_{p_{rp}}$ for X2(I) and Y2(I)
- Z3(I,J) - values of $C_{p_{rp}}$ for X3(I) and Y3(I)
- Z4(I,J) - values of $C_{p_{rp}}$ 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 α returned to VDR from INTSCS
- ALFA - α input when experiment simulation is carried out
- ALFC - α_{rpIII}
- ALFCN - new value of α_{rpIII} for next iteration
- ALFD - α_{rpIV}
- ALFDN - new value of α_{rpIV} for next iteration
- ALF1 - α_I
- ALF2 - α_{II}
- ALF3 - α_I
- ALF4 - α_{III}
- AM - mach number
- CPA - $C_{p_{rpI}}$
- CPB - $C_{p_{rpII}}$
- CPC - $C_{p_{rpIII}}$
- CPD - $C_{p_{rpIV}}$
- EPSPS - relative difference between present static pressure and its value in the previous iteration
- EPSPSG - convergence criterion (on the static pressure)
- FALF - final α value (at convergence)
- FPHI - final ϕ 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 C_p '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, α values (must be identical for all calibration surface matrices used)
- NY - number of calibration surface matrix ordinates, ϕ values (must be identical for all calibration surface matrices used)
- PDYN - dynamic pressure
- PHI - value of ϕ returned to VDR from INTSCS
- PHIB - ϕ_{rpII}
- PHII - ϕ input when experiment simulation is carried out
- PHI1 - ϕ_{rpI}
- PHI2 - ϕ_{rpII}
- PHI3 - ϕ_{rpI}
- PHI4 - ϕ_{rpIII}
- 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 - α value at which scanning starts
 XRLU - lowest α value of calibration surface (as a condition it must be identical for probes A and B)
 XRUP - highest α 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

YPA - pitch setting of probe A
YPB - 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 - Cp_{rpA} values of probe A
ZAX - transformed Cp_{rpA} values of probe A
ZAMAX - maximum of ZA array
ZAMIN - minimum of ZA array
ZB - Cp_{rpB} values of probe B
ZBX - transformed Cp_{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 pressure 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

TYPE A PROBE

TYPE B PROBE

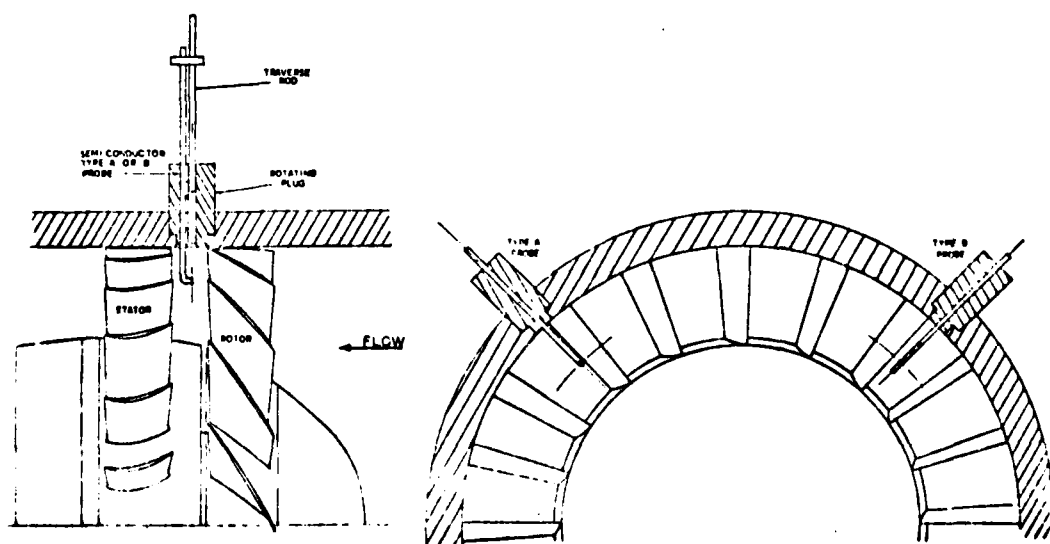
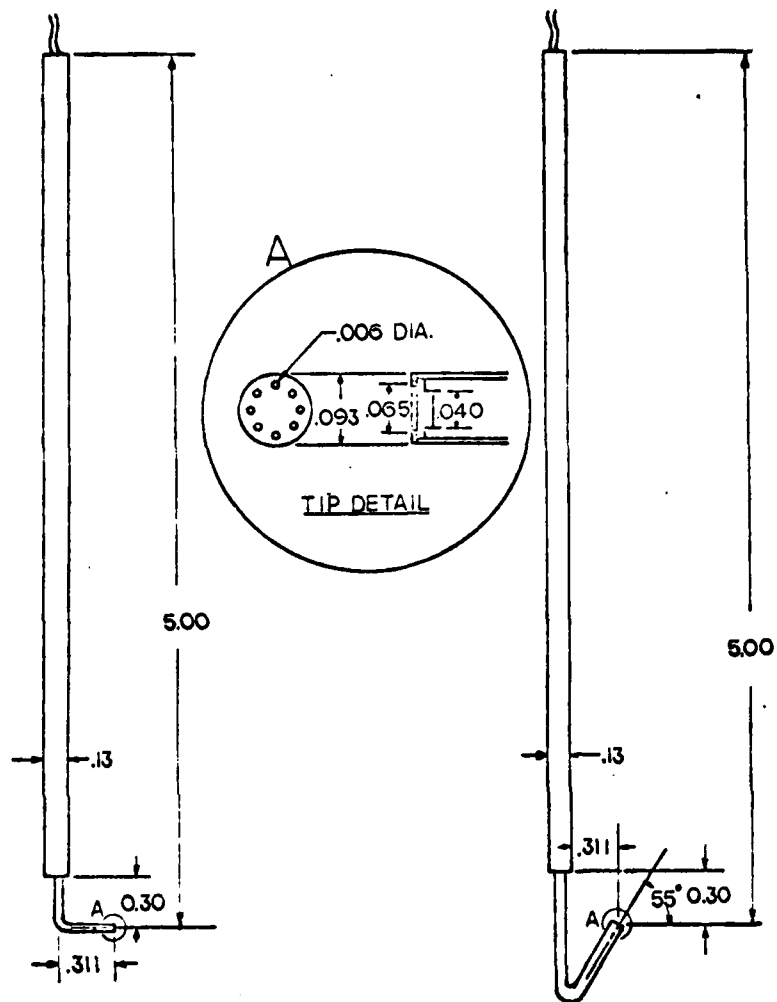


FIGURE 1. TYPE A AND TYPE B PROBES AND COMPRESSOR INSTALLATION.

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:

$$P_p = P_p(\alpha, \phi, P_T, P_S) \quad (1)$$

If a pressure coefficient is defined as

$$C_{p_{rp}} = \frac{P_p - P_S}{P_T - P_S} \quad (2)$$

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

$$C_{p_{rp}} = C_{p_{rp}}(\alpha_{rp}, \phi_{rp}) \quad (3)$$

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.

$0 < Ma < 0.7$ 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.

$$\begin{aligned} C_{p_{Ai}} &= \frac{P_{p_{Ai}} - P_S}{P_T - P_S} = C_{p_{Ai}}(\alpha, \phi) \\ C_{p_{BII}} &= \frac{P_{p_{BII}} - P_S}{P_T - P_S} = C_{p_{BII}}(\alpha, \phi) \end{aligned} \quad (4)$$

it should be pointed out here, to avoid misunderstanding that α and ϕ 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: α , ϕ , P_T and P_S . 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 α and ϕ i.e., for a given C_p and α there exist two ϕ values, or for a given C_p and ϕ there are two α values which satisfy eq. 3.

- 3) As a result of the measurement the value of P_p rather than of 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 α and ϕ . 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

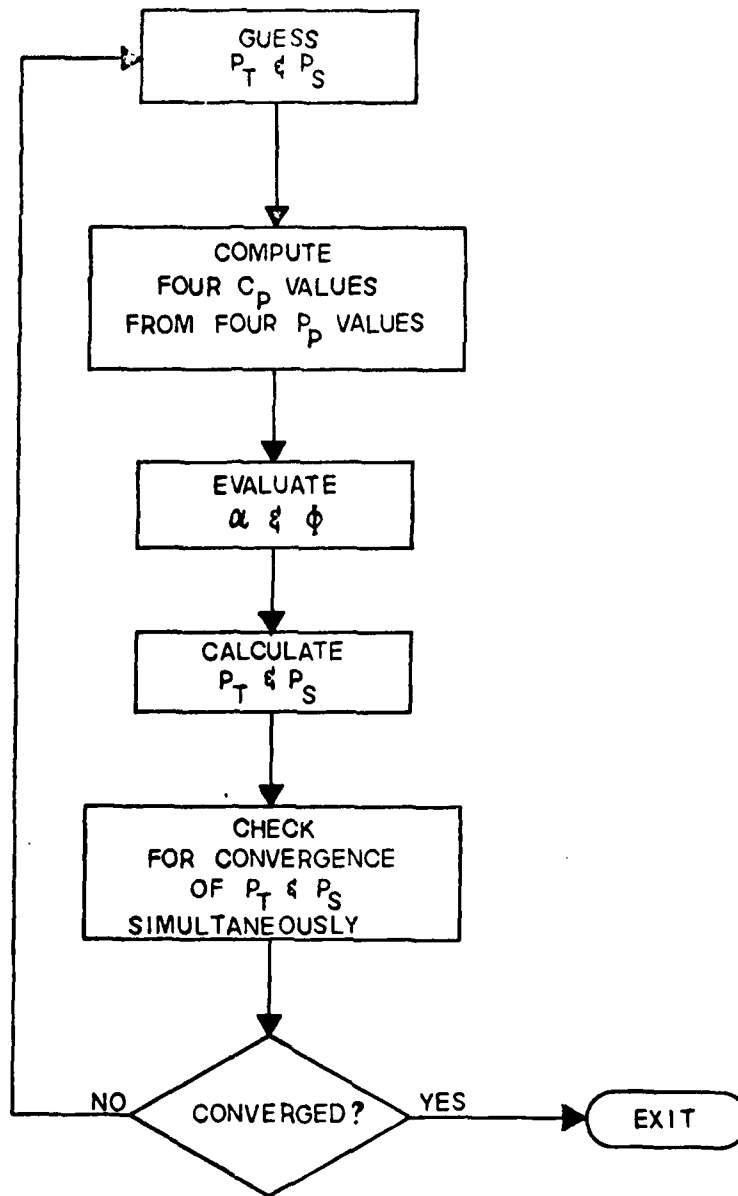


Fig. 2: Unstable iterative solution using measurement of A type and B type probes only.

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.

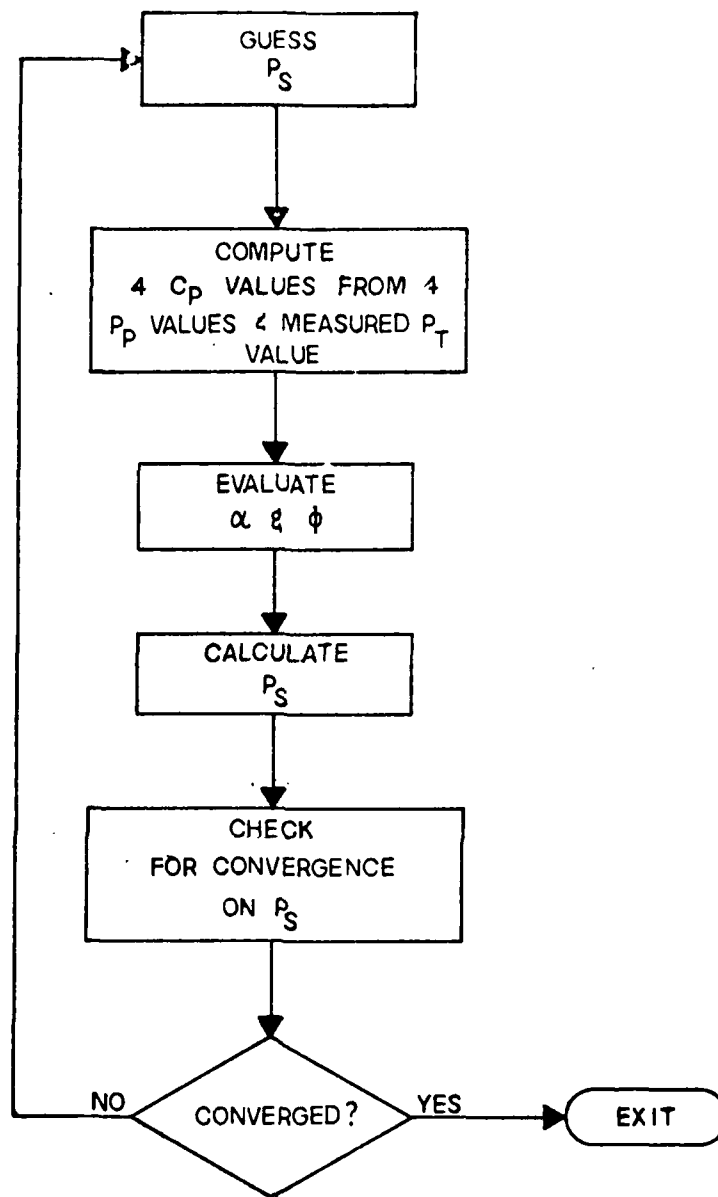


Fig. 3: Stable iterative solution using measurements of A type and B type probes as well as measurements of a Kiel probe.

Evaluation of α and ϕ (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 $Cp_{rp} = Cp_{rp}(\alpha_{rp}, \phi_{rp})$, where α_{rp} and ϕ_{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 $Cp = Cp(\alpha, \phi)$ can be written as

$$Cp = Cp[(\alpha_p + \alpha_{rp}), (\phi_p + \phi_{rp})] \quad (5)$$

where α_p and ϕ_p are the probe tip axis angular settings relative to the laboratory space. It is clear from eq. (5) that Cp can be derived from Cp_{rp} by a constant translation: α_p, ϕ_p on the α, ϕ plane. As each Cp_{rp} can be viewed as a hill with its peak at $\alpha_{rp} = 0$ and $\phi_{rp} = 0$. The Cp 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:

$$I) \quad \alpha_p = 0 \quad \phi_p = 0$$

$$\text{III) } \alpha_p = \alpha_{p\text{III}} \quad \phi_p = 0$$

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

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

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

For this case the topography of the calibration surfaces will appear as four hills. The three with their peaks at points $(0,0)$; $(+\alpha_{p\text{III}}, 0)$ and $(-\alpha_{p\text{IV}}, 0)$ are the translated hills $C_{p\text{rpA}}$ and the fourth with its peak at $(0, \phi_{p\text{II}})$ is the translated hill $C_{p\text{rpB}}$. Their contours of constant C_p are then as shown in figure 4.

Assume now that a velocity vector with yaw α and pitch ϕ is to be measured. These values of α and ϕ will be sensed uniquely by the probes at their four angular settings. Were equation (4) single valued, the values of α and ϕ could be uniquely evaluated, as the single intersection point between the projections of appropriate lines of constant C_p on the α, ϕ plane*. In the present case of the double valued functions, the lines of constant C_p are closed curves and more than a single intersection point do exist.

* C_p 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) C_p values on the different hills.

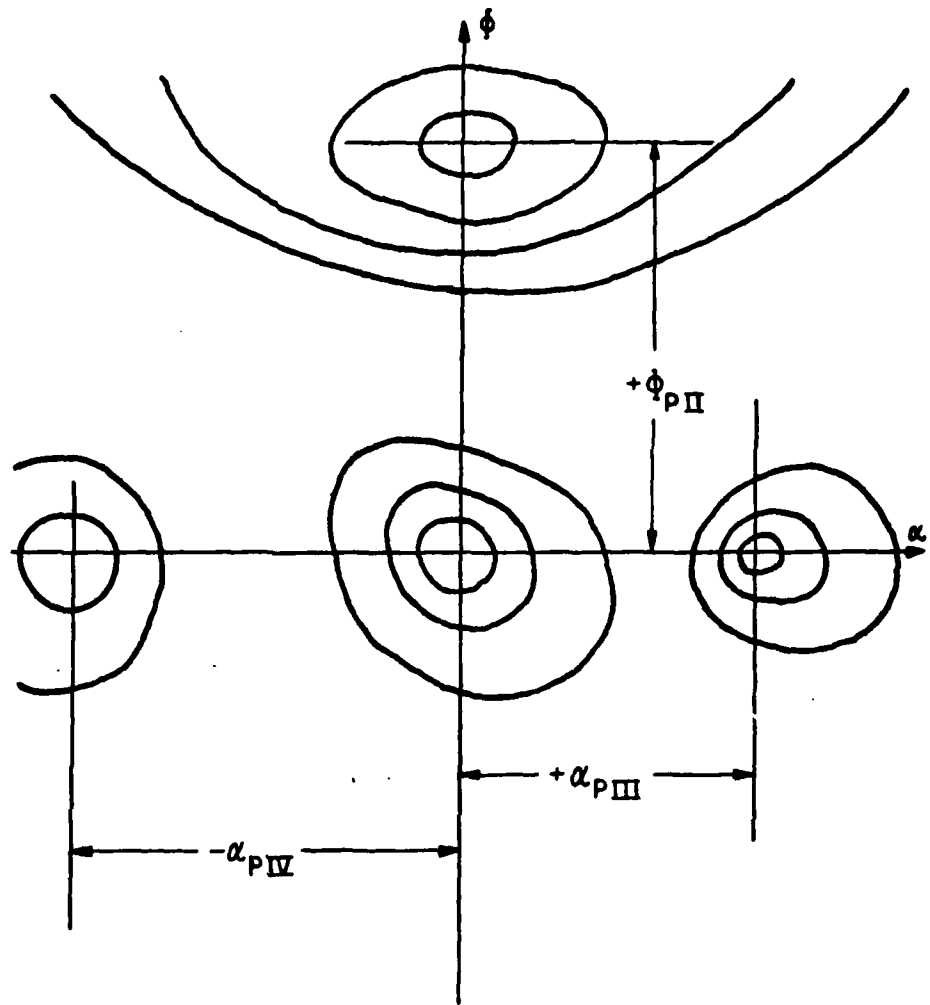


Fig. 4: Projections of $C_p = \text{const.}$ lines of the four calibration surfaces (the center hill is for probe A at $\alpha_p = 0$, $\phi_p = 0$, the top hill is for probe B at $\alpha_p = 0$, $\phi_p = \phi_{pII}$, the right hill is for probe A at $\alpha_p = +\alpha_{pIII}$, $\phi_p = 0$ and the left hill is for probe A at $\alpha_p = -\alpha_{pIV}$, $\phi_p = 0$).

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 α and ϕ 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 α and ϕ , are given in the following paragraph.

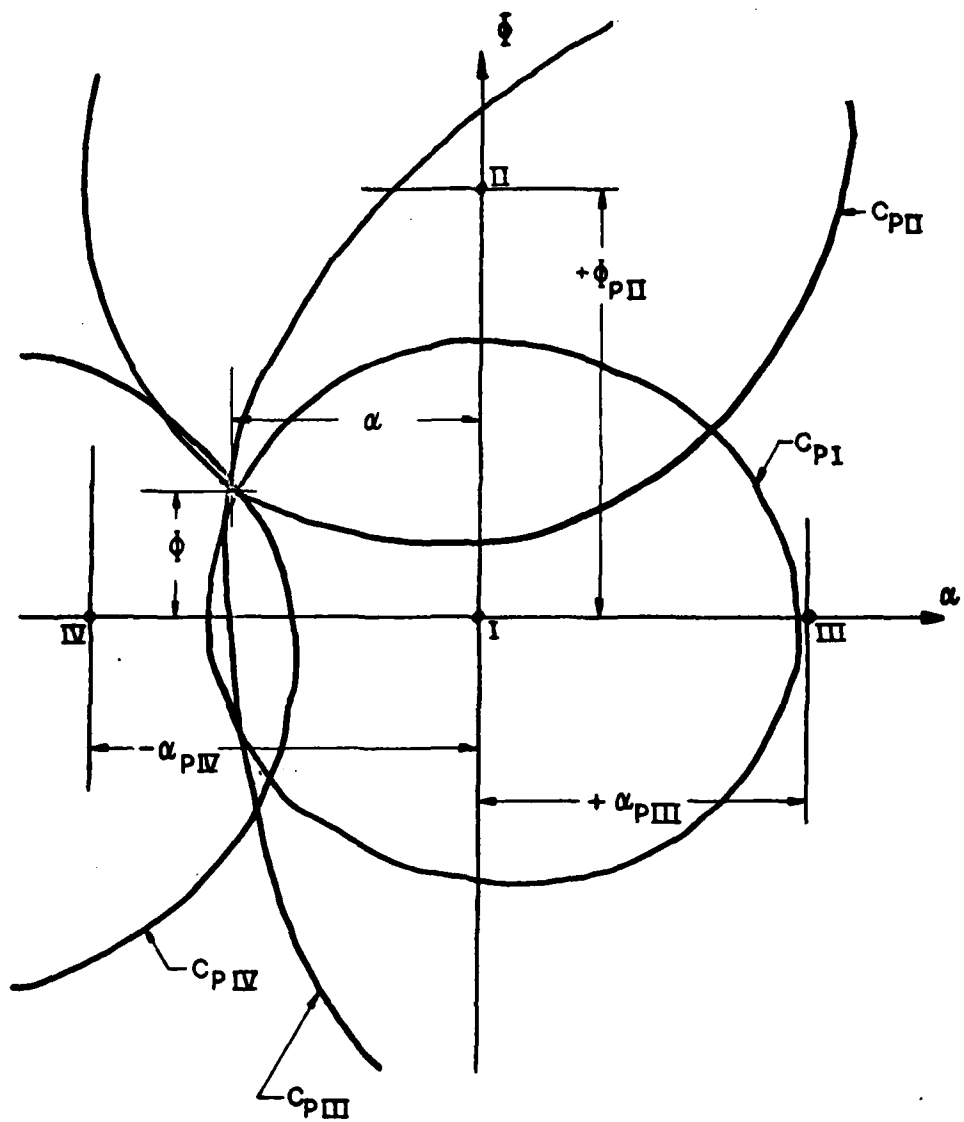


Fig. 5: An example showing a number of intersection points. The real one is in the second quadrant.

Evaluation of the intersection point coordinates (numerical Procedure)

The calibration surfaces Cp_{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 Cp_{rp} value in the string. The string starts with the value of Cp_{rp} at a point $(-\alpha_{rpLo}; -\phi_{rpLo})$ and ends with the value of Cp_{rp} at a point $(+\alpha_{rpUp}; +\phi_{rpUp})$. This sequence must be kept and can not be changed.

Each of the closed $Cp = \text{const.}$ curves projections in figs. 2 and 3 is the projection of the line of intersection between a plane parallel to the α, ϕ plane at a height equal to Cp 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 α, ϕ plane at a height Cp 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 $Z = Cp$) 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 $Cp = Cp(\alpha, \phi)$).

The subroutine has the following limitations:

- 1) No roots can be found on the lower $Y = \text{const.}$ boundary

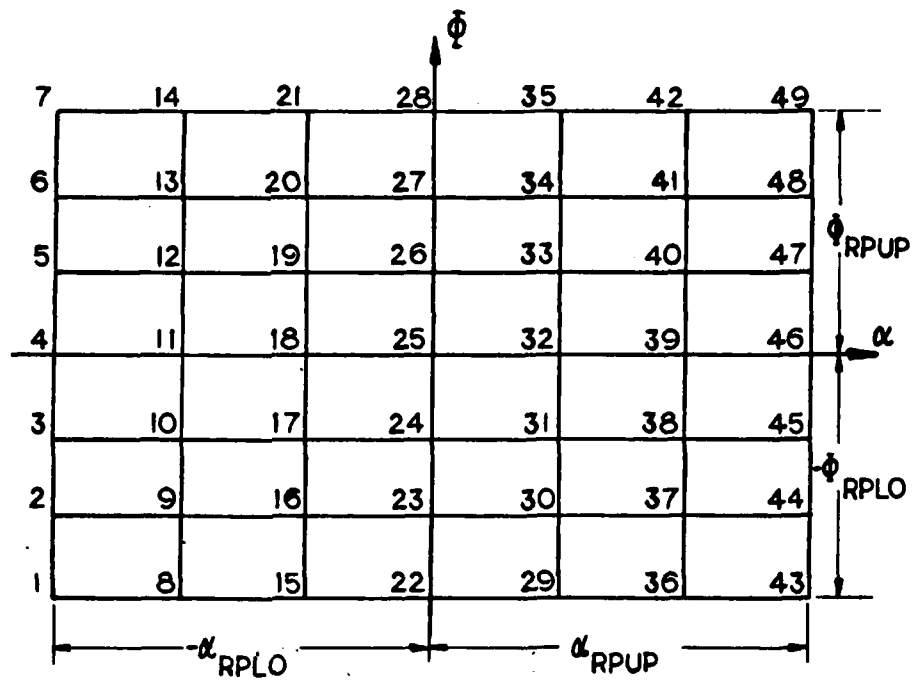


Fig. 6: The order of a calibration surface array.

- 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_{\min} to X_{\max} and from Y_{\min} to Y_{\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_{\min} to X_{\max} and from Y_{\min} to Y_{\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 $(ZX(I-1) - HT)/(ZX(I) - HT) < 0$. 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 Cp in the Fortran rotation used in this program.

$$\begin{aligned}
 XS = & X(I-1) + ABS(ZX(I-1)-HT)*(XM-X(I-1))/(ABS(HT-ZM) + \\
 & + ABS(ZX(I-1)-HT)) \qquad \qquad \qquad (6)
 \end{aligned}$$

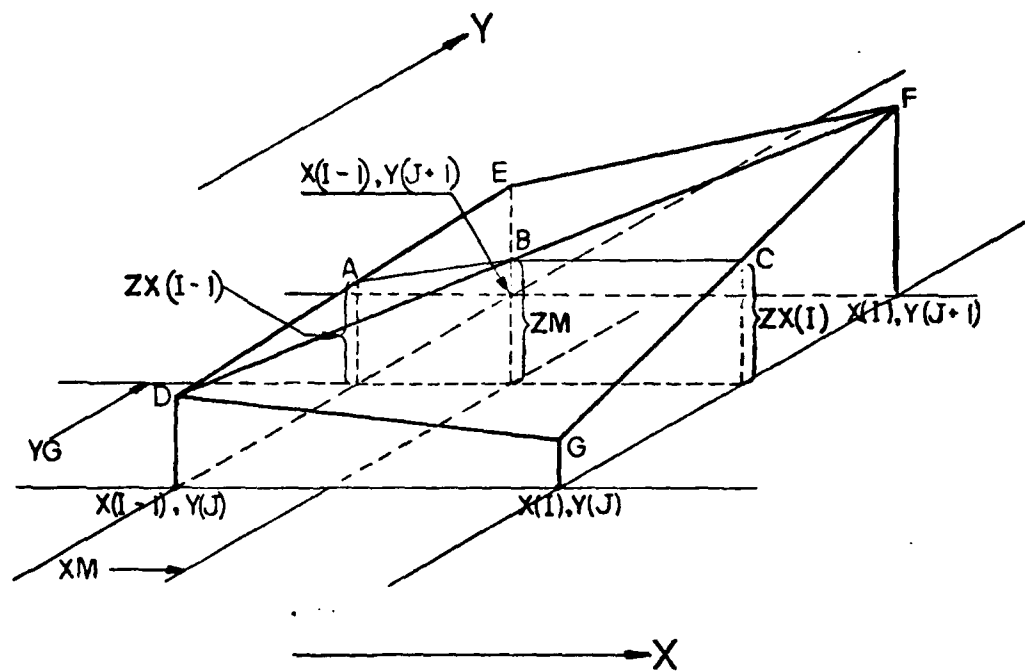


Fig. 7: The geometry of a linearized calibration surface element comprising two plane triangles and its intersection with a plane normal to X, Y along $Y = YG$.

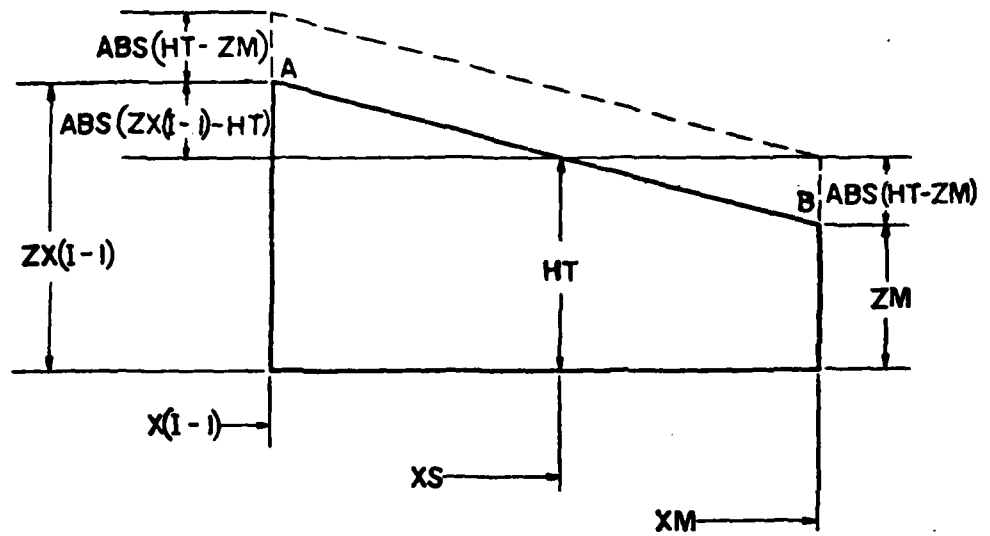


Fig. 8: The geometry for a penetration through a left hand (first) triangle.

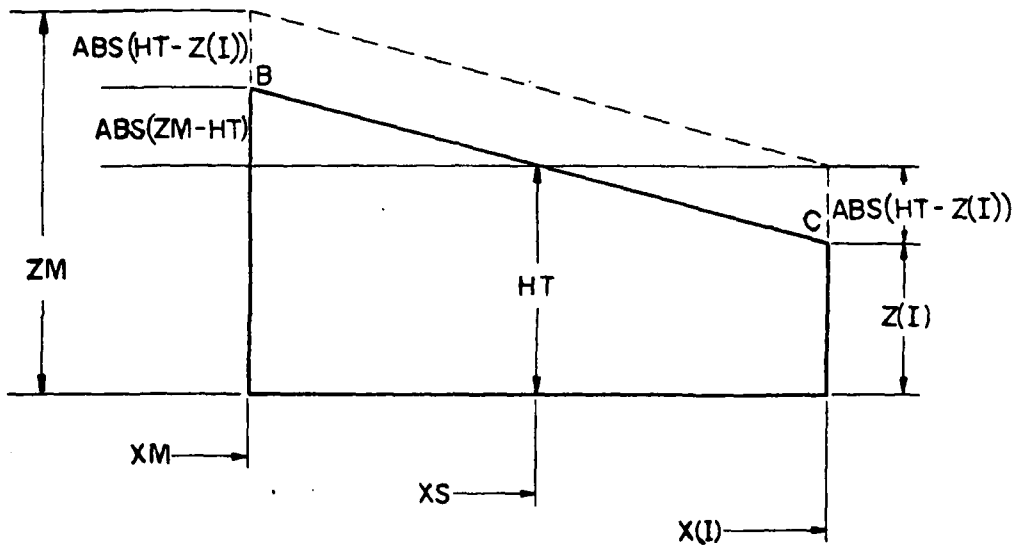


Fig. 9: The geometry for a penetration through a right hand (second) triangle.

$$XS = XM + (X(I)-XM)*ABS(ZM-HT)/(ABS(ZM-HT) + ABS(HT-Z(I))) \quad (7)$$

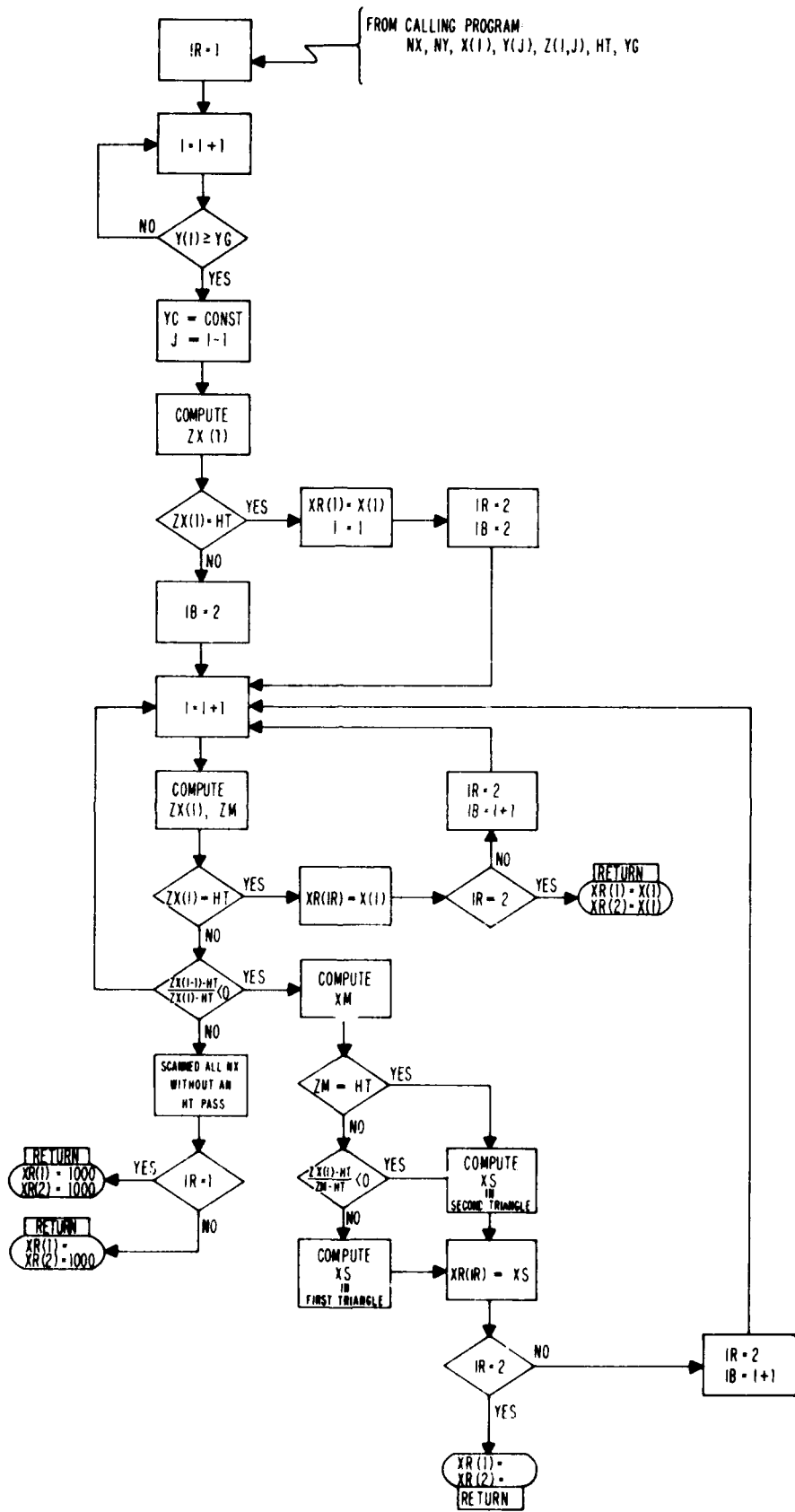
Equations 6 and 7 are invariant to the slope of the calibration surface i.e., the slope of the straight lines AB and BC.

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 ϕ) 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:

AJL(RAJL,YRJ)	BJL(RBJL,YRJ)
BJR(RBJR,YRJ)	AJR(RAJR,YRJ)
BIL(RBIL,YR)	AIL(RAIL,YR)
AIR(RAIR,YR)	BIR(RBIR,YR)

* The 10,000 scans are governed by a program constant in the line "DO 140" and can be arbitrarily varied.

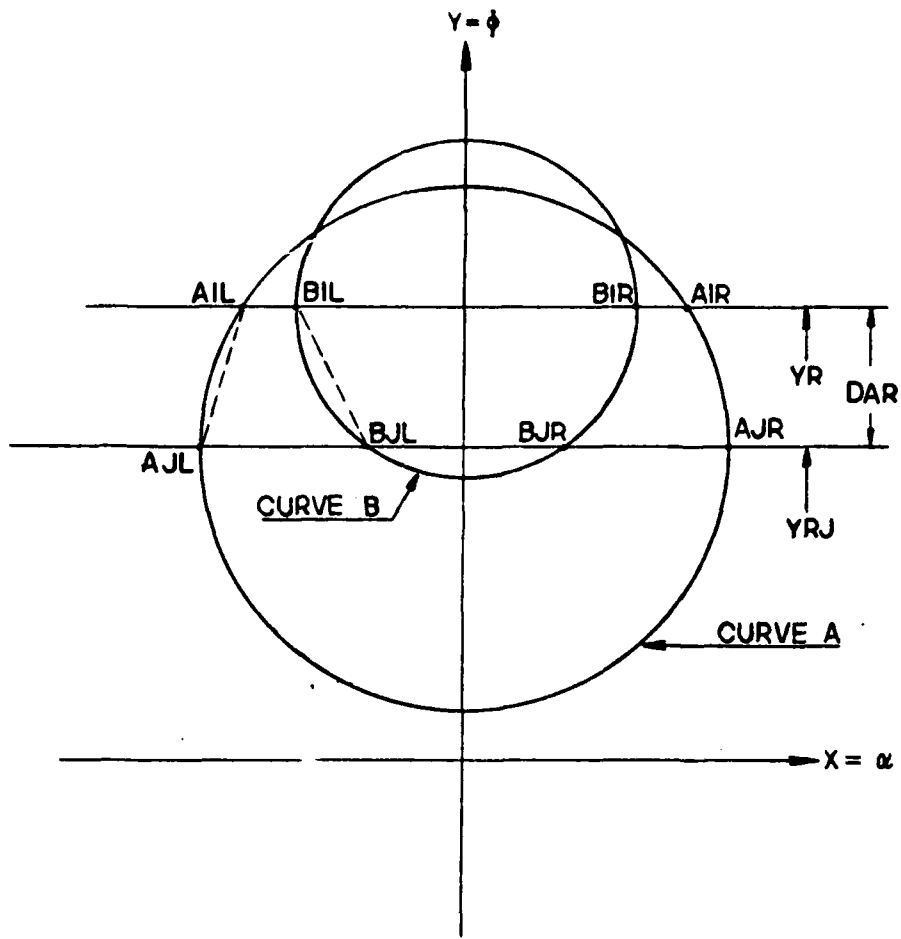


Fig. 11: The geometry for two successive scans when no intersection point exists.

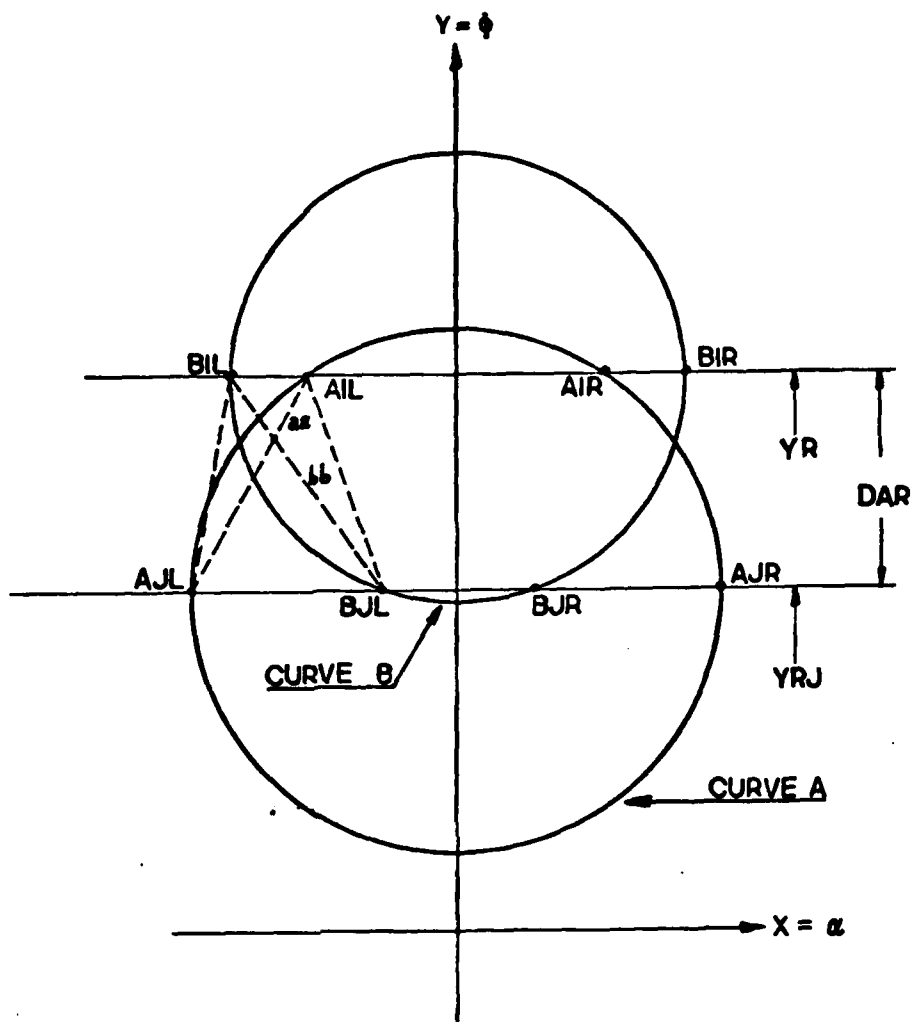


Fig. 12: The geometry for two successive scans when an intersection point does exist.

The left intersection point illustrated in fig. 12 is the intersection point of the lines aa and bb, each described by its equation:

$$\text{for aa } Y = (AA)X + BA \quad (8)$$

$$\text{for bb } Y = (AB)X + BB \quad (9)$$

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

$$AA = (YRJ-YR)/(RAJL - RAIL) \quad (10)$$

$$BA = YR - AA*RAIL \quad (11)$$

$$AB = (YRJ-YR)/(RBJL-RBIL) \quad (12)$$

$$BB = YR - AB*RBIL \quad (13)$$

The coordinates of the intersection point to be calculated are

$$X = (BB-BA)/(AA-AB) \quad (14)$$

$$Y = X*AA + BA \quad (15)$$

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 $C_p = \text{const.}$ curves are far from being tangent. But in

practice a situation of almost tangent curves can arise when X (or α) 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 ϕ to a maximum value of ϕ 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_{p_I} and $C_{p_{II}}$ of fig. 5 are evaluated.

However, in course of the reduction of the measured data, scans in the direction of α in steps of $\Delta\alpha$ are also necessary to evaluate the intersections of the curves C_{p_I} and $C_{p_{III}}$ or C_{p_I} and $C_{p_{IV}}$. 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 α and ϕ 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 (α, ϕ) 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

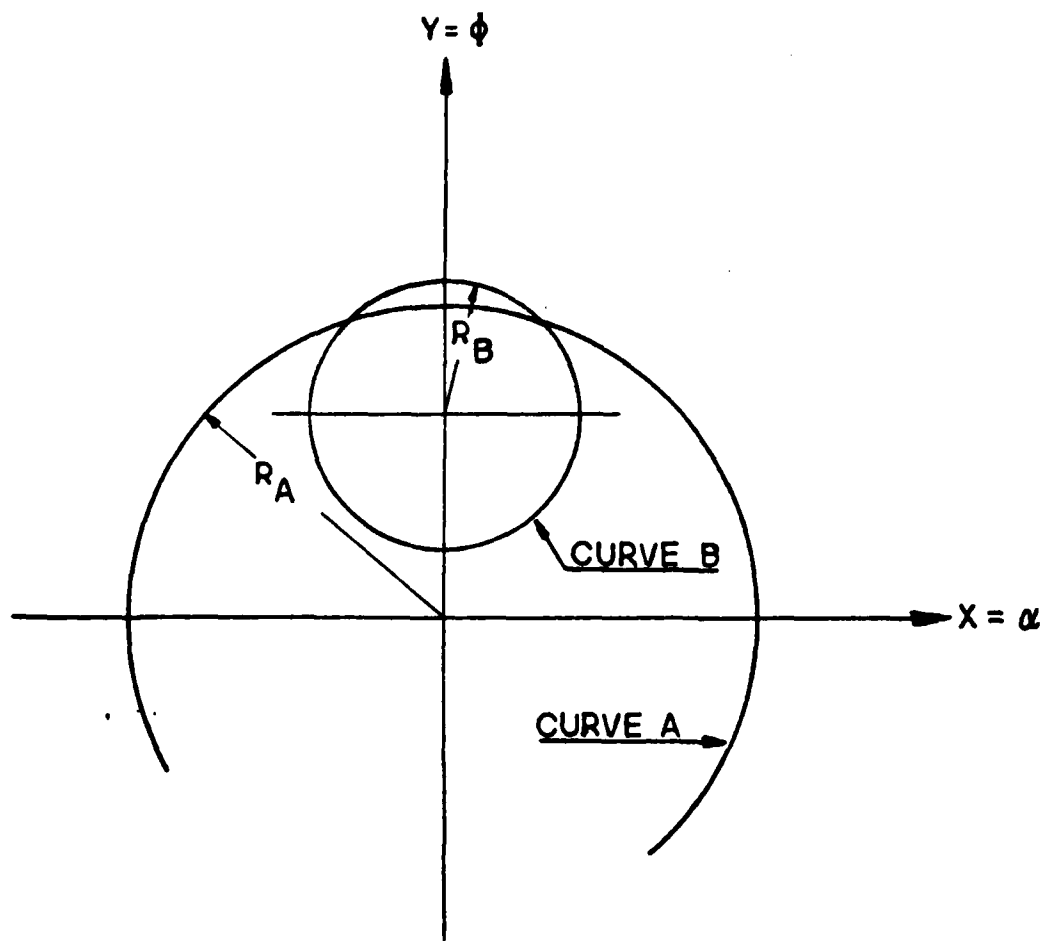


Fig. 13: The geometry when two closed $C_p = \text{const.}$ curves are almost tangent.

for the evaluation of α and ϕ 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 ϕ 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 ALF1, 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 Cp_{rp} 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

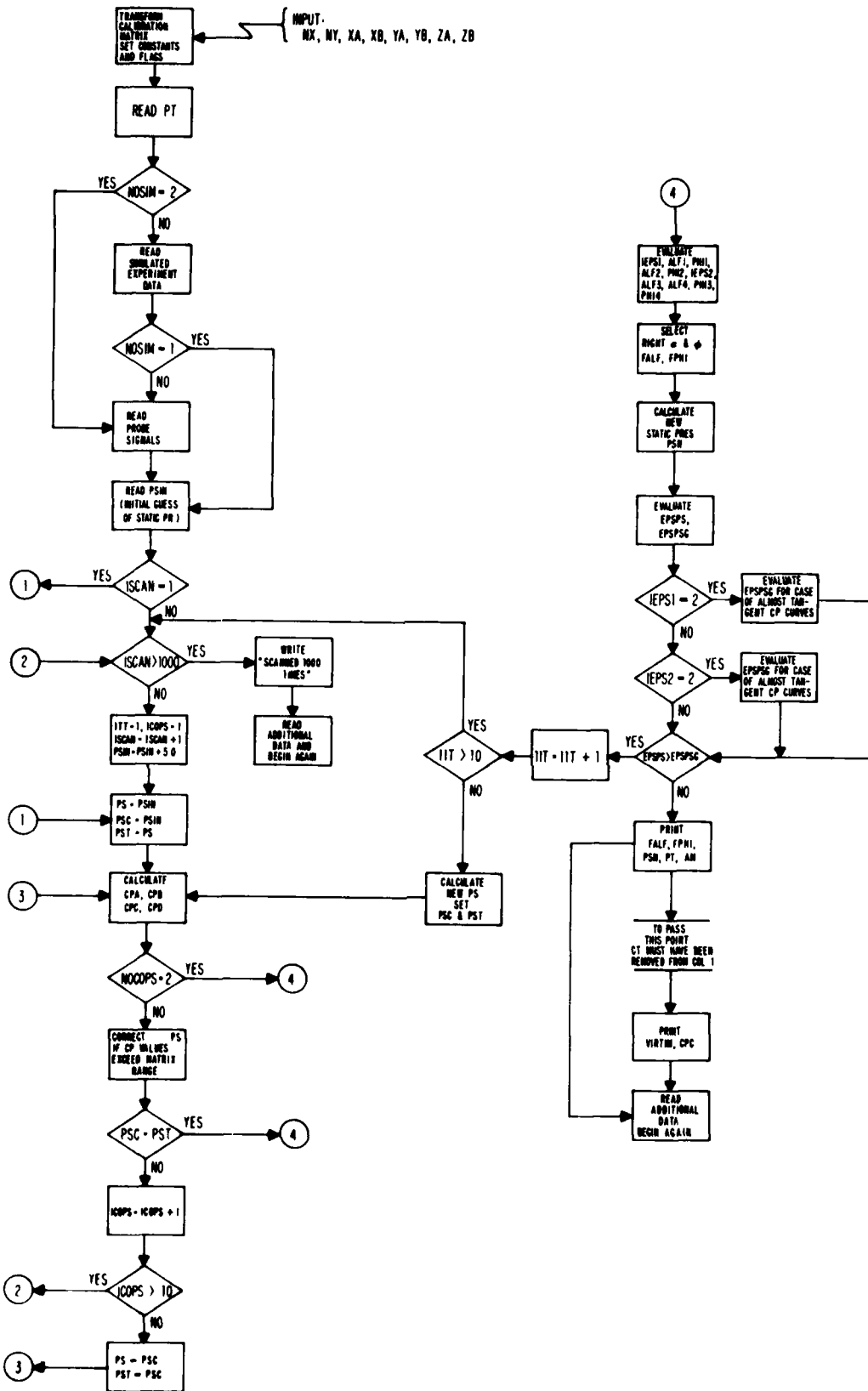


FIGURE 15. FLOW DIAGRAM OF VDR

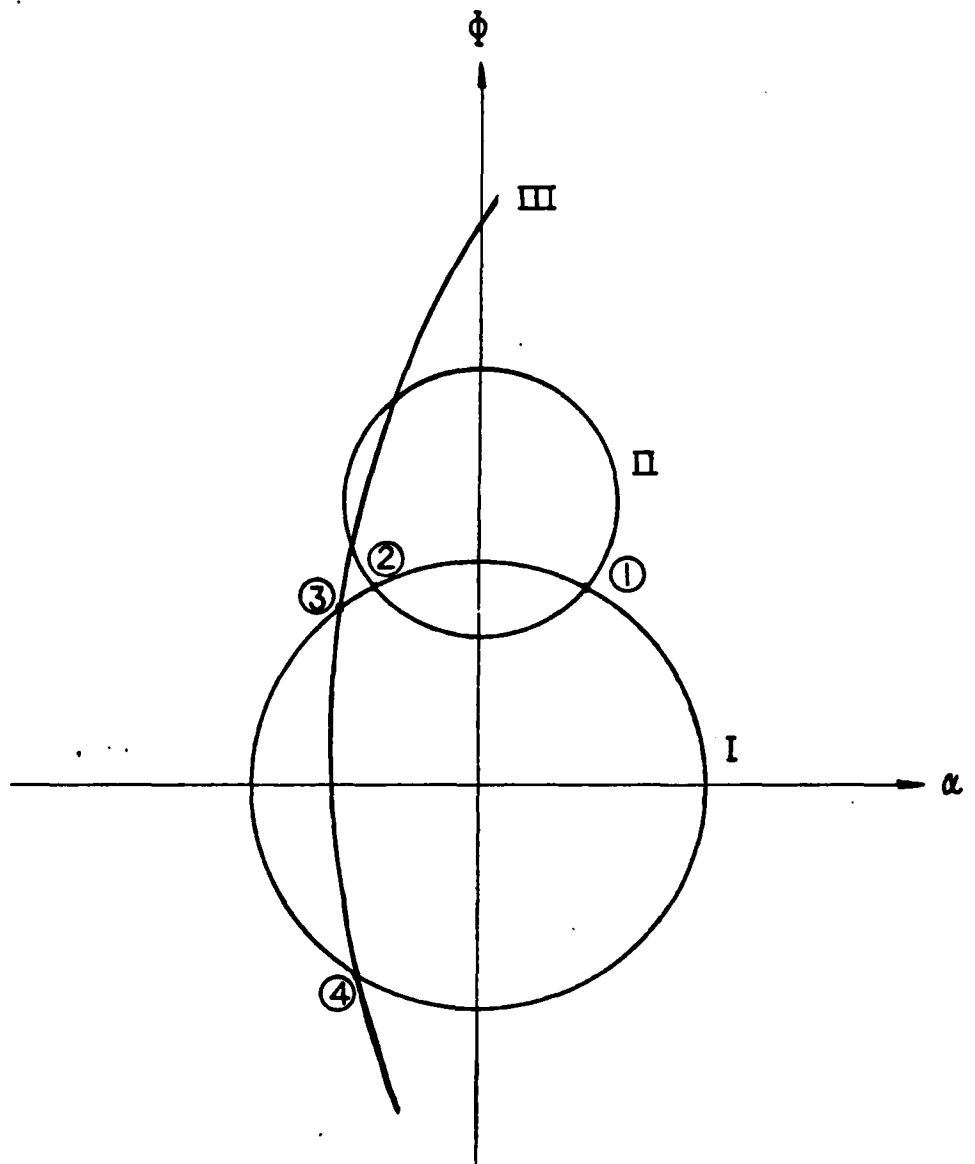


Fig. 16: Selection of the proper intersection point in VDR.

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 α and ϕ are temporarily known (see fig. 3). Using α and ϕ a new Cp_{rpIV} (for probe A in position IV) is computed by linear interpolation using subroutine INTPLT. With this new value of Cp_{rpIV} and with Pp_{IV} 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 ZA, ZB and their linearity. With coarser arrays convergence problem will increase and accuracy of results decrease.
- 2) Probe settings YPB, XPC, XPD (with the following probe settings being fixed and not variable YPA = YPC = YPD = 0.0; XPA = XPB = 0.0). The optimum setting is about $+20^\circ$ to 25° . 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 σ 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

α , ϕ , 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: $YPB = 250$, $XPC = 250$, $XPB = -250$. 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

1. Dunker R. J., Strinning P. E., and Weyer, H. B., "Experimental Study of the Flow Field Within a Transonic Axial Compressor Rotor by Laser Velocimetry and Comparison With Through-Flow Calculations", ASME Journal of Engineering for Power, Vol. 100, pp. 279-286, April 1978.
2. Shreeve, P. P., Simmons J. M., Winters K. A., and West J. C. Jr., "Determination of Transonic Compressor Flow Field by Synchronized Sampling of Stationary Fast Response Transducers", Symposium on Non-Steady Fluid Dynamics, ASME 1978 Winter Annual Meeting, San Francisco, Dec 1978. (To be published in ASME Journal of Fluids Engineering)
3. Thompkins W. T. Jr., and Kerrebrock J. L., "Exit Flow From a Transonic Compressor Rotor", AGARD Conference Proceedings No. 177, Unsteady Phenomena in Turbomachinery, pp. 6-1 to 6-23. Meeting held at Naval Postgraduate School, Monterey, California, 22-26 September 1975.
4. Shreeve R. P., McGuire A. G., and Hammer J. A., "Calibration of a Two Probe Synchronized Sampling Technique for Measuring Flows Behind Rotors", paper to be presented at IEEE, Eighth International Congress in Instrumentation in Aerospace Simulation Facilities, Naval Postgraduate School, Monterey, September 24-26 1979. Published as IEEE ICIA SF Record of Proceedings.

APPENDIX I

LISTING OF VDR

07/08/79 18.43.34

FILE: VDR FORTRAN P1

N P S

```

C START OF VDR
CT INTECFR A(6)
CT IASC=0
CT IACI=C
DIMENS ION ALF(10),PHI(10),XA(40),YA(40),
1ZA(40,40),XB(40),YB(40),ZB(40,40),XAX(40),YAX(40),
2ZAX(40,40),XBX(40),YBX(40),ZBX(40,40)
119 FORMAT(10,1)
131 FORMAT(F7.1)
130 FORMAT(F5.1)
120 FORMAT(F8.5)
132 FORMAT(F10.5)
135 FORMAT(F15.5)
133 FORMAT(F17.5)
C FEAD PROBE CALIBRATION DATA
READ(2,119) NA,NY
DO 121 I=1,NX
121 READ(2,130) XA(I)
DO 122 I=1,NY
122 READ(2,130) YA(I)
DO 123 J=1,NX
123 READ(2,120) ZAX(I,J)
DO 221 I=1,NX
221 READ(2,130) XB(I)
DO 222 I=1,NY
222 READ(2,130) YB(I)
DO 223 J=1,NX
223 READ(2,120) ZBX(I,J)
C TRANSFORM CALIBRATION MATRICES
DO 224 I=1,NY
224 XAX(I)=XA(I)
DO 225 I=1,NX
225 YAX(I)=YA(I)
DO 226 J=1,NX
226 ZAX(I,J)=ZAX(I,J)
DO 227 I=1,NX
227 XBX(I)=XB(I)
DO 228 I=1,NX
228 YBX(I)=YB(I)
DO 229 J=1,NX
229 ZBX(I,J)=ZBX(I,J)
C FRCL SETTINGS
YPA=C.C
YPE=55.
XPA=0.0
XPC=20.
XPL=-2.
C PROGRAM CONSTANTS
YRLF=-3.
YRIN=-30.
XFLC=-3.
XRLP=30.
XRIN=-30.
ICCP=1
NOCIP=1
NOSIM=1
ILT=1
ISCA=1
RELXPS=3.5
C EXPERIMENT SIMULATION
401 FORMAT(F15.4)
193 READ(5,401) PT
WRITE(6,401) PT
IF(NCS14.EQ.0) GO TO 500
READ(5,401) PS
INT00010
INT00020
INTCC030
INT00040
INT00050
INT00060
INTCC070
INTCC080
INT00090
INTCC100
INTCC110
INTCC120
INTCC130
INTCC140
INT00150
INTCL160
INT00170
INT00180
INTCC190
INTCC140
INT00200
INT00210
INT00220
INTCC230
INTCC240
INT00250
INTCC260
INT00270
INT00280
INTCC290
INTCC300
INT00310
INTCC320
INT00330
INTCC340
INTCC350
INT00360
INTCC370
INT00380
INTCC390
INT00400
INT00410
INT00420
INTCC430
INT00440
INT00450
INTCC460
INT00470
INT00480
INT00490
INT00500
INTCC510
INTCC520
INT00530
INTCC540
INT00550
INT00560
INTCC570
INT00580
INTCC590
INT00600
INT00610
INTCC620
INTCC630
INT00640
INTCC650
INT00660
INT00670
INTCC680
INT00690
INTCC700

```

```

WRITE(6,401) PS
READ(5,401) ALFA
WRITE(6,401) ALFA
READ(5,401) PHII
WRITE(6,401) PHII
CALL INTPLT(XA,YA,ZA,ALFA,PHII,NX,NY,CPA)
ALFD=ALFA-APC
CALL INTPLT(XA,YA,ZA,ALFD,PHII,NX,NY,CPC)
ALFD=ALFA-APD
CALL INTPLT(XA,YA,ZA,ALFD,PHII,NX,NY,CPD)
PHIB=PHII-YPB
CALL INTPLT(XB,YB,ZB,ALFA,PHIB,NX,NY,CPB)
PDYN=PT-PS
PPA=CPA*PDYN+PS
PPB=CPB*PDYN+PS
PPC=CPC*PDYN+PS
PPD=CPD*PDYN+PS
C READ MEASUREMENTS DATA
50C IF(INCSA.EQ.1) GO TO 501
READ(5,135) PPA
WRITE(6,135) PPA
READ(5,135) PPB
WRITE(6,135) PPB
READ(5,135) PPC
WRITE(6,135) PPC
READ(5,135) PPD
WRITE(6,135) PPD
501 READ(5,135) PSIN
WRITE(6,135) PSIN
18C IF(ISCAN.EQ.1) GO TO 181
IF(ISCAN.GT.1000) GO TO 305
IT=1
ICUPS=1
ISCAN=ISCAN+1
FSIN=PSIN*5.0

181 PS=PSIN
FSC=FSIN
PST=PS
C CALCULATES PRESSURE COEFFICIENTS
30C PDYN=PT-PS
CPA=(PPA-PS)/PDYN
CPB=(PPB-PS)/PDYN
CPC=(PPC-PS)/PDYN
CPD=(PPD-PS)/PDYN
C CORRECTS PS ASSUMPTION TO ENSURE CP ARE IN CALIBRATION RANGE
IF(MTCPS.EQ.2) GO TO 301
ZAMAX=-10000.
CALL MAXXY(ZA,ZAMAX,NX,NY)
IF(CPA.GT.ZAMAX) PSC=1.01*PS
IF(CPC.GT.ZAMAX) PSC=1.01*PS
IF(CPD.GT.ZAMAX) PSC=1.01*PS
ZEMAX=-10000.
CALL MAXXY(ZB,ZEMAX,NX,NY)
IF(CPB.GT.ZEMAX) PSC=1.01*PS
ZAMIN=10000.
CALL MINXY(ZA,ZAMIN,NX,NY)
IF(CPA.LT.ZAMIN) PSC=C.99*PS
IF(CPC.LT.ZAMIN) PSC=C.99*PS
IF(CPD.LT.ZAMIN) PSC=C.99*PS
ZBMIN=10000.
CALL MINXY(ZB,ZBMIN,NX,NY)
IF(CPB.LT.ZBMIN) PSC=C.99*PS
IF(PSC.EQ.PST) GO TO 301
ICUPS=ICUPS+1
IF(ICUPS.GT.10) GO TO 180
PS=PSC
PST=PSC
GO TO 300
C CALCULATES THE ALF & PHI ANGLES FOR 'I' & 'II'

```

```

INT00710
INT00720
INT00730
INT00740
INT00750
INT00760
INT00770
INT00780
INT00790
INT00800
INT00810
INT00820
INT00830
INT00840
INT00850
INT00860
INT00870
INT00880
INT00890
INT00900
INT00910
INT00920
INT00930
INT00940
INT00950
INT00960
INT00970
INT00980
INT00990
INT01000
INT01010
INT01020
INT01030
INT01040
INT01050
INT01070
INT01080
INT01090
INT01100
INT01110
INT01120
INT01130
INT01140
INT01150
INT01160
INT01170
INT01180
INT01190
INT01200
INT01210
INT01220
INT01230
INT01240
INT01250
INT01260
INT01270
INT01280
INT01290
INT01300
INT01310
INT01320
INT01330
INT01340
INT01350
INT01360
INT01370
INT01380
INT01390
INT01400

```

FILE: VDR FCRTXN PI N D S

```

301 CALL INTSCS(YPA,YP3,YNLC,YFUP,YKIN,
1CPA,CPC,ALF,PHI,AA,YA,ZA,
2XA,YE,EA,AX,YAX,ZAX,XEX,YXA,ZEX,IX,NY,IEPS)
IEPS1=IEPS
ALF1=ALF(1)
ALF2=ALF(2)
PHI1=PHI(1)
PHI2=PHI(2)
C CALCULATES THE ALF & PHI ANGLES FOR 'I' & 'III'
CALL INTSCS(XPA,XPC,ALF3,ALF4,APUP,ARIN,
1CPA,CPC,PHI,ALF,XAX,YAX,ZAX,
2XA,YA,ZA,AA,YA,ZA,AA,YA,ZA,AA,YA,ZA,NY,NX,IEPS)
IEPS2=IEPS
ALF3=ALF(1)
ALF4=ALF(2)
PHI3=PHI(1)
PHI4=PHI(2)
C SELECTS THE PROPER ALF & PHI ANGLES OUT OF THE FOUR VALUES
FALF=(ALF1+ALF2+ALF3+ALF4)/4-AMAX1(ALF1,ALF2,ALF3,
1ALF4)-(AMIN1(ALF1,ALF2,ALF3,ALF4))/2
FPHI=(PHI1+PHI2+PHI3+PHI4)/4-AMAX1(PHI1,PHI2,PHI3,PHI4)
1-AMIN1(PHI1,PHI2,PHI3,PHI4))/2
C CALCULATES IEPS FROM DATA OF POSITION IV
ALFCN=FALE-XPU
CALL INFL(XA,YA,ZA,ALFCN,FPHI,NX,NY,CPU)
ALFCN=FALE-XPC
CALL INFL(XA,YA,ZA,ALFCN,FPHI,NX,NY,CPC)
PSN=(PPC*CPU-PPD*CPC)/(CPU-CPC)
C CONVERGENCE TESTS
EPSPS=.33*(EPSN-PS)/PS
EPSPSC=.0000000009*(PT**2.0)-.00000098*PT
IF(IEPS1-.0002) EPSPSC=.0000000009*(PT**2.0)-.0000006*PT
IF(IEPS2-.0002) EPSPSG=.0000000009*(PT**2.0)-.0000006*PT
IF(IEPSPS-.07*EPSPSG) GO TO 310
GO TO 303
11T=11T+1
IF(11T.GT.10) GO TO 180
PS=PS+RELAPS*(PSN-PS)
FSC=PS
PST=PS
GO TO 300
303 WRITE(6,171) FALF
171 FFORMAT('C', ' VELOCITY YAW IS' ,F15.2)
WRITE(6,172) FPHI
172 FFORMAT(' VELOCITY PITCH IS' ,F15.2)
WRITE(6,173) PS
173 FFORMAT(' STATIC PRESSURE IS' ,F15.3)
WRITE(6,174) PST
174 FFORMAT(' TOTAL PRESSURE IS' ,F15.3)
AM=SLFT(11T/PSN)**0.285714-1.0)*5.0)
WRITE(6,175) AM
175 FFORMAT(' ARCH NUMBER IS' ,F15.5)
CT CALL IACLN(1)
CT VIRTIM=(A(5)-A(6))/76800
CT CPL=A(6)-A(6)/76800
CT WRITE(6,450) VIRTIM
450 FFORMAT(' VIRTUAL TIME IS' ,F15.5)
CT WRITE(6,451) CPU
451 FFORMAT(' TOTAL CPU IS' ,F15.5)
CT IASC=A(5)
CT IACN=A(6)
CT GO TO 193
305 WRITE(6,307)
307 FFORMAT(' SCANNED 1000 TIMES')
GO TO 193
END
SUEROUTINE INTSCS(ARPA,ARD2,ARL3,ARUP,ARIN,
1HT1,HT2,RES1,RES2,X1,Y1,Z1,
2X2,Y2,Z2,AA,YA,ZA,AA,YA,ZA,AA,YA,ZA,N1,N2,IEPS)
DIMENSIV RES1(10),RES2(10),A(25(10)),A1(40),Y1(40),

```

```

INT01410
INTC1420
INTC1430
INTC1440
INTC1450
INTC1460
INTC1470
INTC1480
INTC1490
INTC1500
INTC1510
INTC1520
INTC1530
INTC1540
INTC1550
INTC1560
INTC1570
INTC1580
INTC1590
INTC1590
INTC1590
INTC1600
INTC1610
INTC1620
INTC1630
INTC1640
INTC1650
INTC1660
INTC1670
INTC1680
INTC1690
INTC1700
INTC1710
INTC1720
INTC1730
INTC1740
INTC1750
INTC1760
INTC1770
INTC1780
INTC1790
INTC1800
INTC1810
INTC1820
INTC1830
INTC1840
INTC1850
INTC1860
INTC1870
INTC1880
INTC1890
INTC1900
INTC1910
INTC1920
INTC1930
INTC1940
INTC1950
INTC1960
INTC1970
INTC1980
INTC1990
INTC2000
INTC2010
INTC2020
INTC2030
INTC2040
INTC2050
INTC2060
INTC2070
INTC2080
INTC2090
INTC2100

```

FILE: VUR FORTR: P1 N P S

```

121(40,40),X2(40),Y2(40),Z2(40,40),X3(40),Y3(40),
Z3(40,40),X4(40),Y4(40),Z4(40,40)
RES1(1)=5000.
RES1(2)=5000.
RES2(1)=9000.
RES2(2)=9000.
IEPS=1
ISL=1
ISP=1
CAR=1.
AR2R=AR1R
GO TO 152
150 AR2R=AR2R+CAR
AR2R=AR2R
152 ARM1=AR2R-ARP1
ARM2=AR2R-ARP2
C CHECK FOR LOWER CALIBRATION RANGE
IF (ARM1.LT.ARL) GO TO 150
IF (ARM2.LT.ARL) GO TO 150
K=1
C CALCULATES INITIAL PENETRATION POINTS
CALL PENPTS(N1,N2,X1,Y1,Z1,HT1,ARM1,ARRES)
RAJL=ARRES(1)
RAJR=ARRES(2)
CALL PENPTS(N1,N2,X2,Y2,Z2,HT2,ARM2,ARRES)
RBJL=ARRES(1)
RBJR=ARRES(2)
C CALCULATES SUCCESSIVE PENETRATION POINTS
CO 140 I=1,10000
AR2R=AR2R+CAR
ARM1=AR2R-ARP1
ARM2=AR2R-ARP2
C CHECK FOR UPPER CALIBRATION RANGE
IF (ARM1.GT.ARUP) GO TO 195
IF (ARM2.GT.ARUP) GO TO 195
CALL PENPTS(N1,N2,X1,Y1,Z1,HT1,ARM1,ARRES)
RAJL=ARRES(1)
RAJR=ARRES(2)
CALL PENPTS(N1,N2,X2,Y2,Z2,HT2,ARM2,ARRES)
RBJL=ARRES(1)
RBJR=ARRES(2)
IF (RAJL.EQ.1000.) ISL=10
IF (RAJR.EQ.1000.) ISL=10
IF (RBJL.EQ.1000.) ISR=10
IF (RBJR.EQ.1000.) ISR=10
IF (RAJL.EQ.1000.) ISL=10
IF (RAJR.EQ.1000.) ISR=10
IF (RBJL.EQ.1000.) ISL=10
IF (RBJR.EQ.1000.) ISR=10
C CHECK FOR DIRECT HIT BY LEFT INTERSECTION, EVALUATE IF REQUIRED
IF (ISL.NE.10) GO TO 164
ISL=1
GO TO 183
164 IF (RAJL.NE.RBIL) GO TO 180
RES1(K)=RBIL
RES2(K)=AR2R
IF (K.EQ.2) GO TO 155
K=2
GO TO 133
C CHECK FOR INTERMEDIATE LEFT INTERSECTION, EVALUATE IF REQUIRED
180 IF (ISL.NE.10) GO TO 160
ISL=1
GO TO 183
160 IF ((RAJL-RBIL)/(RAJL-RBIL).GT.C.O) GO TO 183
AA=(AR2R-AR2R)/(RAJL-RAJL)
BA=AR2R-AR2R
AB=(AR2R-AR2R)/(RBIL-RBIL)
BB=AR2R-AR2R
RES1(K)=(AA-AB)/(AA-AB)
RES2(K)=AA*RES1(K)+BA

```

```

INTC2110
INTC2120
INTC2130
INTC2140
INTC2150
INTC2160
INTC2170
INTC2180
INTC2190
INTC2200
INTC2210
INTC2220
INTC2230
INTC2240
INTC2250
INTC2260
INTC2270
INTC2280
INTC2290
INTC2300
INTC2310
INTC2320
INTC2330
INTC2340
INTC2350
INTC2360
INTC2370
INTC2380
INTC2390
INTC2400
INTC2410
INTC2420
INTC2430
INTC2440
INTC2450
INTC2460
INTC2470
INTC2480
INTC2490
INTC2500
INTC2510
INTC2520
INTC2530
INTC2540
INTC2550
INTC2560
INTC2570
INTC2580
INTC2590
INTC2600
INTC2610
INTC2620
INTC2630
INTC2640
INTC2650
INTC2660
INTC2670
INTC2680
INTC2690
INTC2700
INTC2710
INTC2720
INTC2730
INTC2740
INTC2750
INTC2760
INTC2770
INTC2780
INTC2790
INTC2800

```

FILE: VDR FORTRAN 91 N P S

```

IF(K.EQ.2) GO TO 155
K=2
C CHECK FOR DIRECT HIT OR RIGHT INTERSECTION, EVALUATE IF REQUIRED
183 IF (ISR.NE.10) GO TO 161
ISR=1
GO TO 163
161 IF (RA1R.NE.RB1R) GO TO 182
RES1(K)=RB1R
RES2(K)=A*ZR
IF(K.EQ.2) GO TO 155
K=2
GO TO 163
C CHECK FOR INTERMEDIATE RIGHT INTERSECTION, EVALUATE IF REQUIRED
182 IF (ISR.NE.10) GO TO 182
ISR=1
GO TO 163
162 IF ((FAJR-REJ)/ (PAJR-RB1R).GT.0.0) GO TO 163
AA=(AF2R-AR2R)/(RAJR-RA1R)
EA=AR2R-AA*RA1R
AB=(AF2R-AR2K)/(RAJR-RB1R)
RB=AR2R-AB*RA1R
RES1(K)=(ES-PA)/(AA-AB)
RES2(K)=AA*RES1(K)+EA
IF(K.EQ.2) GO TO 155
K=2
C RE-INITIATE
163 AR2R=AR2R
RA1R=RA1R
RAJR=RAJR
RB1R=RB1R
140 RB1R=RB1R
WRITE(6,210)
210 FORMAT(' LOOP 140 THROUGH, NO POINTS FOUND')
GO TO 155
195 IF (RES1(1).EQ.9000.0) GO TO 191
IF (RES1(2).EQ.9000.0) GO TO 191
IF (RES2(1).EQ.9000.0) GO TO 191
IF (RES2(2).EQ.9000.0) GO TO 191
GO TO 155
C EVALUATION OF INTERSECTIONS WHEN CURVES ARE ALMOST TANGENT
191 YG=0.0
IEPS=2
CALL PENPTS (I2,N1,X3,Y3,Z3,HT1,YG0,ARRES)
R1=ABS(ARRES(2)-ARRES(1))/2
CALL PENPTS (I2,N1,X4,Y4,Z4,HT2,YG0,ARRES)
R2=ABS(ARRES(2)-ARRES(1))/2
RES2(1)=((R1**2)-(R2**2)+(AR2**2))/(2.0*AR2)
RES2(2)=RES2(1)
RES1(1)=SIGN(ABS((R1**2)-(RES2(1)**2)))
RES1(2)=-RES1(1)
155 RETURN
END
SUBROUTINE PENPTS (IX,NY,X,Y,Z,HT,YG,XR)
DIMENSION Y(40),ZA(40),ZC(40,40),X(40),XR(10)
IA=1
C SEARCH FOR J OF LOWER Y LINE
DO 101 I=1,NY
IF (Y(I).GE.YG) GO TO 102
101 CONTINUE
102 J=I-1
C COORDINATE SEARCH FOR ZERO PASS
C NEXT 4 LINES ARE EXECUTED IN FIRST SEARCH ONLY
YC=(YG-Y(J))/(Y(J+1)-Y(J))
ZA(1)=(Z(I,J+1)-Z(I,J))*YC+Z(I,J)
IF (ZX(1).EQ.HT) GO TO 107
I3=2
112 DO 103 I=1,IX
ZX(I)=(Z(I,J+1)-Z(I,J))*YC+Z(I,J)
ZM=(Z(I,J+1)-Z(I-1,J))*YC-Y(J)/(Y(J+1)-Y(J))+Z(I-1,J)
IF (ZX(I).EQ.HT) GO TO 108

```

```

INTC2810
INT02820
INTC2830
INT02840
INTC2850
INT02860
INTC2870
INT02880
INTC2890
INT02900
INTC2910
INT02920
INTC2930
INT02940
INTC2950
INT02960
INTC2970
INT02980
INTC2990
INT03000
INTC3010
INT03020
INTC3030
INT03040
INTC3050
INT03060
INTC3070
INT03080
INTC3090
INT03100
INTC3110
INT03120
INTC3130
INT03140
INTC3150
INT03160
INTC3170
INT03180
INTC3190
INT03200
INTC3210
INT03220
INTC3230
INT03240
INTC3250
INT03260
INTC3270
INT03280
INTC3290
INT03300
INTC3310
INT03320
INTC3330
INT03340
INTC3350
INT03360
INTC3370
INT03380
INTC3390
INT03400
INTC3410
INT03420
INTC3430
INT03440
INTC3450
INT03460
INTC3470
INT03480
INTC3490
INT03500

```

FILE: VOP FORTRAN P1 N P S

```

103 IF((ZX(I-1)-HT)/(ZX(I)-HT).LT.0.0) GO TO 104
CONTINUE
GO TO 110
C CALCULATION OF PENETRATIONS IN EITHER TRIANGLE
104 XM=X(I-1)+X(I)-X(I-1)*(YG-Y(J))/(Y(J+1)-Y(J))
IF(ZM.EQ.HT) GO TO 105
IF((ZX(I)-HT)/(ZM-HT).LT.0.0) GO TO 105
DJ=ABS(HY-ZM)+ABS(ZX(I-1)-HT)
XS=X(I-1)+X(I)*ABS(ZX(I-1)-HT)/DJ
GO TO 110
105 XS=XM+(X(I)-XM)*ABS(ZM-HT)/(ABS(ZM-HT)+ABS(ZX(I)-HT))
GO TO 110
C LOGICAL PENETRATIONS ACCUMULATION
107 XR(1)=X(I)
I=1
GO TO 113
108 XR(IR)=X(I)
GO TO 111
110 XR(IR)=XS
111 IF(IR.EQ.2) GO TO 114
113 IR=2
IB=I+1
GO TO 114
116 IF(IR.EQ.1) GO TO 117
XR(2)=1000.0
GO TO 114
117 XR(1)=1000.0
XR(2)=1000.0
114 RETURN
END
SUBROUTINE INTPL(X,Y,Z,XG,YG,AX,NY,ZRES)
C LINEAR INTERPOLATION ON CALIBRATION SURFACE TO EVALUATE
C CP VALUE AT COORDINATES XG,YG.THE RESULT RETURNED IS ZRES
C DIMENSION X(40),Y(40),Z(40,40)
C SEARCH FOR I OF LEFT X LINE
DO 1 IC=1,NX
IF(X(IC).GE.XG) GO TO 2
CONTINUE
1 IC=1
C SEARCH FOR J OF LOWER Y LINE
DO 3 JC=1,NY
IF(Y(JC).GE.YG) GO TO 4
CONTINUE
3 JC=1
4 AL=(Y(J+1)-Y(J))/(X(I+1)-X(I))
BL=Y(J)-AL*X(I)
YCR=(AL*XG)+BL
IF(YG.EQ.YCR) GO TO 7
IF(YG.GT.YCR) GO TO 5
C RESULTING Z IS IN LOWER TRIANGLE
X3=X(I+1)
Y3=Y(J)
Z3=Z(I+1,J)
GO TO 6
C RESULTING Z IS IN UPPER TRIANGLE
5 X3=X(I)
Y3=Y(J+1)
Z3=Z(I,J+1)
C GENERAL CALCULATION: GOOD FOR BOTH TRIANGLES
6 X1=X(I)
Y1=Y(J)
Z1=Z(I,J)
X2=X(I+1)
Y2=Y(J+1)
Z2=Z(I+1,J+1)
AP=((Z1-Z3)*(Y1-Y2)-(Z1-Z2)*(Y1-Y3))/((X1-X3)*(Y1-Y2)-(X1-X2)
1*(Y1-Y3))
BP=((Z1-Z2)-AP*(X1-X2))/(Y1-Y2)
CP=Z1-AP*X1-BP*Y1
ZRES=AP*XG+BP*Y1+CP

```


APPENDIX 2

LISTING OF SWVDR

07/08/75 12.45.15

FILE: SAVER FORTRAN PL N P S

```
C START OF SAVER
DIMENS(11) ALF(10),PHI(10),XA(40),YA(40),
1ZA(40,40),XB(40),YB(40),ZB(40,40),XAX(40),YAX(40),
2ZA(40,40),XAX(40),YBX(40),ZBX(40,40)
119 FORMAT(2I5)
131 FORMAT(F7.1)
130 FORMAT(F5.1)
120 FORMAT(F8.0)
132 FORMAT(F10.5)
135 FORMAT(F15.0)
136 FORMAT(F17.0)
C READ PROJECT CALIBRATION DATA
READ(2,119) IX,IY
DO 121 I=1,IX
READ(2,130) XA(I)
121 CONTINUE
DO 122 I=1,IY
READ(2,130) YA(I)
122 CONTINUE
DO 123 I=1,IX
DO 123 J=1,IY
READ(2,120) ZA(I,J)
123 CONTINUE
DO 221 I=1,IX
READ(2,130) XB(I)
221 CONTINUE
DO 222 I=1,IY
READ(2,130) YB(I)
222 CONTINUE
DO 223 I=1,IX
DO 223 J=1,IY
READ(2,120) ZB(I,J)
223 CONTINUE
C TRANSFORM CALIBRATION MATRICES
DO 224 I=1,IY
XAX(I)=YA(I)
DO 225 I=1,IX
YAX(I)=XA(I)
DO 226 I=1,IX
DO 226 J=1,IY
ZAX(I,J)=ZB(J,I)
DO 227 I=1,IY
XB(I)=YE(I)
DO 228 I=1,IX
YB(I)=XB(I)
DO 229 I=1,IY
DO 229 J=1,IX
ZBX(J,J)=ZB(J,I)
C FRCBE SETTINGS
YPA=0.0
YPB=55.
XPA=0.0
XPC=20.
XPD=-20.
C FRCGRM CONSTANTS
YPLD=-30.
YRUP=30.
YRIN=-30.
XRLD=-30.
XRUP=30.
XRIN=-30.
ICOPS=1
NCCCF5=2
NOSIM=2
IIT=1
ISCAN=1
RELXPS=0.5
C EXPERIMENT SIMULATION
193 CONTINUE
401 FORMAT(F15.4)
```

```
INT00010
INT00020
INT00030
INT00040
INT00050
INT00060
INT00070
INT00080
INT00090
INT00100
INT00110
INT00120
INT00130
INT00140
INT00150
INT00160
INT00170
INT00180
INT00190
INT00200
INT00210
INT00220
INT00230
INT00240
INT00250
INT00260
INT00270
INT00280
INT00290
INT00300
INT00310
INT00320
INT00330
INT00340
INT00350
INT00360
INT00370
INT00380
INT00390
INT00400
INT00410
INT00420
INT00430
INT00440
INT00450
INT00460
INT00470
INT00480
INT00490
INT00500
INT00510
INT00520
INT00530
INT00540
INT00550
INT00560
INT00570
INT00580
INT00590
INT00600
INT00610
INT00620
INT00630
INT00640
INT00650
INT00660
INT00670
INT00680
INT00690
INT00700
```

FILE: SWDR FCRTAN P1 N P S

```

READ(5,401) PT
WRITE(6,401) PT
IF(INCSIM.EQ.2) GO TO 500
READ(5,401) PS
WRITE(6,401) PS
READ(5,401) ALFA
WRITE(6,401) ALFA
READ(5,401) PHII
WRITE(6,401) PHII
CALL INTPLT(XA,YA,ZA,ALFA,PHII,NX,NY,CPA)
ALFC=ALFA-APC
CALL INTFLT(XA,YA,ZA,ALFC,PHII,NX,NY,CPC)
ALFB=ALFA-APC
CALL INTFLT(XA,YA,ZA,ALFB,PHII,NX,NY,CPC)
PHIB=PHII-YPB
CALL INTPLT(XB,YB,ZB,ALFA,PHIB,NX,NY,CPB)
PDYN=PT-PS
PPA=CPA*PDYN+PS
PPB=CPB*PDYN+PS
PPC=CPC*PDYN+PS
PPD=CPD*PDYN+PS
C READ MEASUREMENTS DATA
500 IF(INOSIM.EQ.1) GO TO 501
REAL(5,135) PPA
WRITE(6,136) PPA
READ(5,135) PPB
WRITE(6,136) PPB
READ(5,135) PPC
WRITE(6,136) PPC
READ(5,135) PPD
WRITE(6,136) PPD
501 READ(5,135) PSIN
WRITE(6,136) PSIN
180 IF(ISCAN.EQ.1) GO TO 181
IF(ISCAN.EQ.1000) GO TO 305
II=1
ICCP=1
ISCAN=ISCAN+1
PSIN=PSIN+5.0
WRITE(6,136) PSIN
181 CONTINUE
PS=PSIN
PSC=PSIN
PST=PS
C CALCULATES PRESSURE COEFFICIENTS
300 CONTINUE
PDYN=PT-PS
CPA=(PPA-PS)/PDYN
CPB=(PPB-PS)/PDYN
CPC=(PPC-PS)/PDYN
CPD=(PPD-PS)/PDYN
C CORRECTS PS ASSUMPTION TO ENSURE CF ARE IN CALIBRATION RANGE
IF(INDCPS.EQ.2) GO TO 301
ZAMAX=-1000.
CALL MAXARY(ZA,ZAMAX,NX,NY)
IF(CPA.GT.ZAMAX) PSC=1.01*PS
IF(CPB.GT.ZAMAX) PSC=1.01*PS
IF(CPD.GT.ZAMAX) PSC=1.01*PS
ZBMAX=-1000.
CALL MAXARY(ZB,ZBMAX,NX,NY)
IF(CPB.GT.ZBMAX) PSC=1.01*PS
ZAMIN=1000.
CALL MINARY(ZA,ZAMIN,NX,NY)
IF(CPA.LT.ZAMIN) PSC=0.99*PS
IF(CPC.LT.ZAMIN) PSC=0.99*PS
IF(CPD.LT.ZAMIN) PSC=0.99*PS
ZBMIN=1000.
CALL MINARY(ZB,ZBMIN,NX,NY)
IF(CPB.LT.ZBMIN) PSC=0.99*PS
IF(PSC.EQ.PST) GO TO 301

```

```

INTC0710
INTC0720
INTC0730
INTC0740
INTC0750
INTC0760
INTC0770
INTC0780
INTC0790
INTC0800
INTC0810
INTC0820
INTC0830
INTC0840
INTC0850
INTC0860
INTC0870
INTC0880
INTC0890
INTC0900
INTC0910
INTC0920
INTC0930
INTC0940
INTC0950
INTC0960
INTC0970
INTC0980
INTC0990
INTC1000
INTC1010
INTC1020
INTC1030
INTC1040
INTC1050
INTC1060
INTC1070
INTC1080
INTC1090
INTC1100
INTC1110
INTC1120
INTC1130
INTC1140
INTC1150
INTC1160
INTC1170
INTC1180
INTC1190
INTC1200
INTC1210
INTC1220
INTC1230
INTC1240
INTC1250
INTC1260
INTC1270
INTC1280
INTC1290
INTC1300
INTC1310
INTC1320
INTC1330
INTC1340
INTC1350
INTC1360
INTC1370
INTC1380
INTC1390
INTC1400

```

```

ICOPS=ICPS+1
IF (ICPS.GT.10) GO TO 180
FS=PSC
PST=PSC
GO TO 300
301 CONTINUE
C CALCULATES I, II ALF & PHI ANGLES FOR 'I' & 'II'
CALL INTSCS (YPA, YP1, YRL1, YRUP, YKIN,
1CPA, CP1, ALF1, PHI1, XA, YA, ZA,
2XB, YB, ZB, X1X, Y1A, Z1A, X1X, Y1X, Z1X, NX, NY)
ALF1=ALF(1)
ALF2=ALF(2)
PHI1=PHI(1)
PHI2=PHI(2)
C CALCULATES I, II ALF & PHI ANGLES FOR 'I' & 'III'
CALL INTSCS (XPA, XPC, XRL1, XRUP, XRI1,
1CPA, CPC, PHI1, ALF1, XA, YA, ZA,
2XAX, YAX, ZAX, XA, YA, ZA, XA, YA, ZA, NY, NX)
ALF3=ALF(1)
ALF4=ALF(2)
PHI3=PHI(1)
PHI4=PHI(2)
C SELECTS THE PROPER ALF & PHI ANGLES OUT OF THE FOUR VALUES
FALF=(ALF1+ALF2+ALF3+ALF4)/4
FPHI=(PHI1+PHI2+PHI3+PHI4)/4
C CALCULATES I, II PS FROM DATA OF POSITION D AND CORRECTS PT
ALFEN=FALF-XPD
CALL INTPT (XA, YA, ZA, ALFEN, FPHI, NX, NY, CPC)
ALFEN=FALF-XPC
CALL INTPLT (XA, YA, ZA, ALFEN, FPHI, NX, NY, CPC)
PSN=(PC*PCJ-PPJ*PCF)/(CPU-CPJ)
907 WRITE(5,11) IIT, FALF, FPHI, PSN
911 FORMAT(' ITERATION VALUES', I5, 2F6.2, F10.2)
C CONVERGENCE TESTS
EPSPS=ABS((PSN-PS)/PS)
EPSPSG=.000070JOCU77*(PT**2.0)-.0000098*PT
IF (EPSPS.LE.2) EPSPSG=.000000009*(PT**2.0)-.0000038*PT
IF (EPSPS.LE.2) EPSPSG=.000000009*(PT**2.0)-.0000000*PT
IF (EPSPS.GT.EPSPSG) GO TO 310
GO TO 300
310 CONTINUE
IIT=IIT+1
IF (IIT.GT.10) GO TO 180
FS=FS+RELXPS*(PSN-PS)
PSC=PS
PST=PS
GO TO 300
303 WRITE(6,171) FALF
171 FORMAT('01', VELOCITY YAW IS', F15.2)
172 WRITE(6,172) FPHI
172 FORMAT(' VELOCITY PITCH IS', F15.2)
173 WRITE(6,173) PSN
173 FORMAT(' STATIC PRESSURE IS', F15.3)
174 WRITE(6,174) PT
174 FORMAT(' TOTAL PRESSURE IS', F15.3)
174 AM=SCF((PT/PSN)**0.285714-1.0)**5.0)
175 WRITE(6,175) AM
175 FORMAT(' MACH NUMBER IS', F15.5)
GO TO 193
305 WRITE(6,307)
307 FORMAT(' SCANNED 1000 TIMES')
GO TO 193
END
SUBROUTINE INTSCS (ARP1, ARP2, ARLD, ARUP, ARIN,
1HT1, HT2, RES1, RES2, XA, YA, Z1,
2X2, Y2, Z2, X3, Y3, Z3, X4, Y4, Z4, N1, N2)
C DIMENSION RES1(10), RES2(10), X1(40), Y1(40),
1Z1(40,40), X2(40), Y2(40), Z2(40,40), X3(40), Y3(40),

```

INTC1410
INTC1420
INTC1430
INTC1440
INTC1450
INTC1460
INTC1470
INTC1480
INTC1490
INTC1500
INTC1510
INTC1520
INTC1530
INTC1540
INTC1550
INTC1560
INTC1570
INTC1580
INTC1590
INTC1600
INTC1610
INTC1620
INTC1630
INTC1640
INTC1650
INTC1660
INTC1670
INTC1680
INTC1690
INTC1700
INTC1710
INTC1720
INTC1730
INTC1740
INTC1750
INTC1760
INTC1770
INTC1780
INTC1790
INTC1800
INTC1810
INTC1820
INTC1830
INTC1840
INTC1850
INTC1860
INTC1870
INTC1880
INTC1890
INTC1900
INTC1910
INTC1920
INTC1930
INTC1940
INTC1950
INTC1960
INTC1970
INTC1980
INTC1990
INTC2000
INTC2010
INTC2020
INTC2030
INTC2040
INTC2050
INTC2060
INTC2070
INTC2080
INTC2090
INTC2100

```

22J(40,40),X4(40),Y4(40),Z+(40,40)
RES1(1)=9000.
RES1(2)=9000.
RES2(1)=9000.
RES2(2)=9000.
ISL=1
ISR=1
UAR=1.
AR2R=ARIN
GC TC 152
150 AR2R=AR2R+UAK
AR2RJ=AK2P
152 ARV1=AR2R-ARP1
ARM2=AR2R-ARP2
C CHECK FOR LOWER CALIBRATION RANGE
IF (ARM1.LT.AKLO) GC TC 150
IF (ARM2.LT.AKLC) GC TC 150
K=1
C CALCULATES INITIAL PENETRATION POINTS
CALL PENPTS(N1,N2,X1,Y1,Z1,HT1,ARM1,ARRES)
RAJL=ARRES(1)
RAJR=ARRES(2)
CALL PENPTS(N1,N2,X2,Y2,Z2,HT2,ARM2,ARRES)
RBJL=ARRES(1)
RBJR=ARRES(2)
C CALCULATES SUCCESSIVE PENETRATION POINTS
DO 140 J=1,1000
AP2R=AR2R+UAK
ARM1=AR2R-ARP1
ARM2=AR2R-ARP2
C CHECK FOR UPPER CALIBRATION RANGE
IF (ARM1.GT.ARUP) GC TC 195
IF (ARM2.GT.ARUP) GC TC 195
CALL PENPTS(N1,N2,X1,Y1,Z1,HT1,ARM1,ARRES)
RAJL=ARRES(1)
RAJR=ARRES(2)
CALL PENPTS(N1,N2,X2,Y2,Z2,HT2,ARM2,ARRES)
RBJL=ARRES(1)
RBJR=ARRES(2)
IF (RAJL.EQ.1000.) ISL=10
IF (RAJR.EQ.1000.) ISR=10
IF (RBJL.EQ.1000.) ISL=10
IF (RBJR.EQ.1000.) ISR=10
IF (RAJL.EQ.1000.) ISL=10
IF (RAJR.EQ.1000.) ISR=10
IF (RBJL.EQ.1000.) ISL=10
IF (RBJR.EQ.1000.) ISR=10
C CHECK FOR DIRECT HIT TO LEFT INTERSECTION,EVALUATE IF REQUIRED
IF (ISL.NE.10) GC TO 164
ISL=1
GC TC 183
164 IF (RAJL.NE.RAJR) GO TO 180
RES1(K)=RAJL
RES2(K)=AR2R
IF (K.EQ.2) GO TO 155
K=2
GC TC 183
C CHECK FOR INTERMEDIATE LEFT INTERSECTION,EVALUATE IF REQUIRED
180 IF (ISL.NE.10) GO TO 160
ISL=1
GO TO 183
160 IF ((RAJL-RBJL)/(RAJL-RBIL).GT.0.0) GO TO 183
AA=(AR2RJ-AR2P1)/(RAJL-RAJL)
AB=(AR2RJ-AR2P1)/(REJL-RBIL)
BB=AR2K-A3*K*IL
RES1(K)=(AA-AB)/(AA-18)
RES2(K)=AA*RES1(K)+EA
IF (K.EQ.2) GO TO 155
K=2

```

INTC2110
INTJ2120
INTC2130
INTJ2140
INTC2150
INTJ2160
INTC2170
INTJ2180
INTC2190
INTJ2190
INTC2200
INTJ2210
INTC2220
INTJ2230
INTC2240
INTJ2250
INTC2260
INTJ2270
INTC2280
INTJ2290
INTC2300
INTJ2310
INTC2320
INTJ2330
INTC2340
INTJ2350
INTC2360
INTJ2370
INTC2380
INTJ2390
INTC2400
INTJ2410
INTC2420
INTJ2430
INTC2440
INTJ2450
INTC2460
INTJ2470
INTC2480
INTJ2490
INTC2500
INTJ2510
INTC2520
INTJ2530
INTC2540
INTJ2550
INTC2560
INTJ2570
INTC2580
INTJ2590
INTC2600
INTJ2610
INTC2620
INTJ2630
INTC2640
INTJ2650
INTC2660
INTJ2670
INTC2680
INTJ2690
INTC2700
INTJ2710
INTC2720
INTJ2730
INTC2740
INTJ2750
INTC2760
INTJ2770
INTC2780
INTJ2790
INTC2800

FILE: SWVR FORTRAN P1 N P S

C CHECK FOR DIRECT HIT OF RIGHT INTERSECTION, EVALUATE IF REQUIRED

```
185 IF (ISH.NE.10) GO TO 161
    ISH=1
    GO TO 163
161 IF (RA17.NE.RB1R) GO TO 182
    RES1(N)=K31R
    RES2(K)=AR2R
    IF (K.EQ.2) GO TO 189
    K=2
    GO TO 163
```

C CHECK FOR INTERMEDIATE RIGHT INTERSECTION, EVALUATE IF REQUIRED

```
182 IF (ISR.NE.10) GO TO 162
    ISR=1
    GO TO 163
162 IF ((RAJR-RBJR)/(PAIR-RBIR).GT.0.0) GO TO 163
    AA=(AR2R-AJ2R)/(RAJR-RA1R)
    EA=AR2R-AA*RA1R
    AB=(AR2R-AJ2R)/(RAJR-RB1R)
    EB=AR2R-AB*RB1R
    RES1(K)=(AE-EB)/(AA-AB)
    RES2(K)=EA+RES1(K)*RA
    IF (K.EQ.2) GO TO 155
    K=2
```

C RE-INITIATE

```
163 AR2R=AR2R
    RA1R=RA1R
    RAJR=RAJR
    RB1R=RB1R
    RBJR=RBJR
    C.INT INJE
140 WRITE(6,210)
210 FORMAT(' LJO# 140 THROUGH,NT POINTS FOUND')
    GO TO 155
195 IF (RES1(1).EQ.9000.0) GO TO 191
    IF (RES1(2).EQ.9000.0) GO TO 191
    IF (RES2(1).EQ.9000.0) GO TO 191
    IF (RES2(2).EQ.9000.0) GO TO 191
    GO TO 155
```

C EVALUATION OF INTERSECTIONS WHEN CURVES ARE ALMOST TANGENT

```
191 YG=U-U
    CALL PENPTS (N2,N1,X3,Y3,Z3,HT1,YC,ARRES)
    R1=ABS(ARRES(2)-ARRES(1))/2
    CALL PENPTS (N2,N1,X4,Y4,Z4,HT2,YG,ARRES)
    R2=ABS(ARRES(2)-ARRES(1))/2
    RES2(1)=((R1**2)-(R2**2)+(ARP2**2))/(2.0*ARP2)
    RES2(2)=RES2(1)
    RES1(1)=SIGN(ABS((R1**2)-(RES2(1)**2)))
    RES1(2)=-RES1(1)
```

```
155 RETURN
    END
    SUBROUTINE PENPTS (NX,NY,X,Y,Z,HT,YG,XR)
    DIMENSION Y(40),ZX(40),Z(40,40),X(40),XR(10)
```

C SEARCH FOR J OF LOWER Y LINE

```
DO 101 I=1,NY
    IF (Y(I).GE.YG) GO TO 102
101 CONTINUE
    J=I-1
102 C CHOOSE SCAN FOR ZERO PASS
    C NEXT 3 LINES ARE EXECUTED IN FIRST SEARCH ONLY
    YC=(Y(I)-Y(J))/(Y(J+1)-Y(J))
    ZX(I)=(Z(I,I+1)-Z(I,J))*YC+Z(I,J)
    IF (ZX(I).EQ.HT) GO TO 107
```

```
112 GO TO 5 I=IB,IK
    ZX(I)=(Z(I,J+1)-Z(I,J))*YC+Z(I,J)
    Z4=(Z(I,J+1)-Z(I-1,J))*(Y(J+1)-Y(J))+Z(I-1,J)
    IF (ZX(I).EQ.HT) GO TO 108
103 IF ((ZX(I)-HT)/(ZX(I)-RT).LT.0.0) GO TO 106
    CONTINUE
```

INT02610
INT02820
INT02830
INT02840
INT02850
INT02860
INT02870
INT02880
INT02890
INT02900
INT02910
INT02920
INT02930
INT02940
INT02950
INT02960
INT02970
INT02980
INT02990
INT03000
INT03010
INT03020
INT03030
INT03040
INT03050
INT03060
INT03070
INT03080
INT03090
INT03100
INT03110
INT03120
INT03130
INT03140
INT03150
INT03160
INT03170
INT03180
INT03190
INT03200
INT03210
INT03220
INT03230
INT03240
INT03250
INT03260
INT03270
INT03280
INT03290
INT03300
INT03310
INT03320
INT03330
INT03340
INT03350
INT03360
INT03370
INT03380
INT03390
INT03400
INT03410
INT03420
INT03430
INT03440
INT03450
INT03460
INT03470
INT03480
INT03490
INT03500

FILE: SWVDR FORTRAN P1

N P

S

```
GO TO 116
C CALCULATION OF PENETRATIONS IN EITHER TRIANGLE
104 XM=X(I-1)+(X(I)-X(I-1))*(YG-Y(J))/(Y(J+1)-Y(J))
    IF(ZM.EQ.HT) GO TO 105
    IF((ZX(I)-HT)/(ZM-HT).LT.0.0) GO TO 105
    DD=ABS(HT-ZM)+ABS(ZX(I)-HT)
    XS=(I-1)+(XM-X(I-1))*ABS(ZX(I)-HT)/DD
105 GO TO 110
    XS=M+(X(I)-XM)*ABS(ZM-HT)/(ABS(ZM-HT)+ABS(ZX(I)-HT))
    GO TO 110
C LOGICAL PENETRATIONS ACCUMULATION
107 XR(1)=X(I)
    I=1
    GO TO 113
108 XR(I)=X(I)
    GO TO 111
110 XR(I)=XS
111 IF(19.EQ.2) GO TO 114
112 IK=2
    IB=I+1
    GO TO 112
116 IF(19.EQ.1) GO TO 117
    XR(2)=1000.0
    GO TO 114
117 XR(1)=1000.0
    XR(2)=1000.0
114 RETURN
    ENC
SUBROUTINE INTPLT(X,Y,Z,XG,YG,NX,NY,ZRES)
C LINEAR INTERPOLATION ON CALIPER AT IN SURFACE TO EVALUATE
C CP VALUE AT CO-ORDINATES XG,YG. THE RESULT RETURNED IS ZRES
    DIMENSION A(40),Y(40),Z(40,40)
C SEARCH FOR I OF LEFT X LINE
    JO 1 IC=1,NX
    IF(X(IC).GE.XG) GO TO 2
1  CCNTINJE
    I=IC-1
C SEARCH FOR J OF LOWER Y LINE
    DO 3 JC=1,NY
    IF(Y(JC).GE.YG) GO TO 4
3  CCNTINJE
    J=JC-1
    AL=(Y(J+1)-Y(J))/(X(I+1)-X(I))
    BL=Y(J)-AL*X(I)
    YCR=(AL*XG)+BL
    IF(YG.EQ.YCR) GO TO 7
    IF(YG.GT.YCR) GO TO 5
C RESULTING Z IS IN LOWER TRIANGLE
    X3=X(I+1)
    Y3=Y(J)
    Z3=Z(I+1,J)
    GO TO 6
C RESULTING Z IS IN UPPER TRIANGLE
5  X3=X(I)
    Y3=Y(J+1)
    Z3=Z(I,J+1)
C GENERAL CALCULATION: GOOD FOR BOTH TRIANGLES
6  X1=X(I)
    Y1=Y(J)
    Z1=Z(I,J)
    X2=X(I+1)
    Y2=Y(J+1)
    Z2=Z(I+1,J+1)
    AP=((Z1-Z2)*(Y1-Y2)-(Z1-Z2)*(Y1-Y3))/(X1-X3)*(Y1-Y2)-(X1-X2)
    ZRES=AP*XG+BP*YG+C
    GO TO 100
C RESULTING Z IS ON DIVIDING LINE
```

INTC3510
INTC3520
INTC3530
INTC3540
INTC3550
INTC3560
INTC3570
INTC3580
INTC3590
INTC3600
INTC3610
INTC3620
INTC3630
INTC3640
INTC3650
INTC3660
INTC3670
INTC3680
INTC3690
INTC3700
INTC3710
INTC3720
INTC3730
INTC3740
INTC3750
INTC3760
INTC3770
INTC3780
INTC3790
INTC3800
INTC3810
INTC3820
INTC3830
INTC3840
INTC3850
INTC3860
INTC3870
INTC3880
INTC3890
INTC3900
INTC3910
INTC3920
INTC3930
INTC3940
INTC3950
INTC3960
INTC3970
INTC3980
INTC3990
INTC4000
INTC4010
INTC4020
INTC4030
INTC4040
INTC4050
INTC4060
INTC4070
INTC4080
INTC4090
INTC4100
INTC4110
INTC4120
INTC4130
INTC4140
INTC4150
INTC4160
INTC4170
INTC4180
INTC4190
INTC4200

APPENDIX 3

LISTING OF WVDR

07/08/75 16.45.53

FILE: WVLK PENTHAN F1 N P S

```

C START OF WVLK
DIMENSION ALF(10),PHI(10),XA(40),YA(40),
1ZA(40),ZB(40),XB(40),YB(40),ZP(40,40),XAX(40),YAX(40),
2ZAX(40,40),XBX(40),YBX(40),ZBX(40,40)
119 FORMAT(2I5)
131 FORMAT(F7.1)
130 FORMAT(F5.1)
120 FORMAT(F3.5)
132 FORMAT(F10.5)
135 FORMAT(F15.5)
136 FORMAT(F17.5)
C READ PRIME CALIBRATION DATA
READ(2,119) IA,NY
DO 121 I=1,NA
READ(2,120) XA(I)
CONTINUE
121 DO 122 I=1,NY
READ(2,130) YA(I)
CONTINUE
DO 123 I=1,NA
DO 123 J=1,NY
READ(2,120) ZA(I,J)
CONTINUE
DO 221 I=1,IX
READ(2,130) XB(I)
CONTINUE
DO 222 I=1,NY
READ(2,130) YB(I)
CONTINUE
DO 223 I=1,NA
DO 223 J=1,NY
READ(2,120) ZB(I,J)
CONTINUE
C TRANSFORM CALIBRATION MATRICES
DO 224 I=1,NY
XAX(I)=YA(I)
DO 225 I=1,IX
YAX(I)=XB(I)
DO 226 I=1,NY
DO 226 J=1,IX
ZAX(I,J)=ZA(I,I)
DO 227 I=1,NY
XBX(I)=YB(I)
DO 228 I=1,NA
YBX(I)=XB(I)
DO 229 I=1,NY
DO 229 J=1,NA
ZBX(I,J)=ZB(I,I)
C FRICHE SETTINGS
YPA=0.0
YPB=55.
XPA=0.0
XPC=20.
XFD=-2.0
C PROGRAM CONSTANTS
YPLC=30.
YRLP=30.
YRIN=30.
XPLC=30.
XRLP=30.
XRIN=30.
ICCP5=1
NJCUPS=2
NOSIM=1
IIT=1
ISCAN=1
RELAPS=J.0
C EXPERIMENT SAVLLSTACN
19= CONTINUE
401 FORMAT(F15.4)

```

```

INT00010
INT00020
INT00030
INT00040
INT00050
INT00060
INT00070
INT00080
INT00090
INT00100
INT00110
INT00120
INT00130
INT00140
INT00150
INT00160
INT00170
INT00180
INT00190
INT00200
INT00210
INT00220
INT00230
INT00240
INT00250
INT00260
INT00270
INT00280
INT00290
INT00300
INT00310
INT00320
INT00330
INT00340
INT00350
INT00360
INT00370
INT00380
INT00390
INT00400
INT00410
INT00420
INT00430
INT00440
INT00450
INT00460
INT00470
INT00480
INT00490
INT00500
INT00510
INT00520
INT00530
INT00540
INT00550
INT00560
INT00570
INT00580
INT00590
INT00600
INT00610
INT00620
INT00630
INT00640
INT00650
INT00660
INT00670
INT00680
INT00690
INT00700

```

```

READ(5,401) PT
WRITE(6,401) PT
IF (INCSIN.EQ.2) GO TO 500
READ(5,401) PS
WRITE(6,401) PS
READ(5,401) ALFA
WRITE(6,401) ALFA
READ(5,401) PHII
WRITE(6,401) PHII
CALL INTPLT(XA,YA,ZA,ALFA,PHII,NX,NY,CPA)
ALFC=ALFA-XFC
CALL INTPLT(XA,YA,ZA,ALFC,PHII,NX,NY,CPC)
ALFD=ALFA-XFC
CALL INTPLT(XA,YA,ZA,ALFD,PHII,NX,NY,CPD)
PHIR=PHII-YPS
CALL INTPLT(XA,YA,ZA,ALFA,PHIR,NX,NY,CPA)
WRITE(5,402) CPA,CPB,CPC,CPD
402 FORMAT('5F10.5')
PUYN=PT-PS
PPA=CPA*PUYN+PS
PPB=CPB*PUYN+PS
PPC=CPA*PUYN+PS
PPD=CPD*PUYN+PS
WRITE(5,403) PPA,PPB,PPC,PPD
403 FORMAT('4F10.2')
C READ MEASUREMENTS DATA
IF (INCSIN.EQ.1) GO TO 501
500 READ(5,125) PPA
WRITE(6,125) PPA
READ(5,125) PPB
WRITE(6,125) PPB
READ(5,125) PPC
WRITE(6,125) PPC
READ(5,125) PPD
WRITE(6,125) PPD
501 READ(5,126) PSIN
WRITE(6,126) PSIN
IF (ISCAN.EQ.1) GO TO 181
180 IF (ISCAN.EQ.1000) GO TO 305
IIT=1
ICGPE=1
ISCAN=ISCAN+1
FSIN=FSIN+5.0
WRITE(6,126) PSIN
181 CONTINUE
PS=PSIN
PSC=PS
PST=PS
C CALCULATES PRESSURE COEFFICIENTS
300 CONTINUE
WRITE(5,940) IIT
940 FORMAT('C',1) IIT-CALIBRATION NO. ',15)
PUYN=PT-PS
CPA=(PPA-PS)/PUYN
CPB=(PPB-PS)/PUYN
CPC=(PPC-PS)/PUYN
CPD=(PPD-PS)/PUYN
908 WRITE(5,912) CPA,CPB,CPC,CPD
912 FORMAT('4F10.5')
C CORRECTS PRESSURE COEFFICIENTS TO ENSURE CP ARE IN CALIBRATION RANGE
IF (ABS(CPA-1.0) > .2) GO TO 501
ZAMAX=-10000.
CALL MAAPR(ZA,ZAMAX,NX,NY)
IF (CPA.GT.ZAMAX) PSC=1.01*PS
IF (CPC.GT.ZAMAX) PSC=1.01*PS
IF (CPD.GT.ZAMAX) PSC=1.01*PS
ZAMAX=-10000.
CALL MAAPR(ZA,ZAMAX,NX,NY)
IF (CPE.GT.ZAMAX) PSC=1.01*PS
ZAMIN=10000.

```

```

INT00710
INT00720
INT00730
INT00740
INT00750
INT00760
INT00770
INT00780
INT00790
INT00800
INT00810
INT00820
INT00830
INT00840
INT00850
INT00860
INT00870
INT00880
INT00890
INT00900
INT00910
INT00920
INT00930
INT00940
INT00950
INT00960
INT00970
INT00980
INT00990
INT01000
INT01010
INT01020
INT01030
INT01040
INT01050
INT01060
INT01070
INT01080
INT01090
INT01100
INT01110
INT01120
INT01130
INT01140
INT01150
INT01160
INT01170
INT01180
INT01190
INT01200
INT01210
INT01220
INT01230
INT01240
INT01250
INT01260
INT01270
INT01280
INT01290
INT01300
INT01310
INT01320
INT01330
INT01340
INT01350
INT01360
INT01370
INT01380
INT01390
INT01400

```

```

CALL MINJARY(ZA,ZAMIN,NX,NY)
IF(CPA.LT.ZA1IN) PSC=C.99*PS
IF(CPC.LT.ZAMIN) PSC=C.99*PS
IF(CPD.LT.ZA1IN) PSC=C.99*PS
ZMIN=10000.
CALL MINIFY(ZB,ZEMIN,NX,NY)
IF(CPR.LT.ZMIN) PSC=C.99*PS
IF(PSC.EQ.PST) GO TO 301
ICPPS=ICPPS+1
963 WRITE(6,964) ICPPS
964 FORMAT(' ICPPS IS',I5)
IF(ICPPS.GT.10) GO TO 180
PS=PSC
PST=PSC
920 WRITE(6,921) PSC
921 FORMAT(' PSC IS',F20.5)
GO TO 300
301 CONTINUE
C CALCULATES TWO ALF & PHI ANGLES FOR 'I' & 'III'
CALL INTSCS(XPA,YP3,YRLI,YRUP,YRIN,
1CPA,CP2,ALF,PHI,AA,YA,ZA,
2XB,YB,ZB,XAX,YAX,ZAX,XEX,YEX,ZBX,NX,NY)
ALF1=ALF(1)
ALF2=ALF(2)
PHI1=PHI(1)
PHI2=PHI(2)
C CALCULATES TWO ALF & PHI ANGLES FOR 'I' & 'III'
CALL INTSCS(XPA,XPC,XRLO,XRUP,XRIN,
1CPC,CP1,ALF,PHI,AA,YA,ZA,
2XAX,YAX,ZAX,XAY,YA,ZA,NX,NY)
ALF3=ALF(1)
ALF4=ALF(2)
PHI3=PHI(1)
PHI4=PHI(2)
C SELECTS THE PROPER ALF & PHI ANGLES OUT OF THE FOUR VALUES
FALF=(ALF1+ALF2+ALF3+ALF4)-AMAX1(ALF1,ALF2,ALF3,
1ALF4)-AMIN1(ALF1,ALF2,ALF3,ALF4)/2
FPHI=(PHI1+PHI2+PHI3+PHI4)-AMAX1(PHI1,PHI2,PHI3,PHI4)
1-AMIN1(PHI1,PHI2,PHI3,PHI4)/2
900 WRITE(6,901)FALF,FPHI
901 FFORMAT(' SELECTED',2F10.2)
C CALCULATES NEW PS FROM DATA OF POSITION D AND CORRECTS PT
ALF0=FALF-XPD
CALL INTPLT(XA,YA,ZA,ALF0,FPHI,NX,NY,CPD)
ALF0=FALF-XPC
CALL INTPLT(XA,YA,ZA,ALF0,FPHI,NX,NY,CPC)
905 WRITE(6,910) CPC,CPD
910 FFORMAT(' NEW CPC & CPD CALLED',F15.5)
PSN=(PPC*CPD-PPD*CPC)/(LFD-CPC)
907 WRITE(6,911) FALF,FPHI,PST,PT
911 FFORMAT(' ITERATION VALUES',2F6.2,2F10.2)
C CONVERGENCE TESTS
DELPS=PSN-PS
950 WRITE(6,951) PS,PSN,DELPS
951 FFORMAT(' EPSPS VALUE',3F15.5)
952 EPSPS=ABS((PSN-PS)/PS)
WRITE(6,961) EPSPS
961 FFORMAT(' EPSPS IS',F25.15)
EPSPSG=.000000099*(PT**2.0)-.00000098*PT
IF(1EPSPS.GT.2) EPSPSG=.000000009*(PT**2.0)-.0000003*PT
IF(1EPSPS.GT.2) EPSPSG=.000000009*(PT**2.0)-.0000003*PT
IF(1EPSPS.GT.EPSPSG) GO TO 310
GO TO 302
310 CONTINUE
II=IIY+1
IF(II.GT.10) GO TO 305
PS=PS+RELXPS*(PSN-PS)
PSC=PS
PST=PS
924 WRITE(6,925) PS

```

INT01410
INT01420
INT01430
INT01440
INT01450
INT01460
INT01470
INT01480
INT01490
INT01500
INT01510
INT01520
INT01530
INT01540
INT01550
INT01560
INT01570
INT01580
INT01590
INT01600
INT01610
INT01620
INT01630
INT01640
INT01650
INT01660
INT01670
INT01680
INT01690
INT01700
INT01710
INT01720
INT01730
INT01740
INT01750
INT01760
INT01770
INT01780
INT01790
INT01800
INT01810
INT01820
INT01830
INT01840
INT01850
INT01860
INT01870
INT01880
INT01890
INT01900
INT01910
INT01920
INT01930
INT01940
INT01950
INT01960
INT01970
INT01980
INT01990
INT02000
INT02010
INT02020
INT02030
INT02040
INT02050
INT02060
INT02070
INT02080
INT02090
INT02100

FILE:	WVCF	FORTHAN	PL	N	P	S
525	FORMAT(' CURK PS IS ',F15.5)					INT02110
	GO TO 300					INT02120
303	WRITE(6,171) FALF,FPF1,PSN,PT					INT02130
171	FORMAT('3F15.2,F15.3)					INT02140
	GO TO 193					INT02150
305	WRITE(6,307)					INT02160
307	FORMAT(' SCANNED 100 TIMES')					INT02170
	GO TO 193					INT02180
	END					INT02190
	SUBROUTINE INTSCS(ARP1,ARP2,ARL0,ARUP,ARIN,					INT02200
	HT1,HT2,RES1,RES2,X1,Y1,Z1,					INT02210
	Z2,Y2,Z2,X3,Y3,Z3,X4,Y4,Z4,M1,M2)					INT02220
	DIVENSION RES1(10),RES2(10),ARRES(10),A1(40),Y1(40),					INT02230
	L1(40),X2(40),Y2(40),Z2(40),X3(40),Y3(40),					INT02240
	Z3(40),X4(40),Y4(40),Z4(40),40)					INT02250
	RES1(1)=9000.					INT02260
	RES1(2)=9000.					INT02270
	RES2(1)=9000.					INT02280
	RES2(2)=9000.					INT02290
	ISL=1					INT02300
	ISR=1					INT02310
	UAR=1.					INT02320
	AR2R=ARIN					INT02330
	GO TO 152					INT02340
150	AR2R=AR2R+UAR					INT02350
	AR2RJ=AR2R					INT02360
152	AR1=AR2R-AR1					INT02370
	AR2=AR2R-AR2					INT02380
	C CHECK FOR LOWER CALIBRATION RANGE					INT02390
	IF(ARM1.LT.ARL0) GO TO 150					INT02400
	IF(ARM2.LT.ARL0) GO TO 150					INT02410
	K=1					INT02420
	C CALCULATES INITIAL PENETRATION POINTS					INT02430
	CALL PENPTS(N1,N2,X1,Y1,Z1,HT1,ARM1,ARRES)					INT02440
	RAJL=ARRES(1)					INT02450
	RAJR=ARRES(2)					INT02460
	CALL PENPTS(N1,N2,X2,Y2,Z2,HT2,ARM2,ARRES)					INT02470
	RBJL=ARRES(1)					INT02480
	RBJR=ARRES(2)					INT02490
	C CALCULATES SUCCESSIVE PENETRATION POINTS					INT02500
	DO 140 I=1,1000					INT02510
	AR2R=AR2R+UAR					INT02520
	AR1=AR2R-AR1					INT02530
	AR2=AR2R-AR2					INT02540
	C CHECK FOR UPPER CALIBRATION RANGE					INT02550
	IF(ARM1.GT.ARUP) GO TO 195					INT02560
	IF(ARM2.GT.ARUP) GO TO 195					INT02570
	CALL PENPTS(N1,N2,X1,Y1,Z1,HT1,ARM1,ARRES)					INT02580
	RAIL=ARRES(1)					INT02590
	RAIR=ARRES(2)					INT02600
	CALL PENPTS(N1,N2,X2,Y2,Z2,HT2,ARM2,ARRES)					INT02610
	RBIL=ARRES(1)					INT02620
	RBIR=ARRES(2)					INT02630
	IF(RAJL.EQ.1000.) ISL=10					INT02640
	IF(REJL.EQ.1000.) ISL=10					INT02650
	IF(RBJR.EQ.1000.) ISR=10					INT02660
	IF(RAJR.EQ.1000.) ISR=10					INT02670
	IF(RAIL.EQ.1000.) ISL=10					INT02680
	IF(RBIL.EQ.1000.) ISL=10					INT02690
	IF(RAIR.EQ.1000.) ISR=10					INT02700
	IF(RBIR.EQ.1000.) ISR=10					INT02710
	C CHECK FOR DIRECT HIT TO LEFT INTERSECTION, EVALUATE IF REQUIRED					INT02720
	IF(ISL.NE.10) GO TO 164					INT02730
	ISL=1					INT02740
	GO TO 183					INT02750
164	IF(RAIL.NE.RBIL) GO TO 180					INT02760
	RES1(K)=RAIL					INT02770
	RES2(K)=RAJR					INT02780
	IF(K.EQ.2) GO TO 155					INT02790
	K=2					INT02800


```

101 CONTINUE
102 J=J-1
C COARSE SCAN FOR ZERO PASS
C NEXT 4 LINES ARE EXECUTED IN FIRST SEARCH ONLY
YC=(Y(J)-Y(J+1))/(Y(J+1)-Y(J))
ZX(1)=(Z(1,J+1)-Z(1,J))*YC+Z(1,J)
IF(ZX(1).EQ.HT) GO TO 107
112 DO 103 I=1,NX
ZX(1)=(Z(1,J+1)-Z(1,J))*YC+Z(1,J)
ZM=(Z(1,J+1)-Z(1-1,J))*(Y(J+1)-Y(J))+Z(1-1,J)
IF(ZX(1).EQ.HT) GO TO 108
IF((ZX(1)-HT)/(ZX(1)-HT).LT.C.O) GO TO 104
103 CONTINUE
GO TO 116
C CALCULATION OF PENETRATIONS IN EITHER TRIANGLE
104 XM=(X(1)-X(1-1))*(Y(J+1)-Y(J))
IF(ZM.EQ.HT) GO TO 105
IF((X(1)-HT)/(ZM-HT).LT.J.O) GO TO 105
DD=ABS(XM-ZM)+ABS(ZX(1-1)-HT)
XS=X(1-1)+(X4-A(1-1))+ABS(ZX(1-1)-HT)/DD
GO TO 110
105 XS=X+(X(1)-XM)*ABS(ZM-HT)/(ABS(ZM-HT)+ABS(ZX(1)-HT))
GO TO 110
C LOGICAL PENETRATIONS ACCUMULATION
107 XR(1)=A(1)
I=1
GO TO 113
108 XR(I)=X(I)
GO TO 111
110 XR(IR)=XS
111 IF(IR.EQ.2) GO TO 114
113 IR=2
IR=I+1
GO TO 112
116 IF(IR.EQ.1) GO TO 117
XR(2)=1000.0
GO TO 114
117 XR(1)=1000.0
XR(2)=1000.0
114 RETURN
END
SUBROUTINE INTPLY(X,Y,Z,XS,YG,NX,NY,ZRES)
C LINEAR INTERPOLATION ON CALIPRATION SURFACE TO EVALUATE
C CP VALUE AT CJ-ORDINATES XS,YG. THE RESULT RETURNED IS ZRES
C DIMENSION X(40),Y(40),Z(40,40)
C SEARCH FOR I OF LEFT X LINE
DO 1 IC=1,NX
IF(X(1C).GE.XG) GO TO 2
CONTINUE
1 J=IC-1
C SEARCH FOR J OF LOWER Y LINE
DO 3 JC=1,NY
IF(Y(1C).GE.YG) GO TO 4
CONTINUE
4 J=JC-1
AL=(Y(J+1)-Y(J))/(X(I+1)-X(I))
BL=Y(J)-AL*X(I)
YCR=(AL*XS)+BL
IF(YG.EG.YCR) GO TO 7
IF(YG.GT.YCR) GO TO 5
C RESULTING Z IS IN LOWER TRIANGLE
X3=X(I+1)
Y3=Y(J)
Z3=Z(I+1,J)
GO TO 6
C RESULTING Z IS IN UPPER TRIANGLE
5 X3=X(I)
Y3=Y(J+1)
Z3=Z(I,J+1)

```

INT03510
INT03520
INT03530
INT03540
INT03550
INT03560
INT03570
INT03580
INT03590
INT03600
INT03610
INT03620
INT03630
INT03640
INT03650
INT03660
INT03670
INT03680
INT03690
INT03700
INT03710
INT03720
INT03730
INT03740
INT03750
INT03760
INT03770
INT03780
INT03790
INT03800
INT03810
INT03820
INT03830
INT03840
INT03850
INT03860
INT03870
INT03880
INT03890
INT03900
INT03910
INT03920
INT03930
INT03940
INT03950
INT03960
INT03970
INT03980
INT03990
INT04000
INT04010
INT04020
INT04030
INT04040
INT04050
INT04060
INT04070
INT04080
INT04090
INT04100
INT04110
INT04120
INT04130
INT04140
INT04150
INT04160
INT04170
INT04180
INT04190
INT04200

DISTRIBUTION LIST

	<u>No. of Copies</u>
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Office of Research Administration Code 012A Naval Postgraduate School Monterey, California 93940	1
4. Chairman Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
5. Director, Turbo-Propulsion Laboratory Department of Aeronautics Naval Postgraduate School Monterey, California 93940	30
6. Dr. H. J. Mueller Research Administrator Code 310A Naval Air Systems Command Navy Department Washington, D. C. 20360	1
7. Mr. Karl H. Guttman Code 330C Naval Air Systems Command Navy Department Washington, D. C. 20360	1
8. Mr. James R. Patton, Jr. Power Program, Code 473 Office of Naval Research Arlington, Virginia 22218	1
9. Commanding Officer Naval Air Propulsion Test Center Attn: Mr. Vernon Lubosky Trenton, New Jersey 08628	1

- | | | |
|-----|--|----|
| 10. | National Aeronautics and Sapce Administration
Lewis Research Center (Library)
2100 Brookpark Road
Cleveland, Ohio 44135 | 1 |
| 11. | CAG Library
The Boeing Company
Seattle, Washington 98124 | 1 |
| 12. | Library
General Electric Company
Aircraft Engine Technology Division
DTO Mail Drop H43
Cincinnati, Ohio 45215 | 1 |
| 13. | Library
Pratt and Whitney Aircraft
Post Office Box 2691
West Palm Beach, Florida 33402 | 1 |
| 14. | Library
Pratt and Whitney Aircraft
East Hartford, Connecticut 06108 | 1 |
| 15. | Chief, Fan and Compressor Branch
Mail Stop 5-9
NASA Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135 | 1 |
| 16. | Director, Whittle Laboratory
Department of Engineering
Cambridge University
ENGLAND | 1 |
| 17. | Prof. D. Adler
Technion Israel Institute of Technology
Department of Mechanical Engineering
Haifa 32000
Israel | 10 |
| 18. | Prof. A. E. Breugelmans
Institut von Karman de la Dynamique des Fluides
72 Chausee de Waterloo
1640 Rhode-St. Genese
Belgium | 1 |

19. Mr. Robert O. Bullock 1
Air Research Mfg. Corporation
Division of Garrett Corporation
402 South 36th Street
Phoenix, Arizona 85034
20. Dr. F. O. Carta 1
United Technologies Research Labs
400 Main Street
Hartford, Connecticut 06108
21. Prof. Jacques Chauvin 1
Universite D'Aix-Marseille
1 Rue Honnorat
Marseille, France
22. Mr. James V. Davis 1
Teledyne CAE
1330 Laskey Road
Toledo, Ohio 43601
23. Mr. Jean Fabri 1
ONERA
29, Ave. de la Division Leclerc
92 Chatillon
France
24. Prof. Dr. Ing Heinz E. Gallus 1
Lehrstuhl und Institut fur Strahltriebwerke und
Turbolabormaschinen
Rhein.-Westf. Techn. Hochschule Aachen
Templergraben 55
5100 Aachen, Germany
25. Professor J. P. Gostelow 1
School of Mechanical Engineering
The New South Wales Institute of Technology
Australia
26. DR. Ing. Hans-J. Heineman 1
DFVLR-AVA
Bunsenstrasse 10
3400 Gottingen, W. Germany
27. Prof. Ch. Hirsch 1
Vrije Universiteit Brussel
Pleinlaan 2
1050 Brussels, Belgium

28. Prof. J. P. Johnston 1
Stanford University
Department of Mechanical Engineering
Stanford, California 94305
29. Prof. Jack L. Kerrebrock, Chairman 1
Aeronautics and Astronautics Department
31-265 Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
30. Dr. B. Lakshminarayana 1
Professor of Aerospace Engineering
The Pennsylvania State University
233 Hammond Building
University Park, Pennsylvania 16802
31. Mr. R. A. Langworthy 1
Army Aviation Material Laboratories
Department of the Army
Fort Eustis, Virginia 23604
32. Dr. A. A. Mikolajczak 1
Pratt and Whitney Aircraft
Engineering 2H
East Hartford, Connecticut 06108
33. Prof. Dr. L. G. Napolitano 1
Director
Institute of Aerodynamics
University of Naples
Viale C. Augusto
80125 Napoli
Italy
34. Prof. Erik Nilsson 1
Institutionen for Stromningsmaskinteknik
Chalmers Tekniska Hogskola
Fack, 402 20 Goteborg 5
Sweden
35. Prof. Gordon C. Oates 1
Department of Aeronautics and Astronautics
University of Washington
Seattle, Washington 98105
36. Prof. Walter F. O'Brien 1
Mechanical Engineering Department
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

37. Prof. Dr. K. Oswatitsch 1
 Technische Hochschule
 Karlsplatz 13
 Vienna, Austria
38. Dr. P. A. Paranjee 1
 Head, Propulsion Division
 National Aeronautical Laboratory
 Post Bag 1799
 Bangalore - 17
 India
39. R. E. Peacock 1
 School of Mechanical Engineering
 Cranfield Institute of Technology
 Cranfield, Bedford MK43 0AL
 ENGLAND
40. Dr. Bruce A. Reese 1
 Director, Jet Propulsion Center
 School of Mechanical Engineering
 Purdue University
 Lafayette, Indiana 47907
41. Dr. W. Schlachter 1
 Brown, Boveri-Sulzer Turbomachinery Ltd
 Dept. TDE
 Escher Wyss Platz
 CH-8023 Zurich
 Switzerland
42. Dr. George K. Serovy 1
 Professor of Mechanical Engineering
 208 Mechanical Engineering Building
 Iowa State University
 Ames, Iowa 50010
43. Dr. Fernando Sisto 1
 Professor and Head of Mechanical Engineering Department
 Stevens Institute of Technology
 Castle Point, Hoboken, New Jersey 07030
44. Dr. Leroy H. Smith, Jr. 1
 Manager, Compressor and Fan Technology Operation
 General Electric Company
 Aircraft Engine Technology Division
 DTO Mail Drop H43
 Cincinnati, Ohio 45215

45. Dr. W. Tabakoff 1
Professor, Department of Aerospace Engineering
University of Cincinnati
Cincinnati, Ohio 45221
46. Mr. P. Tramm 1
Manager, Research Labs
Detroit Diesel Allison Division
General Motors
P. O. Box 894
Indianapolis, Indiana 46206
47. Prof. Dr. W. Traupel 1
institut für Thermische Turbomaschinen
Eidg. Technische Hochschule
48. Dr. Arthur J. Wennerstrom 1
ARL/LF
Wright-Patterson AFB
Dayton, Ohio 45433
49. Dr. H. Weyer 1
DFVLR
Linder Höhe
505 Porz-Wahn
Germany
50. Mr. P. F. Yaggy 1
Director
U. S. Army Aeronautical Research Laboratory
AMES Research Center
Moffett Field, California 94035