X USAF. Syst Com.
AFAPL-TR-68-142, Pt 8
PART VIII

EDR 5908 Pt 8

PROPULSION SYSTEM FLOW STABILITY PROGRAM (DYNAMIC),

PHASE I FINAL TECHNICAL REPORT,

PART VIII - MODEL FOR PREDICTING COMPRESSOR PERFORMANCE WITH COMBINED CIRCUMFERENTIAL AND RADIAL DISTORTION - COMPUTER PROGRAM

R. A. Novak, R. M. Hearsey, et al
NORTHERN RESEARCH AND ENGINEERING CORPORATION
= Gen Motors Corp. Allison Div

TECHNICAL REPORT AFAPL-TR-68-142, PART VIII

December 1968

F33615-67-C-1848

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of the Air Force Aero Propulsion Laboratory (APTA), Air Force Systems Command, Wright-Patterson Air Force Base, Ohio.

Air Force Aero Propulsion Laboratory
Air Force Systems Command
Wright-Patterson Air Force Base, Ohio

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government Procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AFAPL-TR-68-142
PART VIII

PROPULSION SYSTEM FLOW STABILITY PROGRAM (DYNAMIC)

PHASE I FINAL TECHNICAL REPORT

PART VIII. MODEL FOR PREDICTING COMPRESSOR PERFORMANCE WITH COMBINED CIRCUMFERENTIAL AND RADIAL DISTORTION - COMPUTER PROGRAM

R. A. Novak, R. Hearsey, et al

FOREWARD

This report describes work accomplished in Phase I of the two-phase program, "Propulsion System Flow Stability Program (Dynamic)" conducted under USAF Contract F33615-67-C-1848. The work was accomplished in the period from 20 June 1967 to 30 September 1968 by the Los Angeles Division of North American Rockwell Corporation, the prime Contractor, and the Subcontractors, the Allison Division of General Motors Corporation (supported by Northern Research and Engineering Corporation), the Autonetics Division of North American Rockwell Corporation (supported by the Aeronautical Division of Honeywell, Incorporated), and the Pratt & Whitney Aircraft Division of United Aircraft Corporation.

The program was sponsored by the Air Force Aero Propulsion Laboratory, Wright-Patterson Air Force Base, Ohio. Mr. H. J. Gratz, APTA, Turbine Engine Division, was the Project Engineer.

This volume is Part VIII of twenty parts and was prepared by the Northern Research and Engineering Corporation.

Publication of this report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.

Ernest C. Simpson
Chief, Turbine Engine Division

ABSTRACT

This report describes in detail a computer program designed to enable the performance of axial flow compressors operating under asymmetric inlet conditions to be predicted. The underlying theory is given in Volume VII; a comparison between results obtained and experimental data is made in Volume IX.

The report is divided into a number of sections, the principal ones being those giving a Functional Description of the program, and an Operational Description of the program. The Functional Description is intended to enable the program to be used. The Operational Description of the program is intended to enable the methods used in the program to be understood in detail, hence allowing the program to be modified if desired.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

Symbol	Description
C	Absolute velocity
C_r	Radial component of velocity
C_θ	Circumferential component of velocity
C_z	Axial component of velocity
C_m	Meridional component of velocity
H	Stagnation enthalpy
p	Static pressure
r	Radius
r_m	Radius of curvature of meridional projection of a streamline
S	Entropy
t	Static temperature
M	Mass flow
α	$\tan^{-1}\left(\frac{C_\theta}{C_m}\right)$
ϕ	$\tan^{-1}\left(\frac{C_r}{C_m}\right)$
μ	Angle defining sector centerline in r-θ-plane
γ	Angle made by computing station with r-axis in r-z-plane
θ	Circumferential location of streamline referenced to defined upstream station
m	Meridional location of streamline referenced to defined upstream station
$\bar{A}$	Direction cosine, radial component
$\bar{B}$	Direction cosine, circumferential component

Symbol	Description
$\bar{C}$	Direction cosine, axial component
S	Absolute flow direction
r	Radial direction
θ	Circumferential direction
z	Axial direction
m	Meridional direction
l	Direction defined by direction cosines, $\bar{A}$, $\bar{B}$, $\bar{C}$,
ρ	Density
Σ	Sector circumferential width

Section I

INTRODUCTION

This report is devoted to a detailed description of a computer program designed to enable the performance of axial compressors operating under conditions of combined radial and circumferential distortion to be predicted. The underlying theory is presented in Volume VII, where all the equations and expressions used are derived. Comparison between computed results obtained from the program and experimental data are made in Volume IX.

The report is divided into a number of sections, including a Functional Description of the program, an Operational Description of the program, and Conclusions and Recommendations. The Functional Description of the program is intended to enable the program to be used for its design purpose. The structure of the program is described, and the numerical techniques employed are delineated. Full details of the necessary input data are given. The various options available to the user are described, and recommendations regarding their use are made. The output from the program is described, with diagnostic messages fully detailed with respect to their form, origin, cause, and normal method of elimination, if any. (Sample inputs and results are held over for inclusion in Volume IX.) The Operational Description of the program is intended to convey a detailed knowledge of the methods employed in the program, to the extent of specifying groups of cards that perform specific steps in the computing procedure. This, in conjunction with the subroutine flow charts and other information given, should enable any desired modifications to the program to be assessed for feasibility, and subsequently implemented if appropriate.

The program has been written in Fortran IV for the IBM 7090/7094 IBSYS Operating System, the only additions required being control cards and twelve short MAP routines to prepare twelve intermediate storage units. Full details of all these are given.

Section II

FUNCTIONAL DESCRIPTION OF PROGRAM

PROGRAM STRUCTURE AND LOGIC

This section describes the implementation of the program on an IBM 7094 computer. This discussion is longer than would be the case for most programs due to the number of units required by the program, and also the need for an overlay scheme to enable the computer to accommodate the program. Should it be desired at any time to implement the program on a different computing system, this section should enable the feasibility of the change to be assessed, and the necessary changes to be made. The overall structure of the program is also described in terms of the logical flow of control through each routine. The purpose of each routine is indicated.

As previously described in the companion volume entitled "Basic Theory and Overall Description" in the sections entitled "The Overall Structure of the Computing System" and "Overall Computational Sequence", the solution of the problem revolves around repeated reestimates of the locations and extents of the streamtubes (sectors) into which the compressor is arbitrarily divided at the inlet. Evaluation of conditions in each sector, and subsequent reestimation of the sector locations and extents is termed a cycle. Within each cycle, the conditions within each sector are determined iteratively, and one iteration for conditions in a sector is termed a pass.

In order to handle the large arrays of numbers generated during any one cycle, some of which must be retained for the following cycle, extensive use of "out-of-core" storage is made. Some choices regarding the selection of either magnetic tape or disc (on the IBM 7094 installation at the USAF APL computing facility) are available, and the following considerations control the choices.

Because the computation of the solution to any problem is lengthy, provision is made in the program to stop the calculation at the termination of any cycle, and, presumably after examining the intermediate results obtained, restart the calculation. The data that is stored out-of-core falls into three classes. There is data that is written by the first part of the program and read, on the following cycle, by the second part of the program. This obviously cannot be scratched at the end of a cycle. Finally, there is data that is written by the second part of the program for use by the second part of the program on the following. Two sets of data are involved in this case; on any one cycle one set is being written and the other is being read. The units are then switched for the following cycle. Therefore, if the calculation is to be stopped and restarted after any cycle, both these units must be saveable, but if the calculation is to be stopped only after an even number of cycles, the same unit may always be scratched at the termination of the run. A further consideration is that if a restart is to be made after an odd number of cycles have been previously completed, these units must be switched externally in order to maintain the correct unit relationships.

All units are given integer variable names in the program and are defined numerically by arithmetic replacement statements in the main program. The unit names and uses are shown in Table I, together with some additional information.

For any particular application of the program not all of the "J-Series" and "L-Series" units may be necessary. The following determines the requirements of the program. The use of the "L" and "J-Series" units is similar, and the same rules apply to both. Therefore, only the final (numerical) character is used here to indicate the unit name.

Units 1 and 5 are always used. Unit 1 stores data for the computing stations from the inlet to first blade row inlet. Unit 5 stores data for the computing stations from the outlet of the last blade row to the outlet. Units 2, 3, and 4 are used (in that order of preference) as required, to store data for the stations within the compressor. Data for not more than six stations is contained on each unit, limiting the inlet and outlet regions to six computing stations each, and the computing stations within the compressor to eighteen.

An option in the program to compute the undistorted axisymmetric performance of the compressor leads, when taken, to reduced unit requirements. When the number of sectors is set equal to one, the calculation only proceeds so far as to determine the performance of one sector of the compressor, under the first-cycle, axisymmetric assumption. The program then terminates. The "L-Series" and the "K-Series" units are not used, and no provision for their existence is required.

The overlay structure required to accommodate the program on an IBM 7094 computer controlled by the IBM 7094 IBSYS operating system is shown in Figure 1. Note that this requires the use of the alternate FORTRAN IV Input/Output Package, obtained by specifying the option ALTIO on the $IBJOB control card. Shown on the figure are both the subroutine names and a two-digit number. This number with the prefix "S" is used to identify the source decks of the program, and may be used with, say, the prefix "B" to identify the relocatable binary decks produced by compilation.

Figure 2 is a flow diagram showing the logic of the program in the broadest terms. The main program has as its purpose the reading of a small quantity of data (including an indicator specifying whether the job is a new case or a restart, and also the specification of the number of cycles to be performed), and the subsequent execution of the various components of the program in accord with the specifications. If the job is a new case, Subroutine INPUT is called to read in the bulk of the input data, determine which units will be required to fulfill the data storage requirements, and write the initial data sets onto the "J-Series" units for "Program DISTORT". Three calls are then made in turn, and the sequence is repeated once for each cycle specified. Firstly, Subroutine DISTOR is called. Subroutine DISTOR may be considered to be the main program of "Program DISTORT", which makes the sector analyses. This call is made once for each of the number of sectors that are being used to describe the compressor. (Should the number of sectors have been set equal to one, this point is the termination of the job.) The second call is to Subroutine THETA. Subroutine THETA may be considered to be the main program of "Program THETA" which reestimates the sector locations and extents, and the various circumferential derivatives required by "Program DISTORT". The third call is to Subroutine PCTURE. This section of the program does not perform any calculations or other vital task, but, on request, produces crude plots of the conditions throughout the compressor on the regular line-printer.

Figure 3 is a flow chart for "Program DISTORT". All subroutines called by Subroutine DISTOR are indicated on this flow chart. Subroutine DISTORT calls Subroutine READIN, the main purpose of which is to read in the data for the analysis of the sector under consideration from the "J-Series" of units. Subroutine DISTOR then calls Subroutine SOLVE, which obtains the solution to the momentum and continuity equations. Subroutine SOLVE itself makes a number of calls. Unless a simple radial equilibrium solution is specified, Subroutine LSQLNE is called, which in turn calls Subroutine LSQFIT. This pair of subroutines determines the slope and rate of change of slope of the streamlines according to the latest estimate of their locations. For each computing station which is at a blade row outlet, Subroutine SOLVE calls Subroutine INTERP, which with the subroutines comprising the cascade performance prediction scheme, determines the performance of the blade row preceding the computing station. This call is made once for each streamline. Subroutine INTERP calls Subroutine GRAPH3 to interpolate the data describing the blade, and Subroutine CASCDE to perform most of the cascade analysis. Subroutine CASCDE calls Subroutine OPTANG to determine the minimum loss inlet flow angle to the blade section. Subroutine STLANG is called to determine the stalling and choking inlet flow angles. Subroutine STLANG calls Subroutine OUTANG to determine the outlet flow angle corresponding to the minimum loss inlet angle. Subroutines STLANG and OUTANG are again called by Subroutine CASCDE in order to obtain the actual outlet flow angle. Subroutine CASCDE calls Subroutine MCDEVN to compute the increase in deviation due to a supercritical inlet Mach number, if required. After computing a solution to the momentum and continuity equations, Subroutine SOLVE calls Subroutine BLTHIC to recalculate the annulus wall boundary layer displacement thicknesses. Note that the solution of the momentum and continuity equations and the recalculation of the boundary layers by Subroutine BLTHIC is an iterative calculation carried out for each computing station. Upon every third pass only, Subroutine DISTOR calls Subroutine SEARCH to reestimate the flow in the sector which corresponds to the specified exit static pressure. In order for the solution to be considered converged, the velocities determined by Subroutine SOLVE on successive passes must be within one half a per cent of each other, and the exit static pressure must be within 0.2 per cent of the specified value. These criterions are checked by Subroutine DISTOR (in conjunction with Subroutine SOLVE), and if they are met, Subroutine PRINT is normally called to print the output describing conditions in the sector, and Subroutine PUNCH is called to write the data for "Program THETA" onto the "L-Series" of units. If the solution is not converged, and the specified maximum number of cases has not been performed, Subroutine DISTOR calls Subroutine SOLVE, thus commencing another pass.

Figure 4 is a flow chart for "Program THETA". All subroutines called by Subroutine THETA are shown. Function FINDY linearly interpolates input data at circumferentially averaged streamline radii. The principal function of THETA is to determine the location of sector centerlines. This is necessarily an iterative process involving the circumferential component of the momentum equation. In the extended regions upstream and downstream of the compressor this iterative solution can become unstable and must initially be constrained by the results of a first-order perturbation analysis. This function is served by Subprograms FOUR, NEWRAD, XMEAN, SMITH, GEORGE, EDWIN. Other service operations of THETA are to determine angular sector widths; circumferential gradients of streamline slope and pressure, changes in angular momentum associated with the latter; and the local slope of the sector centerline against the meridional plane. The associated service Functions are SERVUS, DERIV, DELTA.

The processing of large blocks of data by Subroutine THETA is performed for a maximum of 6 axial stations per L-unit and the results must be transmitted to the respective J-unit and auxiliary records (K-unit) before the next block may be read in.

Figure 5 is a flow chart showing Subroutine PCTURE and the subroutines called by it. Subroutine PLOT, which is always called, prepares a polar plot for each computing station showing the position of the sector centerlines. The angular locations are obtained from the "K-Series" unit previously written by Subroutine THETA. These are plotted with the "reference radii", computed by Subroutine THETA, and included in the data written onto the "J-Series" units. The use of the reference radii (the mean of the radii for all sectors) is an approximation, as the streamlines are, in general, at different radii in each sector. However, this error is not significant when the plot is made on a standard line-printer. Subroutine MPLOT is called (if requested by options in the input data), and prepares five plots for each computing station. These are prepared on Cartesian coordinates and show the variation of certain quantities with radius, the radii used for all sectors being the radii of the streamlines as computed for the last of sectors comprising the set used to describe the compressor. The error involved will not normally be discernible on the resulting line-printer produced plot. Results for all sectors at any one station are shown on one graph, and the quantities plotted are meridional velocity, static pressure, total pressure, total temperature, and whirl angle. All the data is obtained from the "L-Series" units written by Program DISTORT.

This completes the description of the logic of the program.

INPUT DATA

A precise definition of the input data required to run the program is given. All options available are mentioned, and discussed below. The following notes appertain to the detailed description. The line number is equated to card number, the location numbers being the columns of the card. Three formats are used for input. "A" indicates alphanumeric characters. "R" indicates real numbers, which are all placed in fields of eight locations. A decimal point should be included to ensure the correct interpretation of the data; the number may be placed anywhere within the specified eight locations. Up to nine numbers, using Locations 1 through 72, are placed on one card. "I" indicates integers, which are all placed in fields of six locations. No decimal point may be used, and the number must be placed in the rightmost locations of the allocated field. Up to twelve numbers, using Locations 1 through 72, are placed on one card.

Line	Location	Type	Input Item	Comments
1	1-72	A	TITLE	A title for the job
2	1-6	I	NTHETA	Number of sectors, 3≤NTHETA ≤13, except NTHETA=1 for an axisymmetric analysis
	7-12	I	NLOOPS	Number of cycles to be performed. Insignificant if NTHETA=1
	13-18	I	IFSTAR	0: New case 1: Restart Must be 0 if NTHETA=1
	19-24	I	IFPRIN	Controls output for cycles after the first
3	1-8	R	RFAC1	Relaxation factor used with circumferential static pressure gradient
	9-16	R	RFAC2	Relaxation factor used in reestimation of angular location of streamlines
	17-24	R	RFAC3	Increment by which perturbation solution is eliminated per cycle after cycle IFAC1

Line	Location	Type	Input Item	Comments
	25-32	R	RFAC4	Not used
4	1-6	I	IFAC1	First cycle (of this run) for which the perturbation solution is partially eliminated
	7-12	I	IFAC2	Number of iterations between perturbation solution and boundary conditions
5	1-8	R	R32(1)	Angular location of first sector centerline in inlet (degrees)
	9-16	R	R32(2)	Angular location of second sector centerline in inlet (degrees)

This list is continued to give NTHETA values in all, the final value being on card (A-1).

If IFSTAR=1, there is no further data.

Line	Location	Type	Input Item	Comments
A	1-6	I	NX	Total number of computing stations $6 \le NX \le 30$
	7-12	I	NLINES	Number of streamlines $3 \le NLINES \le 15$
	13-18	I	NBETA	Number of blading data radii $1 \le NBETA \le 15$
	19-24		NTOPO	Number of radii at which inlet conditions are specified $1 \le NTOPO \le 15$
	25-30	I	IFSIMP	≠2: Full radial equilibrium 2: Simple radial equilibrium; i.e., streamlines have no slope or curvature
	31-36	I	NPASS	Maximum number of passes permitted during any sector analysis
	37-42	I	IFBL	1: No annulus wall b.l. calculation

Line	Location	Type	Input Item	Comments
				2: Annulus wall b. l. calculated
	43-48	I	IFRPM	Controls output for first cycle; see below
	49-54	I	ITER	1: All velocities calculated during iterations printed 2: Normal option
	55-60	I	NPLOT	First pass during which results of cascade analysis are printed
	61-66	I	INCPO	Increment for above
	67-72	I	NWRIT	First pass during which velocity triangle data is printed
A+1	1-6	I	INCWRI	Increment for above
	7-12	I	IFTYPE	0: All stations radial, all solutions subsonic 1: Station lean angles and solution type to be specified
A+2	1-8	R	TOLCX	Tolerance for iterative solution of continuity equation at each station
	9-16	R	HUBLOC	Fraction of blockage area on hub
	17-24	R	GASK	Gas constant (ft lbs per lb deg R)
	25-32	R	CP	Specific heat (Btu per lb deg R)
	33-40	R	VIS1	Viscosity for the blade Reynolds number is given by =VIS1+t*VIS2 (lb per ft sec)
	41-48	R	VIS2	See above (lb per ft sec deg R)

Line	Location	Type	Input Item	Comments
	Lines A+3 through A+4 are entered for computing stations 2 through NX.			
A+3	1-6	I	IBETA2	Primary indicator for method of determining flow angle at the computing station; see below
	7-12	I	IFTHIC	1: Relative flow angle corrected for thickness-chord ratio 2: Correction not made (not required unless IBETA2 is 2 or 4)
	13-18	I	IFCAX	1: Relative flow angle corrected for axial velocity ratio 2: Correction not made (not required unless IBETA2 is 2 or 4)
	19-24	I	IFMACH	1: Relative flow angle corrected for supercritical Mach numbers 2: Correction not made 3: Correction multiplied by XMACH (not required unless IBETA2 is 2 or 4)
	25-30	I	IFREYN	1: Relative flow angle corrected for subcritical Reynolds number operation 2: Correction not made (not required unless IBETA2 is 1, 2, 4, or 5)
	31-36	I	ILOSS	1: Loss coefficient specified 2: Loss coefficient calculated 3: Correction to specified minimum loss coefficient calculated (not required unless IBETA2 is 1, 2, 4, or 5)

Line	Location	Type	Input Item	Comments
	37-42	I	IFMLOS	1: Loss coefficient corrected for Mach number effects 2: Correction not made (not required unless ILOSS is 2 or 3)
	43-48	I	IFLVSI	1: The exponent for the rate of change of loss with incidence will be read in 2: Standard values of the exponent will be used (not required unless ILOSS is 2 or 3)
	49-54	I	IFPROF	1: A loss incidence relationship applicable to subsonic (e.g., NACA 65-Series) blades is used 2: A loss/incidence relationship applicable to transonic (e.g., d.c.a.) blades is used (not required unless ILOSS is 2 or 3)
	55-60	I	IFREYL	1: Loss coefficient corrected for subcritical Reynolds number operation 2: Correction not made (not required if IBETA2 is 6)

If IFLVSI=1, the following line is required

	1-8	R	EXPLOS	Exponent in loss/incidence relationship

If IMACH=3, the following line is required

	1-8	R	XMACH	Multiplying factor for supercritical Mach number deviation correction

If IBETA2 is not equal to 6, the following line is required

	1-8	R	RPM2	The speed of rotation of

Line	Location	Type	Input Item	Comments
				the blade row preceding the computing station (rpm)

The data on Line A+4 take on various meanings depending upon the previously entered values of IBETA2 and ILOSS. The line is repeated NBETA times. When IBETA2 is 1, the following data is entered:

A+4	1-8	R	BBP	Blade data radius (ft)
	9-16	R	STAG	Relative outlet flow angle, measured from axis in direction of rotation (degrees)
	49-56	R	CLOSS	Blade row loss coefficient

When IBETA2 is 2, the following data is entered.

A+4	1-8	R	BBP	Blade data radius (ft)
	9-16	R	STAG	Blade stagger angle, measured from the axis and normally positive (degrees)
	17-24	R	BETA	Blade camber angle, normally positive (degrees)
	25-32	R	SIGMA	Cascade solidity
	33-40	R	BLAK	Blade thickness/chord ratio
	41-48	R	ZBAR	Distance from leading edge to point of maximum camber of blade, as a fraction of chord
	49-56	R	CLOSS	When ILOSS=1, this is the loss coefficient. When ILOSS=3, this is the minimum loss coefficient. Not required when ILOSS=2
	57-64	R	CO	Not required when ILOSS=1. When ILOSS=2 or 3, is additional loss factor

When IBETA2 is 4, STAG and BETA only take different meanings from when IBETA2 is 2, as shown below.

Line	Location	Type	Input Item	Comments
	9-16	R	STAG	Blade inlet angle measured from the axis and normally positive (degrees)
	17-24	R	BETA	Blade outlet angle measured from the axis, and positive unless the blade turns past the axial direction (degrees)

When IBETA2 is 5, the data required is as for IBETA2 is 2, except that BLAK and ZBAR are not required.

When IBETA2 is 6, no data is required.

As mentioned above, line (A+4) is repeated NBETA times, and lines (A+3) and (A+4) are repeated (NX-1) times. The final line is (B-1).

Line	Location	Type	Input Item	Comments
B	1-8	R	DELM(1)	Must be 0.0
	9-16	R	DELM(2)	Fraction of annulus height at inlet between hub and first streamline in flow
	17-24	R	DELM(3)	Fraction of annulus height at inlet between hub and second streamline in flow

This list is continued to include NLINES values, DELM (NLINES) necessarily being 1.0. The final value is on line (C-1).

Line	Location	Type	Input Item	Comments
C	1-8	R	BLAM(1)	Fraction of annulus area at first computing station blocked by boundary layer displacement thicknesses
	9-16	R	BLAM(2)	Fraction of annulus area at second computing station blocked by boundary layer displacement thicknesses

This list is continued to include NX values, the final value being on line (D-1).

Line	Location	Type	Input Item	Comments
D	1-8	R	FLOWI	Specified exit static pressure (lbs per sq ft)
D+1	1-8	R	ESTFLO	Estimate of flow to satisfy

Line	Location	Type	Input Item	Comments
				static pressure specified above (lbs per sec)
D+2	1-8	R	WIDTH	Estimated flow range of compressor characteristic as a fraction of flow
D+3	1-8	R	X(1)	Axial coordinate of intersection of first computing station with axis (ft)
	9-16	R	RHP(1)	Radius of hub at first computing station (ft)
	17-24	R	RSP(1)	Radius of casing at first computing station (ft)

Line D+3 is repeated for each of the NX computing stations, the final line being line (E-1).

If IFTYPE is 0, the next line of data required is line G.

Line	Location	Type	Input Item	Comments
E	1-6	I	IMACH1	0: Subsonic solution to continuity equation at first station is sought 1: Supersonic solution to continuity equation at first station is sought
	7-12	I	IMACH1	0 or 1, as above, for second station

This list is continued for all NX stations, the final value being on line (F-1).

Line	Location	Type	Input Item	Comments
F	1-8	R	ANGLN	Angle between first computing station and radial direction, positive when increasing radius accompanies increasing axial coordinate (degrees)
	9-16	R	ANGLN	Angle as defined above for second computing station

This list is continued for all NX stations, the final value being on line (G-1).

Lines G and G+1 are repeated for each of the NTHETA sectors into which the compressor is divided.

Line	Location	Type	Input Item	Comments
G	1-72	A	COMENT	A title for the sector
	Line G+1 is repeated NTOPO times.			
G+1	1-8	R	BHP	Data radius for inlet conditions (ft)
	9-16	R	TOCO	Inlet total temperature (degrees R)
	17-24	R	POCO	Inlet total pressure (lbs per sq ft)
	25-32	R	AFCO	Inlet whirl angle measured from the axis, position in direction of rotation (degrees)

This completes the input data requirements of the programs, and some comments regarding the available options and their selection now follow.

NLOOPS - It is recommended that an even number of cycles be specified in order to simplify the unit requirements. This is discussed in the section entitled "Program Structure and Logic".

IFPRIN - This indicator may take one of eight values and controls the amount of output data produced on each cycle after the first, and also, for the first cycle, the output produced by Subroutines MPLOT and PCTURE. The particular data items controlled by IFPRIN are some of the input data to Program DISTORT, the results printed by Program DISTORT and the Cartesian plots produced in Subroutine MPLOT.

Shown below are the eight values IFPRIN may take, and the output items produced accordingly.

IFPRIN	OUTPUT
1	DISTORT input
2	DISTORT Input DISTORT Results
3	DISTORT Input DISTORT Results MPLOT Results
4	DISTORT Input MPLOT Results
5	DISTORT Results

IFPRIN	OUTPUT
6	DISTORT Results MPLOT Results
7	MPLOT Results
8	No Results Printed

See the discussion of IFRPM below for recommendations.

RFAC1 - On the nth cycle, the circumferential static pressure gradient is calculated in Subroutine THETA as RFAC1*Gn-1+(1.0-RFAC1)*Gn, where Gn is the value implied by the static pressures on cycle n, and Gn-1 is the value determined in a similar manner on cycle (n-1). It currently appears that RFAC1 should be set to 0.5.

RFAC2 - The angular location of the streamlines is determined using the same relaxation method as described above for static pressure gradient, RFAC2 being the relaxation factor employed. It currently appears that RFAC2 should be 0.5 for machines employing final stator blades, and 0.7 for isolated rotor cases.

RFAC3 - This controls the rate of release from the perturbation solution, as outlined below in the discussion of IFAC1. It currently appears that RFAC3 should set to 0.2, thus releasing the calculation completely from the perturbation solution constraint after five cycles.

IFAC1 - As described in the companion volume entitled "Base Theory and Overall Description" in the sections entitled "Overall Structure of Computing System" and "Overall Computational Sequence" a small perturbation solution is employed in the inlet and outlet ducts during the initial cycles of calculation. Commencing on cycle number IFAC1, the perturbation solution is discarded by using a progressively increasing fraction of the quantities produced by the basic program and a progressively decreasing fraction of the results of the perturbation solution. (The quantities involved are the static pressure in the inlet duct and the flow angle in the exit duct.) Note that IFAC1 refers to the cycle number of the run, which will not be the absolute cycle number for a restart run. It currently appears that IFAC1 should be in the range of 4 to 8.

IFAC2 - Three cycles of perturbation solution plus recalculation of the boundary conditions used to obtain it are normally appropriate.

NLINES - The calculation time will be almost proportional to the number of streamlines. Eleven has produced satisfactory results in the past.

IFSIMP - A simple radial equilibrium solution in which radial components of the flow are neglected, should be obtained more quickly than a full radial equilibrium solution. In the latter case, the streamline slopes and rate of change of slopes are found from a second order polynomial least-squares fit curve through a specified number of points. The number of points must be odd, and IFSIMP is set equal to 2 less than the desired number of points. The number of points must lie in a range of 3 to 11, that is, IFSIMP lies in the range 7 through 9 (and is odd). Generally, 3 should be used, but sometimes when the distance between the computing stations becomes small, stability of the calculation is only maintained by increasing the number of points to 7. This assumes that the option to recalculate the annulus wall boundary layers is taken; stability is greater -- the outer boundaries of the flow are fixed.

NPASS - The solution for any sector normally converges after from about twelve passes for a single stage to twenty passes for a multistage compressor. To give a reasonable assurance of attaining convergence, NPASS should be set to from about 20 to 50, depending upon the number of stages of the compressor.

IFBL - When the option to calculate the annulus wall boundary layer displacement thickness is taken, the values of BLAM and HUBLOC specified in the input data apply for the first cycle only. After that, except at the first computing station, the displacement thickresses on the hub and casing are recalculated during each iteration.

IFRPM - This indicator operates as IFPRIN to control the output from Program DISTORT only, on the first cycle only. Generally, input data to Program DISTORT is not required, the output from Program DISTORT should be obtained, and the Cartesian plots are not generally required before the final cycle, if then.

ITER - The option to print all velocities produced during the iterations to satisfy continuity and momentum is time and paper consuming. Should convergence difficulties arise in the sector analyses, it should preferably be used only on a single-sector, axisymmetric analysis.

NPLOT - The detailed results of the cascade analysis are not essential to understanding the operation of the compressor, but will provide information to assist decisions regarding possible modification of the blading to change the performance of the machine. As convergence generally occurs after between twelve to twenty passes, depending upon the number of stages of the compressor, NPLOT should set to, say, 10 to 18, if this output is desired. Otherwise, it should be set to a number greater than NPASS.

INCPO - Cascade analysis results are printed every INCPO passes after NPLOT. A small change in flow conditions will not normally completely invalidate the results previously produced, so NPLOT should be set to, say, 9.

NWRIT - Velocity triangle data is automatically produced upon convergence, so NWRIT may be set to a number greater than NPASS.

IFTYPE - Supersonic solutions are not generally of interest in compressor analysis, except that occasional interest has been shown in supersonic rotors. Supersonic turbine stators have been used relatively frequently in turbines, however. The calculation procedure becomes ill-conditioned if lean angles greater than about 30 degrees are specified.

TOLCX - It would be logical to include all iteration tolerances in the input, or to relate them to TOLCX and other parameters. This is not done, and a value of 0.001 for TOLCX is normally satisfactory, and completely compatible with the other, built-in tolerances.

HUBLOC - In order to specify an equal displacement thickness on the hub and casing at the inlet, a value of one half the inlet hub/casing radius ratio may be used as a good approximation.

IBETA2 - This indicator takes one of five possible values. When IBETA2 is 1, ILOSS must also be 1. Then the blade row performance, that is relative outlet flow angle and loss coefficient, is fixed by the input data. When IBETA2 is 2, the full range of cascade analysis options for compressor blade performance prediction is available. This is the most frequently used option. When IBETA2 is 4, there is only a small change in the input data, relative to when IBETA2 is 2. When IBETA2 is 5, ILOSS must be 1. A simple prediction of the relative outlet flow angle applicable to inlet guide vanes is employed, which assumes that the point of maximum camber is at mid-chord. When IBETA2 is 6, conditions at the computing stations are determined by taking the total temperature and total pressure unchanged along each streamline from the previous computing station, and the change in angular momentum to be as dictated by the (nonaxisymmetric) circumferential component of the equation of motion.

IFMACH - When used, the option to compute the additional deviation due to supercritical Mach number operation of a blade row can lead to significant changes in the predicted performance. The empirical data used in the method was obtained from tests on NACA 65 Series blades, and thus the correction may require modification for blades which perform differently at elevated Mach numbers. The multiplying factor XMACH may be used to reduce (or increase) the impact of the correction.

IFLVSI - Loss coefficient is determined from the relation CL=WOPT $(1.0+S^{EXPLOS})$ where WOPT is the optimum loss coefficient, S is the incidence normalized with respect to stalling or choking incidence, as appropriate, and EXPLOS is an exponent which may be varied by the user. If IFLVSI is 2, EXPLOS will be automatically set to 4.5 when IFPROF is 1, and 2.0 when IFPROF is 2.

BBP, BHP - All data radii sets must monotonically increase.

NUMERICAL TECHNIQUES

The data supplied as input to the program is linearly interpolated from the two nearest points in the table to obtain values at the required radii. Where extrapolation is required, this is done linearly from the appropriate two last points in the table.

Where differential coefficients are required, these are obtained from the following relation:

$$\left(\frac{dx}{dy}\right)_n = \frac{1}{2}\left(\frac{x_{n+1} - x_n}{y_{n+1} - y_n} + \frac{x_n - x_{n+1}}{y_n - y_{n+1}}\right)$$

These may be applied for all "y" when used to obtain circumferential derivatives, but special cases occur when it is used to obtain derivatives in the direction of the sector centerlines. Then, the derivatives at the boundaries of the finite difference mesh are obtained from:

$$\left(\frac{dx}{dy}\right)_n = \frac{x_n - x_{n-1}}{y_n - y_{n-1}}$$

Integration of the momentum equation in the direction of the sector centerlines is achieved by assuming that certain terms do not vary in the interval between any two streamlines (the radii of which define the finite difference mesh). The equation may be written in the form:

$$\frac{dc_m^2}{dl} = A c_m^2 + B$$

where

$$A = 2\cos^2\alpha\left\{(\bar{A}\cos\phi - \bar{C}\sin\phi)\left(\frac{1}{r_m} + \frac{\tan\alpha}{r}\frac{d\phi}{d\theta}\right) - \tan\alpha\frac{d\tan\alpha}{dl} - \bar{A}\frac{\tan^2\alpha}{r}\right\}$$

$$B = 2\cos^2\alpha\left\{\frac{dH}{dl} - t\frac{dS}{dl} - \frac{\bar{B}}{\rho r}\frac{dP}{d\theta}\right\}$$

and

$$\bar{A} = (1 + \tan^2\gamma + \tan^2\mu)^{-\frac{1}{2}}$$

$$\bar{B} = \bar{A}\tan\mu$$

$$\bar{C} = \bar{A}\tan\gamma$$

Then by assuming that A and B do not vary in the interval $\Delta r = r_2 - r_1$, one may obtain

$$c_m^2 = c_{m_1}^2 e^{A\cdot\Delta r} - \frac{B}{A}\left(1 - e^{A\Delta r}\right)$$

This expression is used in the computer program, the values of A and B being taken as the mean of the values applying to radii r_1 and r_2. The values of r_1 and r_2 of course are taken as the radii of each adjacent pair of streamlines in turn. This integration formula is used because an exact integration of the equation leads to an instability in the iterative solution of the momentum and continuity equations when the term B is large. It should be noted that the approximation made (that A and B do not vary in the interval Δr) vanishes as B vanishes.

A special case occurs if A vanishes. Then the following formula is used:

$$C_m^2 = C_{m_1}^2 + B\Delta r$$

Integration of the continuity equation in the direction of the sector centerlines to give the weight flow in any one sector is performed using a trapezoidal integration, the mean axial component of velocity being multiplied by the mean fluid specific weight in any interval. These sector flows are then added to give the total flow in the compressor.

An integration is required for the satisfaction of the circumferential component of the equation of motion. Again, a trapezoidal rule is used, the mean value of $1/\rho \cdot \partial P/\partial\theta \cdot 1/C_m$ at the end points of the integration interval Δz is being multiplied by Δz to give the required result.

The method of calculating the angle the sector centerlines make with the radial direction is illustrated in Figure 6. At the hub, the angle is taken the angle shown as μ_1, and at the casing the angle μ_4 is used. At all intermediate points, the angle is taken as the mean of the angles such as μ_2 and μ_3.

The angular width of each sector, at each reference radius, is taken as one half the difference between the angular locations of the adjacent sector centerlines. This means that the sector "centerlines" are not, in general, at the center of the area which they describe.

OUTPUT DETAILS

In this section the printed output produced, or potentially produced, by the program is described. Twelve subroutines contribute to the program output, and the various items are here listed by subroutine. Some of the output may be considered "diagnostic" although in a program of this nature the division between "normal" and "diagnostic" output is not well defined. In general, in the following descriptions, the normal output is outlined and the diagnostic output is described in detail with the actual test of the message given.

PRINTED OUTPUT FROM MAIN PROGRAM

The main program produces a printout of the input data on lines 1 through 5 as described in the section entitled "Input Data".

PRINTED OUTPUT FROM SUBROUTINE INPUT

Subroutine INPUT produces a printout of the input data on lines A through (R_1 + 1), as described in the section entitled "Input Data".

If IBETA2 for any computing station is not 1, 2, 4, 5, or 6, or IBETA2 is 1, 2, 4, or 5, and ILOSS is not 1, 2, or 3, the following message is printed.

JOB STOPPED BECAUSE INDICATOR OUT OF RANGE - SUBROUTINE INPUT

Subroutine INPUT also produces a table showing the fourteen tape units defined in the main program, the logical unit numbers that have been given to them, and the uses to which they are put for the particular job.

PRINTED OUTPUT FROM SUBROUTINE DISTOR

Subroutine DISTOR produces a printout showing the results of the annulus-wall boundary-layer calculations. The fraction of the annulus area that is blocked by the boundary-layer displacement thicknesses at each station is shown, for a final, converged pass, if the boundary-layer calculation is being performed. Note that this output is also under the control of the indicators IFPRIN and IFRPM, as described in the section entitled "Input Data".

PRINTED OUTPUT FROM SUBROUTINE READIN

Again under the control of IFPRIN and IFRPM, Subroutine READIN will produce a printout of the sector heading card (COMENT), the sector inlet conditions, the blading details, and the current estimate of the parameters describing the asymmetric flow in the sector. For cycles after the first (in absolute terms), a listing of the estimated streamline radii is also produced.

The following diagnostic messages may be produced.

INVALID VALUE OF IBETA2 ENCOUNTERED.

(Theoretically, this is masked by the check in Subroutine INPUT.)

**** ILOSS GREATER THAN UNITY WHEN IBETA2=1. CALCULATION FOR CASE TERMINATED *****

DATA ERROR - NX GREATER THAN 30, OR NLINES OR NBETA GREATER THAN 15

PRINTED OUTPUT FROM SUBROUTINE SOLVE

Again, under the control of IFPRIN and IFRPM, and if specified by the indicators NPLOT and INCPO (as described in the section entitled "Input Data"), Subroutine SOLVE prints a heading for the cascade analysis performed by Subroutines INTERP, CASCDE, et al.

The following messages may be printed.

VELOCITY VALUE LIMITED AT STATIONxxx STREAMLINExxx PASSxxx CONTINUITY LOOPxxx

This message will occur if a meridional velocity of less than one foot per second is calculated to exist in the stepwise integration process of the momentum equation, during pass number 3, 6, 9, ..., and if this message and the one referring to choking of the machine has not previously occurred on the current pass. The program assumes the flow i the sector is surged when this message occurs.

VELOCITY CHANGE BY MOMENTUM LIMITED AT STATIONxxx STREAMLINExxx PASSxxx CONTINUITY LOOPxxx

This message is printed if, during the fourteenth or fifteenth iteration at a computing station, the meridional velocity determined is outside the range of 0.6 to 1.4 times the value determined on the previous iteration. Difficulty in obtaining a converged solution to the momentum equation is indicated when this message occurs.

STATIC TEMPERATURE RESTRAINED AT STATIONxxx STREAMLINExxx PASSxxx CONTINUITY LOOPxxx

This message is printed if, during the fourteenth or fifteenth iteration at any computing station, a static temperature of less than 0.001 degrees Rankine is calculated. Difficulty in satisfying the continuity equation is indicated, the message rarely occurs.

MACH NUMBER SWITCH PERFORMED AT STATIONxxx PASSxxx CONTINUITY LOOPxxx

This message is printed if at a computing station the mass-flow weighted mean-controlling Mach number is supersonic, and a subsonic solution to the continuity equation is requested, or vice versa. Also, the message described above regarding surge and the one describing choking must not have occurred during the pass. Difficulty in satisfying the continuity equation is indicated if the message occurs repeatedly.

VELOCITY CHANGE BY FLOW LIMITED AT STATIONxxx PASSxxx CONTINUITY LOOPxxx

This message is printed if, during the fourteenth or fifteenth iteration at a computing station, the factor by which the velocities must be multiplied to satisfy continuity is outside the range of 0.8 to 1.2.

Difficulty in satisfying the continuity equation is indicated; the message rarely occurs.

STATIONxxx STREAMLINExxx PASSxxx CONTINUITY LOOPxxx VELOCITYxxxxxx.xx

This is the output generated when the indicator ITER is set to 1 (in conjunction with the output from Subroutine BLTHIC).

CASE 1 ASSUMED CHOKED DURING PASSxxx AT STATIONxxx RATIO OF CHOKING FLOW TO SPECIFIED FLOW = xx.xxxx

This message is printed during passes 3, 6, 9, ..., if the solution is unconverged after fifteen iterations and the program determines that the specified flow cannot be passed. This message, and the one described above regarding surged flow, must have not previously occurred during the pass.

CONTINUITY AND MOMENTUM NOT SATISFIED AT STATIONxxx PASSxxx RATIO OF FLOWS = xx.xxxx RATIO OF VELOCITIES = xx.xxxx

This message is printed when the meridional velocity distribution fails to converge after fifteen iterations, the pass is number 3, 6, 9, ..., and the flow has not been determined to be either surged or choked at a previous station during the pass. The ratio of flows is the specified flow to the achieved flow. (Specified flow means the current estimate of the flow required to give the specified downstream static pressure.) The ratio of velocities is the ratio of the meridional velocities calculated at hub on the fifteenth iteration to that calculated on the fourteenth. This message rarely occurs unless the section is choked.

PRINTED OUTPUT FROM SUBROUTINE SEARCH

Subroutine SEARCH prints the number of the pass on which it was entered, the new estimate of the flow required to give the specified downstream static pressure, and the mass-flow weighted mean-exit static pressure obtained on the current pass. Note that this latter quantity will be meaningless if no valid solution was obtained.

PRINTED OUTPUT FROM SUBROUTINE CASCDE

As directed by the indicators IFRPM, IFPRIN, NPLOT, and INCPO, Subroutine CASCDE produces a printout showing the various steps taken to reach the final estimate of the cascade performance. The following notes appertain when IBETA2 is 2 or 4.

For each blade row there is given a table indicating the various options selected for the performance prediction for the blade row. Then details are given for each strealine. STAGGER, SOLID., CAMBER, THICN., and ZBAR give the cascade section interpolated for the streamline. IN.ANG.

is the relative air inlet-angle to the section. B1DES is the minimum-loss inlet-angle, as given by Subroutine OPTANG, and corrected for extreme solidity, for point of maximum camber, and for thickness/chord ratio. BCHOKE and BSTALL are the choking and stalling inlet-angles as given by Subroutine STLANG and corrected for point of maximum of camber, extreme solidity, and for thickness/chord ratio. B1DESE is the minimum-loss inlet-angle, as given by Subroutine OPTANG, and corrected for point of maximum camber and axial velocity-ratio. B2DESE is the minimum-loss outlet-angle determined by Subroutine OUTANG to correspond to B1DESE. B2DAX is the angle B2DESE corrected for axial velocity-ratio. B2DES is the angle B2DIN corrected for thickness/chord ratio. BE1 is the actual inlet-angle, corrected for axial velocity-ratio. BE2 is the outlet angle determined by Subroutine OUTANG, and corresponding to BE1. B2AXVL is the angle BE2 corrected for axial velocity-ratio. B2THIC is the angle B2AXVL, corrected for thickness/chord ratio. RM1 is the relative-inlet Mach number to the cascade section. VMAX is the ratio of the maximum to the inlet velocities when the section is operating at design condition. CRITM is the critical inlet-Mach number for the blade section. B2M is the angle B2THIC corrected for Mach number. REYN is the section Reynolds number. B2 is the angle B2M corrected for Reynolds number, and is the finally determined outlet angle. DEQDES is the design diffusion factor. WDES is the minimum loss-coefficient. WOPT is the loss coefficient WDES corrected for Mach number. B1SCOR and B1CCOR are the angles BSTALL and BCHOKE corrected for Mach number. CL1 is the loss coefficient WOPT corrected for incidence. CL is the loss coefficient CL1 corrected for Reynolds number, and is the finally determined loss coefficient.

When IBETA2 is 5, a simplified printout details the calculation applied to an inlet guide vane.

PRINTED OUTPUT FROM SUBROUTINE MCDEVN

The following messages may occur, if the particular situations arise on a pass specified by the indicators NPLOT and INCPO.

NON-CONVERGENCE ON CRITM2 AFTER 40 LOOPS

CRITM2 HAS EXCEEDED UNITY - IT IS TEMPORARILY SET EQUAL TO 1.0

These indicate failure of the iterative determination of the Mach number effect on deviation angle, and do not normally occur.

PRINTED OUTPUT FROM SUBROUTINE PRINT

Under the control of the indicators IFPRIN, IFRPM, NWRIT, and INCWRI, as described in the section entitled "Input Data", Subroutine PRINT produces the main sector analysis output. All quantities are headed in a self-explanatory format, with the possible exception of the following items. DE HALL NUMBER is the ratio of relative outlet to inlet velocities for the blade section. DELTA P ON Q is the static pressure rise divided by the difference between the relative inlet total and static pressures for the blade row.

PRINTED OUTPUT FROM SUBROUTINE BLTHIC

The output printed by Subroutine BLTHIC, when called for by the appropriate value of the indicator ITER, consists of blockage and wall boundary-layer displacement thicknesses at each axial station.

PRINTED OUTPUT FROM SUBROUTINE THETA

The printed output of Subroutine THETA provides intermediate information in the form of sector centerline locations and optional modifications imposed on static pressures and absolute air angles by a perturbation analysis in the upstream and downstream region.

A diagnostic message is printed when the angular width of a sector degenerates, i.e., when streamlines of neighboring sectors converge or diverge excessively. The message begins:

THE WIDTH RATIO OF SECTORxxx AT STREAMLINExxx, STATIONxxx IS EITHER GREATER THAN 2.0 OR LESS THAN 0.5

This message is followed by a printout of the circumferential distribution of the original and corrected tangent values of the absolute air angle at the particular radius and station. This latter correction is arbitrary and only intended to continue program operation.

PRINTED OUTPUT FROM SUBROUTINE OVERAL

Subroutine OVERAL produces a printout of the integrated performance of the compressor. The total-pressure ratio is determined by summing the work input required to produce the total-pressure rise in each streamtube isentropically, and then converting the resulting total work input back to an isentropic pressure ratio. In this case, streamtube means the subdivision of each sector by the streamlines, giving a possible maximum of 182 streamtubes.

PRINTED OUTPUT FROM SUBROUTINE PLOT

Subroutine PLOT produces a polar plot for each computing station, showing the location of each sector centerline.

PRINTED OUTPUT FROM SUBROUTINE MPLOT

Under the control of the indicator IFPRIN, Subroutine MPLOT produces plots showing the variation of meridional velocity, total and static pressure, total temperature, and flow angle in each sector at each computing station.

MISCELLANEOUS INFORMATION

Various items regarding the computer program not included elsewhere are given here.

CONTROL CARDS, MAP ROUTINES

The program has been written in FORTRAN IV for the IBM 7090/7094 IBSYS Operating System as implemented on the Directly Coupled System employed at the USAF APL Computing Facility. All non-Fortran cards required to run the program are described here.

With the exception of the standard input and output units, the unit numbers assigned to the units used by the program are arbitrary. The values shown in Table II are assumed to the following notes.

The following cards are required to execute the program. A $JOB card is required to initiate the job. $SETUP cards should then be used to specify magnetic tape reels for units A(1), A(7), and B(3), also, if required by the particular job, A(2), A(3), and A(4). A $ASSIGN card then prepares SYSUT3 to receive the overlays of the program, a considerable speed-up in program execution occurring when the overlays are stored on magnetic tape instead of disk. A $EXCUTE job then specifies the IBJOB system. On the following $IBJOB card the following options are recommended: ALTIO (mandatory), MAP, NOLOGIC, GO, NOSOURCE (if applicable). Then follow twelve MAP routines relating the Fortran and logical unit numbers given in the Table II, and also the modes and block sizes given in Table I. The remainder of the deck consists of the 42 routines of which the program is comprised, with the appropriate $ORIGIN cards placed amongst them to create the Overlay Structure shown on Figure 1. Note that these should specify the links to reside on SYSUT3, and REW options at strategic points assist the program execution. A $INCLUDE card places six COMMON blocks in the same link as Subroutine DISTOR. For initial compilation, each source routine is preceded by a $IBFTC card, and a $* card where there is no $ORIGIN card. Finally, a $DATA card separates the program from the input data deck. The following shows all the cards described above, in the form required for compilation of the program from source deck, loading, and execution.

```
$JOB
$SETUP A(1)     xxx
$SETUP A(2)     xxx
$SETUP A(7)     xxx
$SETUP B(3)     xxx
$ASSIGN         SYSUT3
$EXECUTE        IBJOB
```

```
$IBJOB            ALTIO,MAP,NOLOGIC,SOURCE,GO
$IBMAP  U01       4,DECK,M94,XR7
        ENTRY     .UN01.
.UN01.  PZE       UNIT01
UNIT01  FILE      ,A(1),INOUT,BLK=22,BCD
        END
$*
$IBMAP  U02       4,DECK,M94,XR7
        ENTRY     .UN02.
.UN02.  PZE       UNIT02
UNIT02  FILE      ,A(2),INOUT,BLK=22,BCD
        END
$*
$IBMAP  U03       4,DECK,M94,XR7
        ENTRY     .UN03.
.UN03.  PZE       UNIT03
UNIT03  FILE      ,A(3),INOUT,BLK=22,BCD
        END
$*
$IBMAP  U04       4,DECK,M94,XR7
        ENTRY     .UN04.
.UN04.  PZE       UNIT04
UNIT04  FILE      ,A(4),INOUT,BLK=22,BCD
        END
$*
$IBMAP  U07       4,DECK,M94,XR7
        ENTRY     .UN07.
.UN07.  PZE       UNIT07
UNIT07  FILE      A(7),INOUT,BLK=22,BCD
        END
$*
$IBMAP  U08       4,DECK,M94,XR7
        ENTRY     .UN08
.UN08.  PZE       UNIT08
UNIT08  FILE      A(8),INOUT,BLK=22,BCD
        END
$*
$IBMAP  U09       4,DECK,M94,XR7
        ENTRY     .UN09.
.UN09.  PZE       UNIT09
UNIT09  FILE      A(9),INOUT,BLK=22,BCD
        END
$*
$IBMAP  U10       4,DECK,M94,XR7
        ENTRY     .UN10.
.UN10.  PZE       UNIT10
UNIT10  FILE      B(0),INOUT,BLK=22,BCD
        END
$*
$IBMAP  U11       4,DECK,M94,XR7
        ENTRY     .UN11.
```

```
.UN11. PZE     UNIT11
UNIT11 FILE    B(1),INOUT,BLK=22,BCD
       END
$*
$IBMAP U12     4,DECK,M94,XR7
       ENTRY   .UN12.
.UN12. PZE     UNIT12
UNIT12 FILE    B(2),INOUT,BLK=22,BCD
       END
$*
$IBMAP U13     4,DECK,M94,XR7
       ENTRY   .UN13.
.UN13. PZE     UNIT13
UNIT13 FILE    B(3),INOUT,BLK=256,BIN
       END
$*
$IBMAP U14     4,DECK,M94,XR7
       ENTRY   .UN14.
.UN14. PZE     UNIT14
UNIT14 FILE    B(4),INOUT,BLK=256,BIN
       END
$*
$IBFTC B01     DECK,M94,XR7,NOLIST,NODD
   Source deck S01
$ORIGIN        ONE,UT3
$IBFTC B02     DECK,M94,XR7,NOLIST,NODD
   Source deck S02
$ORIGIN        ONE,UT3
$INCLUDE       CALC,NEW,RESULT,WENT,TOGOSS,DATA
$IBFTC B03     DECK,M94,XR7,NOLIST,NODD
   Source deck S03
$ORIGIN        TWO,UT3
$IBFTC B04     DECK,M94,XR7,NOLIST,NODD
   Source deck S04
$ORIGIN        TWO,UT3
$IBFTC B05     DECK,M94,XR7,NOLIST,NODD
   Source deck S05
$*
$IBFTC B06     DECK,M94,XR7,NOLIST,NODD
   Source deck S06
$ORIGIN        THREE,UT3
$IBFTC B07     DECK,M94,XR7,NOLIST,NODD
   Source deck S07
$*
$IBFTC B08     DECK,M94,XR7,NOLIST,NODD
   Source deck S08
$*
$IBFTC B09     DECK,M94,XR7,NOLIST,NODD
   Source deck S09
$*
$IBFTC B10     DECK,M94,XR7,NOLIST,NODD
```

```
   Source deck S10
$ORIGIN         THREE,UT3,REW
$IBFTC B11      DECK,M94,XR7,NOLIST,NODD
   Source deck S11
$*
$IBFTC B12      DECK,M94,XR7,NOLIST,NODD
   Source deck S12
$*
$IBFTC B13      DECK,M94,XR7,NOLIST,NODD
   Source deck S13
$*
$IBFTC B14      DECK,M94,XR7,NOLIST,NODD
   Source deck S14
$*
$IBFTC B15      DECK,M94,XR7,NOLIST,NODD
   Source deck S15
$*
$IBFTC B16      DECK,M94,XR7,NOLIST,NODD
   Source deck S16
$*
$IBFTC B17      DECK,M94,XR7,NOLIST,NODD
   Source deck S17
$*
$IBFTC B18      DECK,M94,XR7,NOLIST,NODD
   Source deck S18
$*
$IBFTC B19      DECK,M94,XR7,NOLIST,NODD
   Source deck S19
$*
$IBFTC B20      DECK,M94,XR7,NOLIST,NODD
   Source deck S20
$ORIGIN         TWO,UT3
$IBFTC B21      DECK,M94,XR7,NOLIST,NODD
   Source deck S21
$*
$IBFTC B22      DECK,M94,XR7,NOLIST,NODD
   Source deck S22
$ORIGIN         ONE,UT3
$IBFTC B23      DECK,M94,XR7,NOLIST,NODD
   Source deck S23
$*
$IBFTC B24      DECK,M94,XR7,NOLIST,NODD
   Source deck S24
$*
$IBFTC B25      DECK,M94,XR7,NOLIST,NODD
   Source deck S25
$*
$IBFTC B26      DECK,M94,XR7,NOLIST,NODD
   Source deck S26
$*
$IBFTC B27      DECK,M94,XR7,NOLIST,NODD
```

```
   Source deck S27
$*
$IBFTC B28      DECK,M94,XR7,NOLIST,NODD
   Source deck S28
$*
$IBFTC B29      DECK,M94,XR7,NOLIST,NODD
   Source deck S29
$*
$IBFTC B30      DECK,M94,XR7,NOLIST,NODD
   Source deck S30
$*
$IBFTC B31      DECK,M94,XR7,NOLIST,NODD
   Source deck S31
$*
$IBFTC B32      DECK,M94,XR7,NOLIST,NODD
   Source deck S32
$*
$IBFTC B33      DECK,M94,XR7,NOLIST,NODD
   Source deck S33
$*
$IBFTC B34      DECK,M94,XR7,NOLIST,NODD
   Source deck S34
$*
$IBFTC B35      DECK,M94,XR7,NOLIST,NODD
   Source deck S35
$*
$IBFTC B36      DECK,M94,XR7,NOLIST,NODD
   Source deck S36
$*
$IBFTC B37      DECK,M94,XR7,NOLIST,NODD
   Source deck S37
$ORIGIN         FOURE,UT3
$IBFTC B38      DECK,M94,XR7,NOLIST,NODD
   Source deck S38
$ORIGIN         FOURE,UT3,REW
$IBFTC B39      DECK,M94,XR7,NOLIST,NODD
$ORIGIN         ONE,UT3,REW
$IBFTC B40      DECK,M94,XR7,NOLIST,NODD
   Source deck S40
$*
$IBFTC B41      DECK,M94,XR7,NOLIST,NODD
   Source deck S41
$*
$IBFTC B42      DECK,M94,XR7,NOLIST,NODD
   Source deck S42
$DATA
   Input data deck
```

UNIT REWINDING

During the execution of the program the twelve intermediate storage units are rewound as required and this note indicates the points in the program where these rewinds occur. REWIND instructions are only executed for units required by a particular job, so that units 2, 3, and 4 of the "J-Series" and "L-Series" may not be rewound (or otherwise referenced) during a particular run. Also, when the number of sectors is set equal to 1, none of the "L-Series" and "K-Series" units are referenced. The following list is ordered in the way the program proceeds in execution:

Location	Units Rewound and Used
MAIN Program (If number of sectors exceeds 1)	KTAPE1 and KTAPE2 rewound
Subroutine INPUT (For new case only)	"J-Series" units required rewound, and then written
(Note: Point A is here)	
MAIN Program (if number of sectors exceeds 1)	LTAPE1 rewound, and then written
Subroutine READIN (For first sector only)	"J-Series" units required rewound, and then read
Subroutine PUNCH (For first sector only, and if number of sectors exceeds 1)	"L-Series" units required, LTAPE1 apart, rewound. "L-Series" units required written
(Note: If number of sectors equals 1, job terminates here)	
MAIN Program	"J-Series" and "L-Series" units required rewound
Subroutine THETA	KTAPE1 (or KTAPE2) read, KTAPE2 (or KTAPE1) written
Subroutine PLOT	KTAPE2 (or KTAPE1) rewound, and read. "J-Series" units required rewound, and read
Subroutine PCTURE (If Cartesian plots requested)	"L-Series" units required rewound, and read
MAIN Program	Identities of "K-Series" units switched, units rewound

(Control returns to Point A for cycles of calculation.)

BOUNDARY CONDITION MODIFICATION DURING COMPUTATION

There are two principal reasons why it may be desirable to modify the boundary conditions specified initially in the input data. Firstly, upon completing one computation, it may be desired to determine the solution for a second compressor operating condition; for example, at a different exit static pressure. The results of the previous calculation will probably be a better first approximation to the new solution than the axisymmetric assumption made initially by the program. Thus, the best procedure would be to employ the "J-Series" and "K-Series" tapes generated by the first calculation. However, a means is required to modify the specified exit static pressure recorded on unit JTAPE5. Secondly, some problems are not readily solved when the boundary conditions of interest are specified in the initial data to the program. Two cases when this has occurred are in the analysis of the GE/NASA Rotor 1B, and the analysis of the NACA Five-Stage Compressor. Both of these are described in Volume IX. In the former one, specification of the actual inlet total pressure profile leads, on the first cycle of calculation, to the result that the static pressure is highest in the region of low total pressure. Subsequent violent corrections occur if this calculation is pursued. In the latter case, the steepness of the characteristic leads to difficulties with the initially assumed constant sector widths.

These problems may be overcome by modifying the initially specified boundary conditions cycle by cycle. This can be accomplished by simple modifications to Subroutine READIN (source deck identifier: S04). For example, suppose that it is required to change the inlet total pressure specified initially. Then an instruction is inserted in Subroutine READIN following card S04 0050 and reading:

```
READ(5,26)(XMU(1,J),J=7,NTOPO)
```

Note that the standard IBM 7094 input unit number is assumed. This overwrites the inlet total pressures read from unit JTAPE1 previously, and causes the new values to be incorporated onto unit LTAPE1, and hence onto unit JTAPE1 for the next cycle. Of course, if the next cycle is also to be performed using the modified form of Subroutine READIN, data will again be required by the instruction inserted. The data requirements are NTHETA*NLOOPS lists of values of inlet total pressure, punched in the usual eight location real format, up to nine values to a card, with each list starting on a new card. The number of values in each list is NTOPO as specified in the initial data for the first cycle if the run is a new case (IFSTAR=0). Otherwise, and, in any case, for each subsequent cycle, NLINES values are required, NTOPO being reset to NLINES in Subroutine THETA. When the initial value of NTOPO prevails, the data values apply at the radii BHP initially specified; otherwise, they apply at the streamline radii at the first computing station specified by the inlet hub and casing radii (RHP(1) and RSP(1)), the inlet blockage (BLAM(1)), and the indicated split of the inlet annulus by the streamlines (DELM(1) through DELM(NLINES)). The additional data is placed after all the regular input data in which the specified inlet total pressure distribution (POCO) is nonconsequential.

Section III

OPERATIONAL DESCRIPTION OF PROGRAM

STEPWISE ANALYSIS PROCEDURE

Here the calculation performed by the computer program is described as a series of numbered steps. In addition to the number given to each step, there is also given the name of the subroutine that performs the step. (MAIN is used to indicate that a step is performed by the Main Program.)

1. MAIN - The units used by the program are assigned numerical values. Some data is read in, and a corresponding printout is produced.

2. MAIN - If the job is a new case, the following three steps are executed. Otherwise, control passes to Step 6.

3. INPUT - The compressor specification together with inlet conditions and outlet static pressure is read in, and a correspondin printout is produced.

4. INPUT - The number of units required to execute the job is determined, and a table giving this information is printed.

5. INPUT - The initial data for "Program DISTORT" is written onto the "J-Series" units.

6. MAIN - The remaining steps form a loop that is performed once for each of the specified number of cycles of operation of the program.

7. MAIN - Unless an axisymmetric analysis is desired (that is, the number of sectors has been set to 1), a small section of data is written onto unit LTAPE1.

8. MAIN - Steps 9 through 101 form a loop that is performed once for each sector.

9. DISTOR - Some constants are initialized.

10. READIN - Input data for the sector analysis is read from the "J-Series" units.

11. READIN - The relaxation factor used when reestimating the streamline radii is determined from

$$R_L = A_1 \cdot A_2$$

where R_L is the relaxation factor,
A_1 is the square of the ratio of the annulus height at a computing station to distance between the next station and the computing station, the largest value being used

and A_2 = 0.1667 if IFSIMP = 1 or 2
= 0.02383 = 3
= 0.007933 = 5
= 0.00361 = 7
= 0.00194 = 9

12. READIN - If specified, a printout of certain items of the input data is made.

13. READIN - On the first cycle only (in absolute terms), the radii of the streamlines are estimated. The streamlines at the hub (and casing) are given by the hub (and casing) radii, plus (or minus) the appropriate radial increment to account for the boundary layer blockage specified in the input data. At the first computing station, the radii of the remaining streamlines are determined from the specification in the input data of the increments of the annulus between the hub and each successive streamline. At all other computing stations, the annulus is divided by the streamlines so that each pair of streamlines enclose the same fraction of annulus area at all stations.

14. SOLVE - The curvature of the streamlines at the compressor inlet and outlet is set to zero.

15. GRAPH3 - The sector inlet conditions are interpolated from the input data.

16. LSQLNE, LSQFIT - Unless simple radial equilibrium is specified in the input data, the slopes and curvatures of the streamlines are determined by a least-squares polynomial curve fitting procedure. The number of points used in the fitting process is specified in the input data.

17. SOLVE - For the first pass only, the inlet velocity is estimated from continuity using the total density at the hub.

18. SOLVE - Steps 19 through 93 form a loop that is performed once for each computing station.

19. SOLVE - For the first pass, and at stations other than the first, the meridional velocity is estimated to be the mean of those previously computed at hub and casing at the previous computing station.

20. GRAPH3 - For stations other than the first, the following quantities are interpolated at the streamline radii:

$$\frac{1}{\rho}\frac{\partial P}{\partial \theta}, \quad \frac{A_I}{A_1}, \quad \mu, \quad \frac{\partial \phi}{\partial \theta}$$

21. SOLVE - If the station follows a blade row, control passes to Step 25.

22. GRAPH3 - The quantity $\int_{I-1}^{I} \frac{1}{\rho}\frac{\partial P}{\partial \theta}\cdot\frac{1}{C_m}\,dm$ is interpolated at the streamline radii.

23. SOLVE - The total temperature and pressure on each streamline is set equal to the corresponding value at the previous computing station.

24. SOLVE - Control passes to Step 68.

25. SOLVE - Steps 26 through 67 form a loop performed once for each streamline.

26. SOLVE - Parameters controlling blade performance, including relative inlet angle and Mach number, are determined.

27. INTERP - Blade specifications are interpolated on the streamline. If IBETA2 is 1, the relative outlet flow angle and loss coefficient are interpolated and then control passes to Step 65. If IBETA2 is 2 or 5, the blade camber angle, stagger angle, and point of maximum camber are interpolated at the streamline inlet and outlet radii. The camber and stagger angles are corrected for local streamline slope using the expression:

$$\Sigma_{corrected} = \arctan(\tan\Sigma\cos\phi)$$

The blade inlet and outlet angles are then determined from

$$\beta_{B1} = \gamma_1 + \arctan(X\cdot Y)$$

where

$$Y = \frac{-(1-X) + ((1-X)^2 - 4X\tan^2\theta_1)^{\frac{1}{2}}}{2X\tan\theta_1}$$

and

$$X = \frac{\bar{z}_1(2-3\bar{z}_1)}{(3\bar{z}_1-1)(\bar{z}_1-1)}$$

$$\beta_{B2} = \gamma_2 + \arctan(Y)$$

where $$Y = \frac{-(1-x) + ((1-x)^2 - 4x\tan^2\theta_2)^{\frac{1}{2}}}{2x\tan\theta_2}$$

and $$X = \frac{\bar{z}_1(2-3\bar{z}_1)}{(3\bar{z}_2-1)(\bar{z}_2-1)}$$

If IBETA2 is 4, the blade inlet and outlet angles are given directly by interpolation from the data, only requiring correction for local streamline slope using the expression given previously.

28. INTERP - A streamline camber angle is found from the difference between the blade outlet and inlet angles. A mean maximum camber point is found by interpolation at the mean of the streamline inlet and outlet radii. A streamline stagger angle is found from:

$$\gamma = \beta_{B1} - \chi$$

where $$\chi = \arctan(y \cdot z)$$

$$y = \frac{-(1-z) + ((1-z)^2 - 4z\tan^2\theta_2)^{\frac{1}{2}}}{2z\tan\theta}$$

$$z = \frac{\bar{z}(2-3\bar{z})}{(3\bar{z}-1)(\bar{z}-1)}$$

The blade thickness/chord ratio and solidity are interpolated at the mean streamline radius, and corrected for streamline slope using the expression

$$\Sigma_{corrected} = \Sigma\,\frac{\cos\phi_1 + \cos\phi_2}{2}$$

The specified loss coefficient and additional loss factor (if applicable) are interpolated at the streamline outlet radius. (All interpolations are made in Subroutine GRAPH3.)

29. CASCDE - The various indicators specifying the blade performance calculation to be made are selected from the appropriate subscripted arrays. A branch is made accoring to the indicator IBETA2.

30. CASCDE - If IBETA2 is 5, the loss coefficient is given directly from the input data. The relative outlet angle is given by

$$\beta_2 = \frac{\theta}{2} + \gamma - 0.17\frac{\theta}{\sigma}$$

On the first pass, control passes to Step 62. On subsequent passes, control passes to Step 60.

31. CASCDE - If IBETA2 is 2 or 4, Steps 32 through 62 are performed.

32. OPTANG - The minimum loss angle of attack is determined as a function of blade camber angle and cascade solidity. A series of second-order polynomial curves give angle of attack as a function of solidity for fixed camber angles, linear interpolation between the curves giving the final result.

33. CASCDE - The minimum loss inlet angle is given by the sum of the minimum loss angle of attack and the blade stagger angle. The empirical data was derived for blades having their maximum camber point at midchord. If the blade section under consideration has its maximum camber at other than midchord, the minimum loss inlet angle is adjusted, assuming that the incidence implied by the minimum loss inlet angle for a blade having its maximum camber point at midchord applies.

Thus $$\beta_{min.loss} = (\beta_{min.loss})_{a5} + \left(\frac{\theta}{2} + \gamma\right) - \beta_{B1}$$

and $$\beta_{B1} = \frac{\gamma + \arctan\left(X\left(-(1-X) + \left((1-X)^2 - 4X\tan^2\theta\right)^{\frac{1}{2}}\right)\right)}{2X\tan\theta}$$

and $$X = \frac{z(2-3z)}{(3z-1)(z-1)}$$

34. CASCDE - In order to allow a correction to the deflection achieved by the blade section for axial velocity-ratio, the axial velocity-ratio is determined. On the first pass, when no estimate of the outlet axial-velocity has been previously computed, the axial velocity-ratio is assumed to be unity. It is also set equal to unity if the data specifies that no axial velocity-ratio correction is to be made. In the general case, the axial velocity-ratio is computed from

$$VR = \frac{C_{m2}}{C_{m1}}\left(1 + VF\left(1 - \frac{C_{m2}}{C_{m1}}\right)\frac{C_{m1}}{C_{m2}}\right)$$

where

$$VF = 0 \quad \text{if} \quad \theta \geq 20^\circ$$

$$= \frac{20-\theta}{A_0} \quad \text{if} \quad \theta < 20^\circ$$

and VR is limited to the range of 0.8 to 1.25.
The minimum loss inlet angle is then corrected (for the purpose of determining the design deflection) according to the formula

$$\beta_{1,corrected} = \arctan\left(\frac{2\tan\beta_1}{1 + VR}\right)$$

35. STLANG - A series of second-order polynomials gives the stalling and choking inlet angles for cascade sections, curves giving the choking and stalling angles as functions of stagger for fixed cambers and solidities. The choking and stalling inlet angles are determined for the two nearest cambers, for solidities of 0.5, 1.0, and 1.5. For each of these solidities, the choking and stalling angles are determined (by linear interpolation) for the correct camber.

36. GRAPH2 - The choking and stalling inlet angles are determined for the actual solidity using the following expressions, as appropriate.

$$\beta = \beta_{1.0} + \frac{\beta_{0.5} - \beta_{1.0}}{0.4142}\left(\sigma^{-\frac{1}{2}} - 1.0\right)$$

if $\sigma < 1.0$

$$\beta = \beta_{1.5} + \frac{\beta_{1.0} - \beta_{1.5}}{0.1835}\left(\sigma^{-\frac{1}{2}} - 0.8165\right)$$

if $\sigma \geq 1.0$

37. OPTANG - The outlet flow deviation angle is determined from

$$\delta = m\theta\sigma^{-\frac{1}{2}}$$

where $m = m_1 (1 - w(1 - 2\bar{z}))$

$$w = -1.5478\,\gamma' + 2.07 \qquad \text{if } \gamma \leq 30$$

$$ = -0.8766\,\gamma' + 1.72 \qquad \text{if } \gamma > 30$$

$\gamma' = \gamma$ in radians

$$m_1 = 0.2158 + 0.0538\,\gamma' + 0.05\,\gamma'^2 + 0.06258\,\gamma'^3 - 0.0272\,\gamma'^4$$

38. OUTANG - Two corrections to be applied to the calculated deviation angle are computed. If the relative inlet Mach number onto the blade section is greater than 1.0, the outlet angle is decreased by 0.4 degrees. The outlet angle is increased by an amount given by

$$3.0 \exp(-0.22\,\theta) \cdot \rho$$

where ρ = 1.0 if the relative inlet Mach number is less than 1.0
= 0.0 if the relative inlet Mach number is greater than 1.3
= 4.0 - 3.8 M if the relative inlet Mach number (M) is between 1.0 and 1.3.

39. OUTANG - The relative outlet flow angle corresponding to the minimum loss inlet angle is determined by adding the deviation found in the two previous steps to the blade outlet angle, which is computed using the relationships described in Step 28.

40. OUTANG - The outlet flow angle is corrected for the effect of the inlet angle varying from the minimum loss value. (The computation will be ineffective during this use of Subroutine OUTANG). If the actual inlet angle is less than the minimum loss value (determined in Step 35), the outlet angle is assumed to be independent of the inlet angle, and equal to the value at minimum loss. If the actual inlet angle is greater than the minimum loss value, the rate of change of outlet angle is determined as a function of solidity, for two inlet angles. The two

inlet angles used are the two nearest to the actual inlet angle in a set of four that are 0, 30, 50, and 70 degrees. The following expressions give the rate of change of outlet angle with inlet angle.

$$\beta_1 = 0 \qquad \frac{d\beta_2}{d\beta_1} = 1.28 \exp(-3.46\sigma)$$

$$\beta_1 = 30^\circ \qquad \frac{d\beta_2}{d\beta_1} = 1.08 \exp(-2.98\sigma)$$

$$\beta_1 = 50^\circ \qquad \frac{d\beta_2}{d\beta_1} = 0.925 \exp(-0.9\sigma)$$

$$\beta_1 = 70^\circ \qquad \frac{d\beta_2}{d\beta_1} = 0.925 \exp(-0.9\sigma)$$

The rate of change of outlet angle with inlet angle is linearly interpolated for the actual inlet angle. The corrected outlet flow angle is then given by

$$\beta_2 = \beta_{2,\,min\ loss} + \frac{d\beta_2}{d\beta_1}\left(\beta_1 - \beta_{min\ loss}\right)$$

41. CASCDE - If the point of maximum camber of the blade section is not at midchord, the stalling and choking angles determined in Subroutine STLANG are modified. The incidence is maintained at that determined empirically for NACA 65-Series blades (which have their point of maximum camber at midchord). The procedure is that used previously in Step 27.

If the cascade solidity exceeds 1.6, a correction to the minimum loss, stalling, and choking angles is made. The corrected angles are given by

$$\beta_{corrected} = \beta + 6.07(\sigma - 1.6)$$

The stalling to minimum loss, and minimum loss to choking angle ranges are limited to not less than 2 degrees.

A correction is made to the minimum loss, stall, and choking angles for thickness/chord ratio. The corrected angles are given by

$$\beta_{corrected} = \beta + i(\rho - 1)$$

where i is the previously determined minimum loss incidence

and $$\rho = 0.6 + 0.4\frac{t}{c} \qquad \frac{t}{c} > 0.1$$

$$= 17.125\frac{t}{c} - 71.25\left(\frac{t}{c}\right)^2 \qquad \frac{t}{c} \leq 0.1$$

42. CASCDE - The minimum loss outlet angle is corrected for axial velocity ratio using the relation

$$\beta_{2,corrected} = \text{arc tan}\left(\frac{1}{2}\tan\beta_1\left(1 + \frac{1}{VR}\right)\right)$$

43. CASCDE - If specified in the input data, the minimum loss outlet angle is corrected for the thickness/chord ratio of the blade section. The correction is given by

$$\delta\frac{t}{c} = \theta\frac{m}{b}\left(\frac{t}{c}\,0.625 + 37.[illegible]\left(\frac{t}{c}\right)^2\right)$$

where $m = 0.17 - 0.000[illegible]3\,\beta_1 + 0.0[illegible]0333\,\beta_1^2$

$b = \sigma^x$

$x = 0.9[illegible]5 - 0.0000466\,\beta_1 - 0.000054\,\beta_1^2$

and β_1 is the minimum loss inlet angle determined in Step 15 prior to the correction for thickness/chord ratio.

Then $$\beta_{corrected} = \beta_2 + \delta\frac{t}{c}$$

where β_2 is the minimum loss outlet angle determined in Step 42.

44. CASCDE - The critical Mach number for the blade section is determined using the expressions

$$M_{CRIT} = 1.0 - 0.67CP + 0.3\,CP$$

where $CP = VM^2 - 1.0$

$VM = 1.03 + 0.7\frac{t}{c} + 0.4\frac{t}{c}\cos\beta_1(\tan\beta_1 [illegible]\,\beta_2)$

and β_1 is the minimum loss inlet angle determined in Step 15 prior to the correction for thickness/chord ratio,

β_2 is the minimum loss outlet angle determined in Step 43.

45. CASCDE - Unless the cascade analysis options specify the loss coefficient or minimum loss coefficient, the minimum loss coefficient is determined. The expression used is

$$\bar{\omega} = \frac{2\,\omega^{*}\,\sigma}{\cos\beta_2}$$

where $\omega^{*} = 0.003 + 0.02375\,D - 0.05\,D^2 + 0.125\,D^3$

$$D = VM - \frac{\cos\beta_1}{\cos\beta_2}$$

and β_1 and β_2 are the angles used to determine VM in Step 44.

46. CASCDE - Unless the cascade analysis options specify that no Mach number corrections are to be made to the loss coefficient, the minimum loss coefficient is corrected for Mach number, and the working range of the blade section at the actual relative inlet Mach number is determined.

The optimum loss coefficient (the minimum attainable value at the actual inlet Mach number) is given by

$$\bar{\omega}_{opt} = \bar{\omega}$$

if the relative inlet Mach number does not exceed critical;

$$\bar{\omega}_{opt} = \left(1.0 + 2(M - M_{CRIT})\right)\cdot\bar{\omega}$$

If the inlet Mach number exceeds the critical.

M is the relative inlet Mach number

Two alternative corrections in the blade section working range are available.

For medium Mach number sections, such as the NACA 65-Series blade, the corrections are

$$\beta_{1,min.loss,corrected} = \beta_{1,min.loss}$$

$$\beta_{1,stall,corrected} = \beta_{1,min.loss} + R(\beta_{1,stall} - \beta_{1,min\ loss})$$

$$\beta_{1,choke,corrected} = \beta_{1,min.loss} - R(\beta_{1,min.loss} - \beta_{1,choke})$$

where $R = exp(1.386(0.5-M))$

$\beta_{1,min.loss}$ is the minimum loss inlet angle determined in Step 41,
$\beta_{1,stall}$ is the stalling angle determined in Step 41, and
$\beta_{1,choke}$ is the choking inlet angle determined in Step 41.

For high Mach number sections, such as the double-circular-arc blade, the corrections are

$$\beta_{1,min.loss,corrected} = \beta_{1,min.loss}$$

$$\beta_{1,stall,corrected} = \beta_{1,min.loss} + (0.658 + 0.342 R)(\beta_{1,stall} - \beta_{1,min.loss})$$

$$\beta_{1,choke,corrected} = \beta_{1,min.loss} - R(\beta_{1,min.loss} - \beta_{1,choke})$$

47. CASCDE - The loss coefficient is determined from the inlet angle, and corrected reference angles determined in Step 46, using the expression

$$\bar{\omega} = \bar{\omega}_{opt}(1 + S^{XPLOS})$$

where
$$S = \frac{\beta_1 - \beta_{1,min.loss}}{\beta_{1,stall,corrected} - \beta_{1,min.loss}}$$

or
$$= \frac{\beta_1 - \beta_{1,min.loss}}{\beta_{1,choke,corrected} - \beta_{1,min.loss}}$$

the positive value being selected,

and *XPLOS* is 2.0 for high Mach number sections, and 4.5 for medium Mach number sections, unless the cascade analysis options specify an alternative value.

48. CASCDE - Should the cascade analysis options specify the loss coefficient, this value is selected.

The loss coefficient is limited to a maximum value of 0.25.

49. CASCDE - The actual inlet angle is corrected for axial velocity ratio using the methods described in Step 34.

50. OUTANG - The outlet angle corresponding to the corrected actual inlet angle determined in Step 49 is determined, except that if

this inlet angle exceeds the stalling inlet angle determined in Step 41 prior to the correction for thickness/chord ratio, then the stalling inlet angle is used. The method used to determine the outlet angle is as given in Steps 37 through 40.

51. CASCDE - If the inlet angle is greater than stalling, the outlet angle is determined by assuming the deflection at stall to apply.

52. CASCDE - The outlet angle is corrected for axial velocity ratio using the expression described in Step 42.

53. CASCDE - Unless the cascade analysis options specify otherwise, the outlet angle is corrected for thickness/chord ratio. The expressions used are those described in Step 43.

54. CASCDE - Unless the cascade analysis options specify otherwise, the effect of inlet Mach number upon outlet angle is assessed. If the relative inlet Mach number is greater than unity, it is replaced by value obtained by assuming the normal shock relationship to apply, that is,

$$M_{corr} = \left(\frac{M^2 + \frac{2}{\gamma - 1}}{\frac{2\gamma}{\gamma - 1} M^2 - 1} \right)^{\frac{1}{2}}$$

If the value so obtained is less than the critical Mach number, control passes to Step 59.

55. MCDEVN - The relative inlet velocity function at critical condition is obtained from

$$\frac{W_{1,c}}{\sqrt{T_0}} = \left(\frac{gJR}{\frac{1}{M_{1,c}^2} + \frac{\gamma - 1}{2}} \right)^{\frac{1}{2}}$$

56. MCDEVN - Continuity across the cascade section gives

$$M_{2,c} = \frac{A_1 \cos \beta_{1,c}}{A_2 \cos \beta_{2,c}} \cdot M_{1,c} \left\{ \frac{1 + \frac{\gamma - 1}{2} M_{2,c}^2}{1 + \frac{\gamma - 1}{2} M_{1,c}^2} \right\}^{\frac{\gamma + 1}{2(\gamma - 1)}}$$

where A_1 is the streamtube inlet area
A_2 is the streamtube outlet area
$\beta_{1,c}$ is the minimum loss inlet angle determined in Step 15 prior to the correction for thickness/chord ratio

$\beta_{2,c}$ is the minimum loss outlet angle determined in Step 43
$M_{1,c}$ is the critical inlet Mach number determined in Step 44
$M_{2,c}$ is the critical outlet Mach number

A first approximation to the critical outlet Mach number is obtained by setting $M_{2,c}$ to 1.0 on the R.H.S of the above equation, that is, using

$$M_{2,c} = \frac{A_1 \cos\beta_{1,c}}{A_2 \cos\beta_{2,c}} \cdot M_{1,c}\left(\frac{1}{1+\frac{\gamma-1}{2}M_{1,c}^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

$$= C_{11}$$

The value of $M_{2,c}$ is then refined by repeated application of the expression

$$M_{2,c} = C_{11}\left(1 + \frac{\gamma-1}{2}C_{12}^2\right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

where C_{11} is retained as defined above
and C_{12} is replaced by the latest estimate of $M_{2,c}$.

57. MCDEVN - The critical velocity ratio across the cascade is obtained from

$$\frac{w_{2,c}}{w_{1,c}} = \left(\frac{gJR}{\frac{1}{M_{1,c}^2}+\frac{\gamma-1}{2}}\right)^{\frac{1}{2}} \frac{\sqrt{T_0}}{w_{1,c}}$$

The actual inlet velocity function is determined from

$$\frac{w_1}{\sqrt{T_0}} = \left(\frac{gJR}{\frac{1}{M_1^2}+\frac{\gamma-1}{2}}\right)^{\frac{1}{2}}$$

By assuming the critical velocity ratio to apply, the outlet velocity function is found from

$$\frac{w_2}{\sqrt{T_0}} = \frac{w_1}{\sqrt{T_0}}\,\frac{w_{2,c}}{w_{1,c}}$$

A change in relative total temperature due to radius change of the streamline in a rotating blade row being ignored. The outlet Mach number corresponding to this velocity function is determined from

$$M_2 = \left(\frac{1}{\frac{gJRT_0}{w_2^2}-\frac{\gamma-1}{2}}\right)^{\frac{1}{2}}$$

58. MCDEVN - Continuity gives the outlet flow angle corresponding to the actual inlet Mach number and minimum loss inlet angle, using

$$\cos\beta_2 = \cos\beta_1 \frac{A_1}{A_2} \frac{w_1}{w_2} \left(\frac{1 + \frac{\gamma-1}{2} M_2^2}{1 + \frac{\gamma-1}{2} M_1^2} \right)^{\frac{1}{\gamma-1}}$$

59. CASCDE - The actual outlet angle is now corrected for the effects of Mach number using the relation $\beta_{2,corrected} = \beta_2 + X(\beta_{2M} - \beta_{2,min.loss})$

where β_2 is the outlet angle determined in Step 52
$\beta_{2,M}$ is the corrected outlet determined in Step 57
$\beta_{2,min.loss}$ is the minimum loss outlet angle determined in Step 44
X is 1.0, unless specified otherwise in the cascade analysis options

60. CASCDE - Except on the first pass, the blade row Reynolds number is calculated from

$$R_n = \frac{C_m \cdot \Delta x \cdot P}{\cos\beta_1 \cdot t \cdot \mu \cdot R}$$

where C_m is the inlet meridional velocity component
ΔX is the axial blade chord as defined by the two appropriate computing stations
P is the static pressure
β_1 is the actual inlet angle
t is the static temperature
μ is the viscosity, determined from $\mu = VIS1 + t \cdot VIS2$
and R is the gas constant.

Unless specified otherwise by the cascade analysis options, the loss coefficient is corrected for Reynolds number effects. The corrected loss coefficient is given by

$$\bar{\omega}_{corrected} = \bar{\omega} \quad \text{if} \quad R_n > 2.2 \cdot 10^5$$
$$\bar{\omega}_{corrected} = 2.0 \cdot \bar{\omega} \quad \text{if} \quad R_n < 0.5 \cdot 10^5$$
$$\bar{\omega}_{corrected} = \bar{\omega}(2.0 - 2.7386 \cdot RR + 1.2336 \cdot RR^3) \quad \text{if} \quad 2.2 \cdot 10^5 > R_n > 0.5 \cdot 10^5$$

where $RR = \log_e R_n - 10.82$

61. CASCDE - Unless specified otherwise by the cascade analysis options, the outlet flow angle is corrected for Reynolds number effects. The deviation due to Reynolds number is given by

$$\delta_R = 0.0 \quad \text{if} \quad R_n > 2.2 \cdot 10^5$$
$$\delta_R = 0.1(\beta_1 - \beta_2) \quad \text{if} \quad R_n < 0.5 \cdot 10^5$$

$$S_R = (\beta_1 - \beta_2)(0.1 - 0.137\,RR^2 + 0.0617\,RR^3)$$

if

$$2.2\cdot 10^5 > R_n > 0.5\cdot 10^5$$

and $$RR = \log_e(R_n) - 10.82$$

A check is made to determine if the Reynolds number correction to outlet angle is larger than that previously applied for Mach number effect. If it is, the Reynolds number correction above is applied. If it is not, the Reynolds number correction is ignored.

62. CASCDE - The outlet flow angle, loss coefficient, and min loss inlet angle are set for use by other sections of the computer program, and, if specified by the output options, the results of the cascade analysis are printed.

63. INTERP - The minimum loss inlet angle is signed. The tangent of the outlet angle is determined, and signed. The additional loss factor is applied to the loss coefficient, using the formula

$$\bar{\omega}_{corrected} = \bar{\omega}\,(1 + \rho)$$

where ρ = additional loss factor

64. INTERP - Should IBETA2 and ILOSS be 1, the preceding cascade analysis has not been performed, and the relative outlet angle and loss coefficient are set equal to values interpolated from the input data.

65. SOLVE - Unless IBETA2 is 1 or 5, or IFCAX is 2, provision is made to modify the relative outlet angle determined by the cascade analysis for axial velocity ratio as the outlet velocity is refined in the iterative determination of the velocity that follows. A number TANA is determined from

$$TANA = \tan\beta_2\left(1 + \frac{1}{VR}\right)$$

where VR is determined in Subroutine CASCDE in Step 35

66. SOLVE - The ratio of the actual to ideal (that is, no loss) relative outlet total pressures for the blade section is calculated using

$$\frac{P_{2r}}{P'_{2r}} = 1 - \bar{\omega}\left(1 - \left(\frac{t}{T_{1r}}\right)^{\frac{\gamma}{\gamma-1}}\right)\left(1 + \frac{V_2^2 - V_1^2}{2gJC_pT_{1r}}\right)^{\frac{\gamma}{\gamma-1}}$$

67. SOLVE - The indicator controlling the iterative solution of the continuity and momentum equations is set. Up to 15 attempts are allowed, except during the first pass, when only ten attempts are permitted. The following quantities which are not a function

of conditions to be iteratively determined are calculated.

$$\tan \gamma$$

where γ is station lean angle in r-z plane

$$\bar{A} = \left(1 + \tan^2\gamma + \tan^2\mu\right)^{-\frac{1}{2}}$$

where μ is station lean angle in r-θ plane

$$\bar{C} = \tan\gamma \cdot \bar{A}$$

$$\bar{A}\cos\phi - \bar{C}\sin\phi$$

$$\left(\bar{A}\sin\phi - \bar{C}\cos\phi\right)\frac{dC_m}{ds}$$

$$\frac{\tan\mu}{\rho r} \cdot \bar{A} \cdot \frac{dP}{d\theta}$$

68. SOLVE - The iterative solution of the momentum and continuity equations commences. Steps 64 and 65 form a loop performed once for each streamline.

69. SOLVE - If the station in the first station, or follows a blade-free space, this step is not executed. If axial velocity ratio corrections to the outlet angle from the blade row have been specified, the outlet flow is computed from

$$\tan\beta_2 = \frac{1}{2}\left(TANA\left(1 + \frac{1}{VR}\right) + \tan\beta_2\right)$$

where VR is computed as described in Step 65, using the latest estimate of C_{m2}

$$\tan\beta_3 = \frac{U_2}{C_{m2}} + \tan\beta_2$$

The absolute flow angle is calculated from

The total temperature is calculated from

$$T_2 = T_1 + \frac{U_2 C_{m2} \tan\beta_2 - U_1 C_{m1} \tan\beta_1}{gJC_p}$$

The total pressure is calculated from

$$P_3 = P_1 \left(\frac{T_3}{T_1}\right)^{\frac{\gamma}{\gamma-1}} \cdot \frac{P_{2r}}{P_{2r}'}$$

70. SOLVE - If the station follows a blade-free space, the absolute flow angle is calculated from

$$\tan\beta_3 = \frac{\tan\beta_1 \cdot C_{m1} \cdot r_1 - \int_{I-1}^{I} \frac{1}{\rho} \cdot \frac{dp}{d\theta} \cdot \frac{1}{C_m} \cdot dm}{r_2 C_{m2}}$$

71. SOLVE - The radial derivatives of total pressure, total temperature and the tangent of absolute flow angle are found at each streamline, using Subroutine RATE.

72. SOLVE - The main terms in the momentum equation are computed at each streamline:

$$A = \left\{ -(\bar{A}\cos\phi - \bar{C}\sin\phi)\left(\frac{1}{r_m} + \frac{\tan\alpha}{r} \cdot \frac{\partial\phi}{\partial\theta}\right) + \bar{A}\left(\tan\alpha \frac{d\tan\alpha}{dr} + \frac{\tan^2\alpha}{r}\right) - (\bar{A}\sin\phi + \bar{C}\cos\phi)\frac{dC_m}{ds} \cdot \frac{1}{C_m \cos\alpha} \right\} \cos^2\alpha$$

$$\frac{t}{T} = 1 - \frac{C_m^2(1 + \tan^2\beta)}{2gJC_pT}$$

$$B = \left\{ \bar{A}\left(gJC_p\frac{dT}{dr} - \frac{t}{T}\left(gJC_p\frac{dT}{dr} - \frac{1}{\rho T} \cdot \frac{dP}{dr}\right)\right) - \frac{\tan\mu}{\rho r} \cdot \bar{A} \frac{\partial P}{\partial\theta} \right\} \cos^2\alpha$$

73. SOLVE - The momentum equation is integrated from the starting point streamline near midstream to the outer wall, using the expressions

$$C_{m,J+1}^2 = C_{m,J}^2 \cdot e^{-(A_J + A_{J+1})\delta l} + \frac{B_J + B_{J+1}}{A_J + A_{J+1}}\left(1 - e^{-(A_J + A_{J+1})\delta l}\right)$$

or $C_{m,J+1} = C_{m,J}^2 + (B_J + B_{J+1})\delta l$

if $A_J + A_{J+1} = 0$

74. SOLVE - The momentum equation is similarly integrated from the starting point to the inner wall.

75. SOLVE - Steps 76 through 78 form a loop performed for each streamline. A check is made on the values of pro- duced by the calculation above. If any value is less than 1.0, it is replaced by a value of 1.0, and an indicator set to note the fact.

76. SOLVE - The meridional velocity is damped with the previously estimated value in the proportions one part the newly calculated value, and three parts the previous estimate. If the velocity so determined differs by more than 40 per cent from the previous estimate, the limiting value is imposed.

77. SOLVE - If the station follows a blade row, the absolute flow angle is reestimated from

$$\tan\beta_3 = \tan\beta_2' + \frac{U_2}{C_{m2}}$$

If the blade row rotates, the total temperature and pressure are reestimated from

$$T_2 = \frac{U_2 C_{m2} \tan\beta_3 - U_1 C_{m1} \tan\beta_1}{gJC_p} + T_1$$

$$P_3 = P_1\left(\frac{T_3}{T_1}\right)^{\frac{\gamma}{\gamma-1}} \cdot \frac{P_{2r}}{P_{2r}'}$$

If the station follows a blade-free space, the flow angle is reestimated from

$$\tan\beta_3 = \frac{\tan\beta_1 r_1 C_{m1} - \int_1^2 \frac{1}{C_m} \cdot \frac{1}{\rho} \cdot \frac{dp}{d\theta} dm}{r_2 C_{m2}}$$

78. SOLVE - The static temperature is calculated from

$$t = T - \frac{C_m^2 (1 + \tan^2 \beta_1)}{2g J C_p}$$

The static temperature is limited to a minimum value of 0.001. The static pressure is calculated from

$$p = \left(\frac{t}{T}\right)^{\frac{\gamma}{\gamma-1}} \cdot P$$

The specific weight is determined from

$$w = \frac{p}{Rt}$$

The Mach number controlling the solution type is determined from

$$M = C_m \left(\frac{1 + \tan^2 \alpha}{2gRt}\right)^{\frac{1}{2}}$$

where α is relative flow angle at blade row exits
α is absolute flow angle elsewhere

79. SOLVE - The total mass flow and mass-flow-weighted mean controlling Mach number are determined.

80. SOLVE - If the controlling Mach number is less than 1.0, and a subsonic solution has been specified, or the controlling Mach number is greater than or equal to 1.0, and a supersonic solution has been specified, Step 82 is next performed. Otherwise, Step 81 is performed.

81. SOLVE - A number is determined that is the ratio of the velocity giving the flow currently achieved by the computation, but on the other branch of the continuity curve. The expression used is

$$R = \left(\frac{1 + \frac{\gamma-1}{2} M^2}{M^4 \left(1 + \frac{\gamma-1}{2M^2}\right)}\right)^{\frac{1}{2}}$$

82. SOLVE - A number is determined that is the ratio of the velocity required to give the desired flow to the current velocity. The expression is

$$R = \left(\frac{w_{spec}}{w_{achieved}}\right)^{\frac{1}{1-M^2}}$$

$\left|\frac{1}{1-M^2}\right|$ is limited to 10.0
and R is limited to the range of 0.8 to 1.2

83. SOLVE - The meridional velocities at all radii multiplied by the number R .

84. SOLVE - Except during the first pass, and when IFBL is not equal to 2, Subroutine SOLVE calls Subroutine BLTHIC.

85. BLTHIC - Subroutine BLTHIC determines the displacement thickness of the boundary layers on the annulus hub and casing. This step is only performed for the first station. The momentum thickness on the hub, θ_h , is determined from

$$\theta_h = \frac{r_{streamline\ inner} - r_{hub}}{1.4}$$

If the specified blockage at the inlet is zero, θ_h will be zero, and the integral I , where

$$I_{HUB} = \int^{inlet} C_{m,HUB}^{4} \cdot dm$$

is set to zero. If θ_h is not zero, the integral is determined from

$$I_{HUB} = \left(\frac{\theta_h \cdot C_m^{3.4}}{0.006}\right)^{1.25}$$

where C_m is the meridional velocity on the innermost streamline at the first station.

This procedure is repeated for the casing.

86. BLTHIC - For stations other than the first, the integral I_{HUB} is completed to the station, using the expression

$$I_I = I_{I-1} + C_{m_I}^{4} \cdot \Delta X$$

where I_I is the integral to the station
I_{I-1} is the integral to the previous station
C_{m_I} is the meridional velocity on the innermost streamline at the station
and ΔX is the axial distance between the stations on the innermost streamline

The momentum thickness is determined from

$$\theta_{HUB} = \frac{0.006}{Cm_{,I}^{3.4}} \cdot I^{0.8}$$

The shape factor is determined from

$$H = \left(\frac{\theta_{HUB,I} - \theta_{HUB,I-1}}{\Delta x} \right) \cdot 30 + 1.5$$

and limited to values in the range

$$1.1 \leq H \leq 2.2$$

The displacement thickness is determined from

$$\delta_{HUB} = H_{HUB} \cdot \theta_{HUB}$$

and is limited to 0.25 times the metal-to-metal annulus heights at the station. A similar procedure produces the displacement thickness on the casing.

A check is made to ensure that the two boundary layer displacement thicknesses do not exceed 0.30 times the total annulus height The radius of the innermost streamline is taken as 0.2 of that value prescribed by the hub plus the displacement thickness determined above, and 0.8 times the previous estimate. The outermost streamline radius is similarly relaxed.

The radii of all streamlines between the innermost and outermost are adjusted so that the proportion of the total annulus area between each pair of streamlines is the same as it was before the inner and outermost were repositioned.

87. BLTHIC - For all stations, the fraction of the total annulus that is blocked by the boundaries is computed.

88. SOLVE - The convergence of the solution of the momentum and continuity equations at the station is checked. Convergence is deemed to have occurred if continuity is satisfied to within the tolerance specified in the data, and the velocity computed on the innermost streamline is within 0.2 per cent of the value previously estimated. If convergence has not occurred, and the maximum number of attempts permitted have not been performed, Step 70 is next performed. If convergence has occurred, Step 90 is next performed. Otherwise, Step 89 is performed.

89. SOLVE - The ratio of the flow specified to the flow achieved is checked. If it is less than 1.0, Step 90 is next performed. If it is greater than 1.0, a check on the maximum possible flow is made. The ratio of the maximum possible flow to the specified flow is computed from

$$\frac{w_{MAX}}{w_{SPEC}} = \frac{w_{ACHIEVED}}{M \cdot w_{SPEC}} \left(\frac{1 + \frac{\gamma-1}{2} M^2}{1 + \frac{\gamma-1}{2}} \right)^{\frac{\gamma+1}{2(\gamma-1)}}$$

If the ratio is less than one, this is noted by the setting of an indicator, unless the indicator has already been set to note surge at this station.

90. SOLVE - Unless surging or choking has already been noted for this pass at a previous computing station, an indicator is set to record the result at the computing station under examination.

91. SOLVE - At the first computing station only, the distribution of flow radially is noted. At subsequent computing stations, and on passes other than first, an estimate of the correct location of the streamlines is made by linearly interpolating for the radii at which the flow increments in inlet for each streamline occur.

The radii of the streamlines are then taken as

$$r = \frac{r_{n-1} \cdot RL + r_{interp}}{1 + RL}$$

where r_{n-1} is the previously assumed streamline radius
r_{interp} is the interpolated streamline location
and RL is the relaxation factor determined in Step 11

92. SOLVE - The loop performed for all stations is terminated.

93. SOLVE - An indicator noting convergence of the meridional velocity distribution is set if at all stations at the hub and casing the meridional velocities for the last two passes were within one half of 1 per cent of each other.

94. DISTOR - Except on Passes 3, 6, 9, ..., control passes to Step 96.

95. SEARCH - The procedure used in Subroutine SEARCH does not lend itself to this type of documentation, and for a rigorous description of the possible steps performed, the reader is referred to the detailed description of the subroutine, in particular, the flow diagram. An outline of the method used is given here.

An indicator is examined which takes one of three values, to indicate either that the flow in the compressor is surged, or choked, or that a valid solution has been obtained.

If the indication is that a valid solution has been obtained, the mass-flow-weighted mean exit static pressure is determined for comparison with the specified value. If during the first entry to Subroutine SEARCH for which a valid solution is indicated the calculation exit static pressure is within 0.2 per cent of the specified value, the estimate of the flow made previously is unchanged. The flow and exit static pressure are noted. If during the first entry to Subroutine SEARCH for which a valid solution is noted, the calculated exit static pressure is below (above) the specified value, the new estimate of the flow is taken as 0.45 of the estimated characteristic width less (more) than the original estimate.

On entries to Subroutine SEARCH after the first in which a valid solution is indicated, the flow required to give the specified downstream static pressure is reestimated by linear interpolation (or extrapolation) from the last two flow/pressure points obtained. Generally, four or five such operations will determine the flow with sufficient accuracy to give the exit static pressure within the specified tolerance of 0.2 per cent.

If on the first entry to Subroutine SEARCH, the indicator is of choked (surged) flow, the flow is reestimated by decreasing (increasing) the original estimate by 0.45 of the estimated characteristic width. This procedure is repeated until an indication of a valid solution is obtained.

If on any entry to Subroutine SEARCH, the indication is of choked (surged) flow, and a valid solution has previously been recorded, the flow is reestimated by decreasing (increasing) the previous estimate by "nesting". Nesting means that the new estimate of the flow is taken to be the mean of the flow giving the failure indication, and the previously recorded flow which gave a valid solution indication.

If two successive (valid) points define a characteristic of zero slope, the flow is reestimated to be the mean of the two previous flows.

If two successive (valid) points define a characteristic of positive slope, the flow is reestimated by increasing the larger of the two previous flows by 0.45 of the estimated characteristic width, except that if two previous points have defined a characteristic of negative slope, and a point recorded after the first valid indication entry to Subroutine SEARCH defined an exit static pressure lower than specified, the flow is reestimated as the mean of the last two estimates of the flow.

96. DISTOR - If the mass-flow-weighted mean exit static pressure last determined by Subroutine SEARCH is not within 0.2 per cent of the specified value, the indicator set by Subroutine SOLVE in Step 94 is reset to indicate that the solution is unconverged.

97. DISTOR - If the solution is converged and annulus wall boundary layer calculations are specified in the input data, and subject to the control of the indicators IFRPM and IFRPIN as described in the section entitled "Input Data", the annulus boundary layer blockages are printed.

98. PRINT - If the solution is converged, or the total permitted number of passes have been performed, or printout has been specified by the indicators NWRIT and INCWRI for the current pass, and subject to the control of the indicators IFRPM and IFPRIN, Subroutine PRINT produces a printout of conditions within the sector.

99. PUNCH - If the solution is converged, or the total permitted number of passes have been performed, and the specified number of sectors is greater than 1, Subroutine PUNCH writes the sector results onto the "L-Series" units for use in "Program THETA".

100. DISTOR - If the solution is unconverged, and the maximum permitted number of passes have not been performed, control passes to Step 14.

101. MAIN - If one sector only is specified, execution terminates.

102. THETA (Section I) - Define initial unit numbers and principal axial stations. Specify various constants.

103. THETA - Read general input, including title and sector centerline angles.

104. THETA - For the first tape only, read for each sector the title, the sector flow, and an indicator governing the reading of additional input.

105. THETA - As governed by this indicator, read only for the first sector general compressor data, consisting of indicators I1-I22 and constants R1-R5.

106. THETA - For the first sector only, read for every axial station of the current input tape, additional indicators (I23-I32) characterizing the blade row input specified at that station.

107. THETA - If the station is not the trailing edge of a blade row, proceed to 109.

108. THETA - For every blade row station, read the pertinent cascade information.

109. THETA - For all axial stations, read compressor data R21-R29.

110. THETA - For axial stations with lean only, read an indicator and the lean angle.

111. THETA - For all axial stations of the current input unit, all sectors, and all streamline locations, read main input arrays of streamline radius, pressure, temperature, meridional velocity, tangent of absolute air angle, and streamline slope.

112. THETA (Section II) - For the first input tape only, define the output arrays at the first axial station (total pressure, temperature, air angle) and prepare other information needed at this station for regular procedure of the computation, such as static pressure, temperature, sector centerline location, static pressure gradient.

113. OVERAL - The flow passing through the compressor is determined by summing the flow through each of the streamtubes defined by the sector boundaries and radially distributed streamlines. Conditions are assumed to remain constant circumferentially within each streamtube, and to be the mean of the conditions at the inner and outer radii thereof. Then the flow increments are given by

$$\Delta W = \frac{\Delta\theta_n}{2}\left(C_{z,n,J} + C_{z,n,J+1}\right)\left(\left(\frac{P}{Rt}\right)_{n,J} + \left(\frac{P}{Rt}\right)_{n,J+1}\right)\left(r_{n,J+1}^2 - r_{n,J}^2\right)$$

and the total flow is

$$W = \sum_{J=1}^{J=\mathrm{NLINES}-1}\left(\sum_{n=1}^{n=\mathrm{NTHETA}} \Delta W\right)$$

This calculation is performed for the first computing station. The mass-flow-weighted mean total temperature in the inlet is determined from

$$T = \frac{\sum_J \left(\sum_n \frac{T_{J+1} + T_J}{2}\right)}{\sum_J \left(\sum_n \Delta W\right)}$$

114. THETA - For all axial stations of every input unit, define at all radial locations the average streamline radius (circumferential average) and the corresponding axial coordinate.

115. OVERAL - The following calculations are made for each computing station, except the first. A mean total pressure ratio is determined from

$$R = \left\{ \frac{\sum_J \left(\sum_n \left(R_{nJ}^{\frac{\gamma-1}{\gamma}} - 1 \right) \Delta W_{nJ} \right)}{W} + 1 \right\}^{\frac{\gamma}{\gamma-1}}$$

where   is the total pressure ratio from the inlet to the station under consideration in streamtube (n, J).

The mass-flow-weighted mean total temperature is determined, as in Step 113. The inlet-to-station isentropic efficiency is determined from

$$\eta_{is} = \frac{R^{\frac{\gamma-1}{\gamma}} - 1}{\Delta T} \cdot T_1$$

The inlet-to-station polytropic efficiency is determined from

$$\eta_{pol} = \frac{\log_e R}{\log_e \frac{T_2}{T_1}} \cdot \frac{\gamma - 1}{\gamma}$$

116. THETA - Interpolate the five data arrays at the reference radii and restore the results in these arrays. This is done by intermediate storage of quantities in singly-dimensioned arrays, from where the interpolated values are transferred back to the original arrays.

117. THETA - Begin processing these data for all current axial stations and sectors, for one streamline at a time. This procedure requires two-dimensional arrays for intermediate storage. Begin with the first streamline or radial location ($J = 1$). Specify the first axial station of the current input tape ($I = IZR1$). Define those quantities ($\theta, \frac{1}{p} \cdot \frac{\partial p}{\partial \theta}$) of the last upstream station that are required in addition to data stored in the five arrays. If Program THETA has been called previously, read for all current axial stations and sectors the previously computed θ and $\frac{1}{p} \cdot \frac{\partial p}{\partial \theta}$ results from the auxiliary unit KTAPE1. Initialize the indicators ICOUNT (counting the number of iterations in the θ-computation) and KATZ (indicating whether or not a constrained iteration on θ is being conducted upstream or downstream of the compressor).

118. THETA (Section III) - Compute the undamped increment of θ from Station I-1 to Station I, for all sectors at the constant radial location, J. This is done as follows:

$$\theta_I = \theta_{I-1} + \int_{I-1}^{I} \frac{\tan\alpha}{r}\, dm$$

where

$$\int_{I-1}^{I} \frac{\tan\alpha}{r}\, dm = \begin{cases} \dfrac{\tan\alpha_I - \tan\alpha_{I-1}}{r_I + r_{I-1}} \left((r_I - r_{I-1})^2 + (z_I - z_{I-1})^2 \right)^{\frac{1}{2}}, & \text{if } r_I \approx r_{I-1}, \\[2ex] \left[\left(\tan\alpha_I - \tan\alpha_{I-1} + \left(\dfrac{r_I \tan\alpha_{I-1} - r_{I-1}\tan\alpha_I}{r_I - r_{I-1}} \right) \log_e \dfrac{r_I}{r_{I-1}} \right) \cdot \right. \\[2ex] \left. \dfrac{\left((r_I - r_{I-1})^2 + (z_I - z_{I-1})^2 \right)^{\frac{1}{2}}}{r_I - r_{I-1}} \right], & \text{if } r_I \neq r_{I-1} \end{cases}$$

The latter expression is a closed form of the definite integral from I to I-1 when both $\tan\alpha$ and r are linear functions of the meridional distance and $r_I \neq r_{I-1}$.

119. THETA - Damping of θ_I is accomplished by using the value of a previous passage through THETA (recorded on KTAPE1 or initialized the first time by the program) and the damping factor RFAC1 as follows

$$(\tilde{\theta}_I)_{LP} = (\theta_I)_{LP} \cdot (1 - RFAC1) + (\tilde{\theta}_I)_{LP-1} \cdot RFAC1$$

The superscript $\sim$ is used to distinguish between damped and undamped θ and $\frac{1}{\rho} \cdot \frac{\partial P}{\partial \theta}$ (defined later).

120. THETA - The quantity transmitted to the output tape is not, $\tilde{\theta}$, the angular coordinate of the sector centerlines (at all sectors, stations, and streamlines), but the ratio of the angular sector width to the initial width at the first upstream station. This is defined and now calculated as follows

$$\left(\frac{\Delta\tilde{\theta}}{\Delta\theta_1} \right)_{I,J=const.} = \frac{\tilde{\theta}_{n+1} - \tilde{\theta}_{n-1}}{(\theta_1)_{n+1} - (\theta_1)_{n-1}} = \Delta\theta$$

(However, the angular coordinate $\tilde{\theta}$ is retained by the program for the ultimate computation of the angle μ.) A test is made to determine whether $\Delta\theta$ exceeds 2 or is less than 0.5. If this is found to be the case, control is directed to the next step; if not, proceed to 122.

121. THETA - A message is written describing the location of the problem and $\tan\alpha_I$ are printed. These are subsequently corrected to give a circumferentially uniform mean value of $\frac{\tan\alpha_I}{r_I}$ and $\frac{\tan\alpha_{I-1}}{r_{I-1}}$ as follows

$$\tan\alpha_I = \frac{\frac{1}{N}\sum_{n=1}^{N_0}\left(\frac{\tan\alpha_I}{r_I} + \frac{\tan\alpha_{I-1}}{r_{I-1}}\right) - \frac{\tan\alpha_{I-1}}{r_{I-1}}}{r_I}$$

This is only intended as an emergency procedure to insure continuation of the program. The modified ($\tan\alpha_I$) are then printed and control goes to Step 118.

122. THETA - Since the integration of $\frac{\tan\alpha}{r}$ proceeds only stepwise from one axial station to the next and the result is immediately damped with the result from a previous cycle, the undamped θ_I must be pre-preserved and θ_{I-1} updated for the next incremental advance. This is now done. At this point, a check is also made, whether or not this integration has reached either the compressor front face or the last downstream calculating station of the entire machine. If neither is the case, or if the mixing factor RFAC5 (used in Steps 131 and 133 to compute pressures from the perturbation analysis) is 1.0 or greater, control passes to Step 135.

123. THETA (Section IV) - If the θ-integration has reached either the last station upstream or downstream of the machine, a perturbation-analysis revision of $\tan\alpha$ for all upstream/downstream stations is called for. The procedures used (associated with the name KATZ) differ in the upstream (KATZ=1) from downstream region (KATZ=2), but in this program are performed by the same sections, the distinction being made by the value of the indicator, KATZ.

124. THETA - If KATZ=2 and ICOUNT=0, the following computations need not be repeated and control passes to Step 131.

125. THETA - Compute circumferential averages of density, meridional velocity, and air angle at the end stations of the upstream region and at the back face of the compressor.

126. THETA - Store the existing (unrevised) static pressures in the upstream region and the unrevised $\tan\alpha$ in the downstream region for purposes of moderating the results of the perturbation analysis procedure.

127. THETA - Define boundary-value functions in the upstream and downstream region.

$\frac{P_{TOTAL}}{\rho_{STATIC}}$ at the first upstream region

$\frac{P_{STATIC}}{\rho_{STATIC}}$ at the compressor front face

$\frac{P_{TOTAL}}{P_{STATIC}}$ at the compressor rear face

$(C_m)^2_{mean} \cdot \frac{\alpha - \alpha_{mean}}{\cos^2\alpha_{mean}}$ at the compressor rear face

128. FOUR - Determine $2N_\theta$ Fourier coefficients for the two boundary-value functions as a function of θ in each region. Ignore the first (zero-order) Fourier constant, which is the circumferential mean value of the function, since the analysis requires only the coefficients defining the deviation of the functions from the mean values.

129. NEWRAP - Calculate k_1, k_2 and W_0, $\hat{W}_1$, $\hat{W}_2$, and V_0, $\hat{V}_1$, $\hat{V}_2$ so that

a. $k_1 = -k_2 = .35$ arbitrarily;

b. W_0 and V_0 are the arithmetical averages of the mean axial and tangential velocities at the two ends of the channel, c. W and V exactly equal the mean axial and tangential velocities at the two ends of the channel, where

$$W = W_0 + \hat{W}_1 \cdot e^{k_1 \cdot z} + \hat{W}_2 \cdot e^{k_2 \cdot z}$$

$$V = V_0 + \hat{V}_1 \cdot e^{k_1 \cdot z} + \hat{V}_2 \cdot e^{k_2 \cdot z}$$

The analytical approximation of the mean velocities along the length of the channel is thus fully defined.

130. SMITH - This procedure is performed for each value of the Fourier index m $(m = 1, \cdots, NF)$, the constants of integration (A_m, B_m, C_m, $D_{m,l}$, $E_{m,l}$, Γ) of the perturbation potential function solution are calculated:

$$\varphi = \sum_{m=-\infty}^{+\infty} \Big\{ A_m \cdot e^{mz} + B_m \cdot e^{-jmS_0 z} + C_m \cdot e^{-mz} + \sum_{l=1}^{2} \Big[D_{m,l} \cdot e^{(k_l + m)z} + E_{m,l} \cdot e^{(k_l - jmS_0)z} + F_{m,l} \cdot e^{(k_l - m)z} \Big] \Big\} e^{jmy}$$

where

$$A_m = A_m^0 + \frac{A_m'}{W_0}$$

$$B_m = B_m^0 + \frac{B_m'}{W_0}$$

$$C_m = C_m^0 + \frac{C_m'}{W_0}$$

and all the constants except A_m, B_m, and C_m pertain specifically to the first-order correction to account for axially nonuniform W

and V. Define the m^{th} complex Fourier coefficients of the specified boundary functions: H_m in the case of stagnation pressure, P_m in the case of static pressure, and θ_m in the case of absolute flow angle. Define the m^{th} boundary-condition coefficient matrix $\{a_{ij}\}$ by which A_m^o, B_m^o, and C_m^o are defined:

$$\begin{Bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{Bmatrix} \begin{Bmatrix} A_m^o \\ B_m^o \\ C_m^o \end{Bmatrix} = \begin{Bmatrix} a_{14} \\ a_{24} \\ a_{34} \end{Bmatrix}$$

where if the i^{th} boundary condition specifies stagnation pressure at $z = Z_1$, then for W, V, and $\frac{dV}{dz}$ defined at Z_2:

$$a_{i1} = -jm \frac{dV}{dz} \cdot e^{mZ_1}$$
$$a_{i2} = -jm \left[\frac{dV}{dz} - jmW(1 + S_o^2)\right] \cdot e^{-jmS_oZ_1}$$
$$a_{i3} = -jm \frac{dV}{dz} \cdot e^{-mZ_1}$$
$$a_{i4} = -jm\, H_m$$

and if the i^{th} boundary condition specifies static pressure at $z = Z_2$, then for W, V, and $\frac{dV}{dz}$ defined at Z_2:

$$a_{i1} = m\left[mW + j\left(mV - \frac{dV}{dz}\right)\right] \cdot e^{mZ_2}$$
$$a_{i2} = m\left[mS_o(V - S_oW) - j\frac{dV}{dz}\right] \cdot e^{-jmS_oZ_2}$$
$$a_{i3} = m\left[mW - j\left(mV + \frac{dV}{dz}\right)\right] \cdot e^{-mZ_2}$$
$$a_{i4} = -jm P_m$$

and if the i^{th} boundary condition specifies flow angle at $z = Z_3$, then for W, V, and $\frac{dV}{dz}$ defined at Z_3:

$$a_{i1} = m[W + jV] \cdot e^{mZ_3}$$
$$a_{i2} = -jm[S_oW - V] \cdot e^{-jmS_oZ_3}$$
$$a_{i3} = -m[W - jV] \cdot e^{-mZ_3}$$
$$a_{i4} = \theta_m$$

A Gauss-Jordan solution is used to solve the three simultaneous, linear, algebraic boundary equations to determine A_m^o, B_m^o, and C_m^o. (This is done in Subroutine CSIMEQ). Define the particular solution coefficients ($D'_{m,l}$, $E'_{m,l}$, $F'_{m,l}$, $l = 1,2$) corresponding to the first-order correction for axially nonuniform mean velocities:

$$D'_{m,l} = A^{\circ}_{m} \cdot \frac{jm\,k_l^2\,\hat{V}_l}{k_l(k_l+2m)(k_l+m+jmS_0)}$$

$$E'_{m,l} = B^{\circ}_{m} \cdot \frac{m\left\{jk_l^2\hat{V}_l - m(1+S_0^2)\left[jm(S_0\hat{W}_l - \hat{V}_l) - k_l\hat{W}_l\right]\right\}}{k_l\left[k_l^2 - m^2(1+S_0^2) - 2jk_l m S_0\right]}$$

$$F'_{m,l} = C^{\circ}_{m}\,\frac{jm\,k_l^2\,\hat{V}_l}{k_l(k_l-2m)(k_l-m+jmS_0)}$$

Define the m^{th} boundary condition constant vector which, along with the previously defined boundary condition coefficient matrix $\{a_{ij}\}$, will define the constants of integration (A'_m, B'_m, and C'_m) for the mean velocity:

$$\begin{Bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{Bmatrix} \cdot \begin{Bmatrix} A'_m \\ B'_m \\ C'_m \end{Bmatrix} = \begin{Bmatrix} a'_{14} \\ a'_{24} \\ a'_{34} \end{Bmatrix}$$

where if the i^{th} boundary condition specifies stagnation pressure at $z=Z_1$, then for W, V, and $\frac{dV}{dz}$ defined at Z_1:

$$\begin{aligned} a'_{14} = \sum_{l=1}^{2} \Big\{ & -\left[k_l^2 W + 2k_l m W - jm\frac{dV}{dz}\right] D'_{m,l}\cdot e^{(k_l+m)Z_1} \\ & - \left[(k_l - jmS_0)^2 W - m^2 W - jm\frac{dV}{dz}\right] E'_{m,l}\cdot e^{(k_l-jmS_0)Z_1} \\ & - \left[k_l^2 W - 2k_l m W - jm\frac{dV}{dz}\right] F'_{m,l}\cdot e^{(k_l-m)Z_1} \Big\} \end{aligned}$$

and if the i^{th} boundary condition specifies static pressure at $z=Z_2$, then for W, V, and $\frac{dV}{dz}$ defined at Z_2:

$$a_{i4}' = \sum_{l=1}^{2} \Big\{ -\Big[W(k_l+m)^2 + jmV(k_l+m) - jm\frac{dV}{dz}\Big]\cdot D_{m,l}'\cdot e^{(k_l+m)Z_2}$$

$$- \Big[W(k_l - jmS_o)^2 + jmV(k_l - jmS_o) - jm\frac{dV}{dz}\Big]\cdot E_{m,l}'\cdot e^{(k_l - jmS_o)Z_2}$$

$$- \Big[W(k_l - m)^2 + jmV(k_l - m) - jm\frac{dV}{dz}\Big]\cdot F_{m,l}'\cdot e^{(k_l - m)Z_2} \Big\}$$

and if the i^{th} boundary condition specifies flow angle at $z = Z_3$, then for W, V, and $\frac{dV}{dz}$ defined at Z_3:

$$a_{i4}' = \sum_{l=1}^{2} \Big\{ -\Big[W(k_l+m) + jmV\Big]\cdot D_{m,l}'\cdot e^{(k_l+m)Z_3}$$

$$- \Big[W(k_l - jmS_o) + jmV\Big]\cdot E_{m,l}'\cdot e^{(k_l - jmS_o)Z_3}$$

$$- \Big[W(k_l - m) + jmV\Big]\cdot F_{m,l}'\cdot e^{(k_l - m)Z_3} \Big\}$$

A Gauss-Jordan solution is used to solve the three simultaneous, linear, algebraic boundary equations for the first-order correction constants of integration A_m', B_m', and C_m'. (This is done in Subroutine CSIMEQ.) The m^{th} set of constants of integration and particular solution constants for the perturbation potential function are put in final forms:

$$A_m = A_m^o + \frac{A_m'}{W_o}$$

$$B_m = B_m^o + \frac{B_m'}{W_o}$$

$$C_m = C_m^o + \frac{C_m'}{W_o}$$

$$D_m = \frac{D_m'}{W_o}$$

$$E_m = \frac{E_m'}{W_o}$$

$$F_m = \frac{F_m'}{W_o}$$

for $l = 1, 2$.

131. GEORGE - If KATZ=1 and ICOUNT=3, new air angles are determined in the downstream duct region. For the specified nondimensionalized axial location, $z = Z$, calculate the m complex Fourier coefficients ($m = 1, \cdots, NF$) of the perturbation tangential velocity, μ_m, and of the perturbation tangential velocity, ν_m:

$$\mu_m = -jm\left\{A_m \cdot e^{mz} + B_m \cdot e^{-jmS_o z} + C_m \cdot e^{-mz} + \sum_{l=1}^{2}\left[D_{m,l} \cdot e^{(k_l+m)z} + E_{m,l} \cdot e^{(k_l-jmS_o)z} + F_{m,l} \cdot e^{(k_l-m)z}\right]\right\}$$

$$\nu_m = \left\{mA_m e^{mz} - jmS_o B_m \cdot e^{-jmS_o z} - mC_m \cdot e^{-mz} + \sum_{l=1}^{2}\left[D_{m,l} \cdot e^{(k_l+m)z} + E_{m,l} \cdot e^{(k_l-jmS_o)z} + F_{m,l} \, e^{(k_l-m)z}\right]\right\}$$

In turn, for each of the i sectors ($i = 1, \cdots, NT$), the perturbation axial and tangential velocities, W and V, are calculated via Fourier summation:

$$W = 2 \cdot \sum_{m=1}^{NF}\left\{\mathrm{REAL}(\mu_m)\cos m y_i - \mathrm{IMAG}(\mu_m) \cdot \sin m y_i\right\}$$

$$V = 2 \cdot \sum_{m=1}^{NF}\left\{\mathrm{REAL}(\nu_m)\cos m y_i - \mathrm{IMAG}(\nu_m)\sin m y_i\right\}$$

The flow angles then follow directly, and these are "mixed" with the original values to moderate the influence of perturbation analysis, using

$$(\tan\alpha)^* = (1 - RFAC5)\frac{V}{W} + RFAC5 \tan\alpha_{original}$$

where

V is the circumferential velocity component

W is the meridional velocity component

$*$ denotes the "mixed" results

The indicator ICOUNT is incremented and controls revert to Step 118.

132. THETA - For the upstream region and for the first time through this section of the program only, compute the "mixing" factor RFAC5 which is progressively modified each time Program THETA is entered. If RFAC5 equals or exceeds 10, the solution will be completely "released", and control passes to Step 135.

133. EDWIN - For the specified nondimensionalized axial location, $z = Z$, calculate the m^{th} complex Fourier coefficients ($m = 1, \cdots, NF$) of the perturbation circumferential pressure function gradient, μ_m, and

of the perturbation pressure function, ν_m :

$$\mu_m = -\Big\{ m\left[mW + j\left(mV - \frac{dV}{dz}\right)\right]A_m \cdot e^{mZ}$$
$$+ \; m\left[mS_o(V - WS_o) - j\frac{dV}{dz}\right]B_m \cdot e^{-jmS_oZ}$$
$$+ \; m\left[mW - j\left(mV + \frac{dV}{dz}\right)\right]C_m \cdot e^{-mZ}$$
$$+ \sum_{l=1}^{2}\Big\{\left[(k_l + m)^2W + jm(k_l+m)V - jm\frac{dV}{dz}\right]D_{m,l}\cdot e^{(k_l+m)Z}$$
$$+ \left[(k_l - jmS_o)^2W + jm(k_l - jmS_o)V - jm\frac{dV}{dz}\right]E_{m,l}\cdot e^{(k_l - jmS_o)Z}$$
$$+ \left[(k_l - m)^2W + jm(k_l - m)V - jm\frac{dV}{dz}\right]F_{m,l}\cdot e^{(k_l - m)Z}\Big\}\Big\}$$

$$\nu_m = -\frac{j}{m}\mu_m$$

where W, V, and $\frac{dV}{dz}$ are all defined at Z. In turn, for each of the i sectors ($i = 1,\ldots,NT$), the perturbation pressure function and its circumferential gradient, $\frac{p}{\rho}$ and $\frac{1}{\rho}\frac{\partial P}{\partial y}$, are calculated via Fourier summation:

$$\frac{P}{\rho} = 2\sum_{m=1}^{NF}\left\{REAL(\nu_m)\cos my_i - IMAG(\nu_m)\sin my_i\right\}$$

$$\frac{1}{\rho}\frac{\partial P}{\partial y} = 2\sum_{m=1}^{NF}\left\{REAL(\mu_m)\cos my_i - IMAG(\nu_m)\sin my_i\right\}$$

(The static pressure gradient is not used.) The static pressures in the flow field are modified by "mixing" the perturbation solution values obtained here with the original values, again using the mixing factor RFAC5.

134. THETA - Compute: the damped static pressure gradient

$$\widetilde{\left(\frac{1}{\rho}\frac{\partial P}{\partial \theta}\right)} = \frac{1}{\rho_n}\cdot\frac{P^*_{n+1} - P^*_{n-1}}{\theta_{n+1} - \theta_{n-1}}(1 - RFAC1) + RFAC1\cdot\left(\frac{1}{\rho}\frac{\partial P}{\partial \theta}\right)_{previous}$$

The corresponding change in angular momentum

$$\Delta rC_\theta = \frac{1}{2}\left(\frac{\left(\frac{1}{\rho}\cdot\frac{\partial P}{\partial \theta}\right)_I}{C_{m,I}} + \frac{\left(\frac{1}{\rho}\cdot\frac{\partial P}{\partial \theta}\right)_{I-1}}{C_{m,I-1}}\right)\left((r_I - r_{I-1})^2 + (z_I - z_{I-1})^2\right)^{\frac{1}{2}}$$

For the upstream region only, and if ICOUNT is less than IFAC2

$$\tan\alpha_I = \frac{\Delta rC_\theta - r_{I-1}\cdot C_{m,I-1}\cdot \tan\alpha_{I-1}}{r_I\cdot C_{m,I}}$$

Increment ICOUNT and return to the appropriate section of Step 118. The number of iterative determinations of $\tilde{\theta}$, p, and α is specified in the input of IFAC2. This iteration is necessary since the perturbation analysis in both regions is based on assumed $\tilde{\theta}$ and yields new p and α corresponding to a revised $\tilde{\theta}$. For the downstream region, Steps 125 through 130 need not be repeated since $\tilde{\theta}$ at the compressor rear face remains unchanged, this being the starting point of the downstream perturbation analysis. Step 124 accomplishes this bypass after the first pass through the calculation. If ICOUNT=IFAC2, control passes to Step 137.

135. THETA (Section V) - In this section the regular computation of the damped static pressure gradient $(\frac{1}{\rho}\frac{\partial p}{\partial \theta})$ and, for the blade-free spaces, the corresponding change in angular momentum, ΔrC_θ, and of the air angle, α, is performed when the perturbation analysis is bypassed. The following procedure is employed here for updating the "previous" static pressure gradient, using the factor RF1

$$\left(\frac{1}{\rho}\frac{\partial p}{\partial \theta}\right)_{\substack{previous,\\ modified}} = (1-RF1)\left(\frac{1}{\rho}\frac{\partial p}{\partial \theta}\right)_{previous} + \frac{1}{\rho_n}\cdot\frac{p_{n+1}-p_{n-1}}{\tilde{\theta}_{n+1}-\tilde{\theta}_{n-1}}\cdot RF1$$

The final value of $(\frac{1}{\rho}\cdot\frac{\partial p}{\partial \theta})$ is set equal to $(\frac{1}{\rho}\frac{\partial p}{\partial \theta})_{\substack{prev.\\ mod.}}$ before being transferred to the output tape. Control then passes to Step 137.

136. If the last station has not yet been reached by the stepwise integration procedure of θ, I is incremented (next axial station) and, if I is less or equal to the last station, control goes to an appropriate section of Step 119.

137. THETA (Section VI) - At this point, all the data processing but for the determination of the angle μ, has been completed for all axial stations of the current unit and for all sectors at a fixed streamline location (J=constant). In order to continue this process for a new streamline location, the auxiliary two-dimensional arrays must be vacated and the pertinent information stored in the original data arrays without erasing data that is still required.

The arrays assignment is as follows:

The radius array is maintained.

$(\frac{1}{\rho}\cdot\frac{\partial p}{\partial \theta})$ is stored in the array of static pressures, p ("screen" or first upstream station excluded).

$\Delta\theta$ is stored in the array of static temperatures, t ("screen" or first upstream station excluded).

ΔrC_θ is stored in the array of meridional velocities, C_m ("screen" or first upstream station excluded).

$\tilde{\theta}$ is stored temporarily in the array of $\tan\alpha$.

$\frac{\partial\phi}{\partial\theta}$ is stored in the array of the streamline slope angle, ϕ, after it has been calculated by the formula:

$$\frac{\partial\phi}{\partial\theta} = \frac{\phi_{n+1} - \phi_{n-1}}{\tilde{\theta}_{n+1} - \tilde{\theta}_{n-1}}$$

The quantities that must be preserved for the processing of a subsequent unit are:

- C_m at all sectors and streamlines of the last station;
- $\tan\alpha$ at all sectors and streamlines of the last station;
- θ (undamped) at all sectors and streamlines of the last station;
- $\frac{1}{\rho}\frac{\partial p}{\partial\theta}$ at all sectors and streamlines of the last station;
- r_{Ref} and z_{Ref} at all streamlines of the last station.

This is achieved by increasing the axial array size (I) by one location for all six three-dimensional input arrays. The input is read into the arrays beginning at relative Station I=2 (leaving I=1 free for transferred data). The output is stored in the arrays beginning at I=1 (leaving the last relative station location for the preservation of C_m, $\tan\alpha$, $(\frac{1}{\rho}\frac{\partial\tilde{p}}{\partial\theta})$, θ, $\tilde{\theta}$). The auxiliary two-dimensional arrays are also increased by one axial station in order to permit a consistent treatment.

138. THETA - Concurrently, $\tilde{\theta}$ and $(\frac{1}{\rho}\frac{\partial\tilde{p}}{\partial\theta})$ are written on KTAPE2 for reference when "Program THETA" is reentered.

139. THETA - J is now incremented and the processing of data repeated for a new streamline location, starting at an appropriate segment of Step 118 until the maximum J is exceeded.

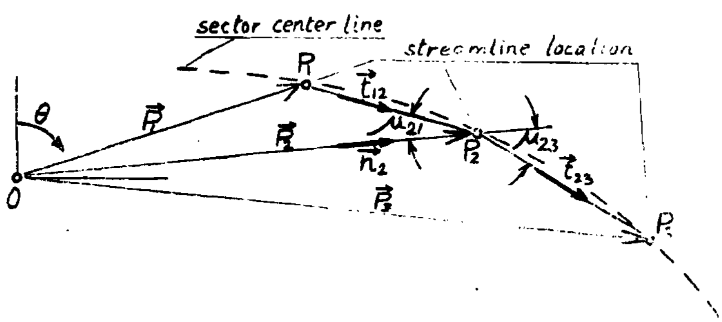
140. THETA - $\tilde{\theta}$ is printed since it will not be transfered later.

141. THETA (Section VII) - When the data have been processed for all streamlines, sectors, and stations of the current input tape, the angle μ is calculated as follows:

$$\vec{t}_{12} = \frac{\vec{P}_2 - \vec{P}_1}{|\vec{P}_2 - \vec{P}_1|}$$

$$\vec{t}_{23} = \frac{\vec{P}_3 - \vec{P}_2}{|\vec{P}_3 - \vec{P}_2|}$$

$$\vec{n}_2 = \frac{\vec{P}_2 - \vec{O}}{|\vec{P}_2 - \vec{O}|}$$



$$\mu_{21} = \arctan\left(\frac{\vec{n}_2 \times \vec{t}_{12}}{\vec{n}_2 \cdot \vec{t}_{12}}\right)$$

$$\mu_{23} = \arctan\left(\frac{\vec{n}_2 \times \vec{t}_{23}}{\vec{n}_2 \cdot \vec{t}_{23}}\right)$$

$$\mu_2 = \frac{1}{2}(\mu_{21} + \mu_{23}) \qquad (at\ P_2)$$

142. THETA (Section VIII) - For the first output unit write onto the appropriate J-unit the following general input information:

The Sector Heading
Indicators I1- 22
Constants R1-R5
Arrays r , P_{TOTAL} , T_{TOTAL} , α , ϕ , for all streamlines at the first station.

Define the first axial station for regular output (IZA1).

143. THETA - Write the absolute station numbers of the first and last station defining the section for which output is about to be written.

144. THETA - Write for each station I23(I) I32(I); if I23(I)=6, write the following:

If I23(I)=2 or 4, and I30(I)=3: EXPLOS
If I23(I)=2 or 4, and I26(I)=3: XMACH
Write R13-R20 for all blade stations
Write r , r_{Ref} , μ , $\frac{1}{\rho}\frac{\partial \tilde{P}}{\partial \theta}$, $\frac{\partial \phi}{\partial \theta}$, $\Delta\theta$, for all streamlines and proceed to Step 146.

145. THETA - For each station if I23(I)=6, write r , r_{Ref} , $\frac{1}{\rho}\frac{\partial \tilde{P}}{\partial \theta}$, $\frac{\partial \phi}{\partial \theta}$, $\Delta\theta$, $\Delta r C_\theta$ for all streamlines.

146. THETA - Write next JTAPE unit number.

147. THETA - If the current tape unit is the last one, write R21(J)-R29(J) at each blade radius if I21(I)=1, write I33(I) and R30(I).

148. THETA - If the current tape unit is the last one, go to 150.

149. THETA (Section IX) - Transfer r_{Ref} and z_{Ref} of the last axial station to the first axial station. Transfer $\frac{1}{\rho}\frac{\partial \tilde{P}}{\partial \theta}$, $\tan\alpha$, C_m , and $\tilde{\theta}$ of the last axial station to the first axial station of the new arrays of r , $\tan\alpha$, C_m , ϕ , respectively. Now define the new input and output tape unit numbers, the absolute and relative station numbers defining the next segment of the compressor and return to Step 111.

150. OVERAL - The weight flow computed in Step 113, the efficiencies computed in Step 115, and the speed of rotation of the first nonstationary blade are printed.

151. PCTURE - Polar plots showing the location of each sector centerline at every station are produced (by Subroutine PLOT). Subject to the control of the indicator IFPRIN, Cartesian plots of conditions at each computing station are produced (by Subroutine MPLOT).

152. MAIN - The identities of the "K-Series" units are switched for the next cycle of computation.

153. MAIN - This point is the termination of one cycle of calculation.

154. MAIN - Upon completion of all specified cycles, execution terminates.

SUBROUTINE DESCRIPTIONS

MAIN PROGRAM

The function of the MAIN Program is to control the overall logic of the program, as is illustrated by Figure 2. This overall program flow chart is also a flow chart for the main program, inasmuch as the subroutine calls are indicated by references to "Program INPUT", "Program DISTORT", "Program THETA", and "Program PICTURE". These are the four major overlays into which the program is divided; the actual corresponding FORTRAN calls are to Subroutine INPUT, Subroutine DISTOR, Subroutine THETA, and Subroutine PCTURE.

The following relates the program to the STEPWISE ANALYSIS PROCEDURE:

Step Number	Cards (Identifier S01)
1	0004 through 0034
2	0035 and 0036
6	0037 and 0075
7	0038 through 0043
8	0046 through 0048
101	0049
152	0070 through 0074
153	0075
154	0076

SUBROUTINE INPUT

The function of Subroutine INPUT is to read in the bulk of the input data required to start a new job, and to prepare the initial "J-Series" input data sets for Subroutine READIN of "Program DISTORT". Note that the subroutine is only called when the first computer run of a job is made; subsequently, the "J-Series" units are prepared by Subroutine THETA. The subroutine is called by the MAIN Program. A flow diagram for the subroutine is shown on Figure 7.

The following relates the subroutine to the STEPWISE ANALYSIS PROCEDURE:

Step Number	Cards (Identifier S02)
3	0009 through 0130
4	0131 through 0185
5	0186 through 0279

SUBROUTINE DISTOR

The function of Subroutine DISTOR is to control the logic of "Program DISTORT", the second major overlay of the program. Figure 3, a flow diagram for "Program DISTORT", serves also as a flow diagram for Subroutine DISTOR. The subroutine is called by the MAIN Program, and calls Subroutine READIN, Subroutine SOLVE, Subroutine SEARCH, Subroutine PRINT, and Subroutine PUNCH.

The following relates the subroutine to the STEPWISE ANALYSIS PRO CEDURE:

Step Number	Cards (Identifier S03)
9	0018 through 0027
94	0032 through 0037
96	0039
97	0040 through 0044
100	0051

SUBROUTINE READIN

The functions of Subroutine READIN are to read from the "J-Series" units the input data for the particular sector analysis, to determine the relaxation factor for the (radial) streamline relocations, to print certain of the input data upon request, and, for the first cycle of computation only, to estimate the streamline locations initially. The subroutine is called by Subroutine DISTOR.

The following relates the subroutine to the STEPWISE ANALYSIS PROCEDURE:

Step Number	Cards (Identifier S04)
10	0019 through 0115
11	0018 through 0138
12	0144 through 0189
13	0192 through 0215

SUBROUTINE SOLVE

The function of Subroutine SOLVE is to compute a solution to the momentum and continuity equations at each computing station. The subroutine is called by Subroutine DISTOR, and calls Subroutine GRAPH3, Subroutine LSQLNE, Subroutine INTERP, and Subroutine BLTHIC. A flow diagram for the subroutine is shown on Figure 8.

The following relates the subroutine to the STEPWISE ANALYSIS PROCEDURE:

Step Number	Cards (Identifier S05)
14	0026 through 0028
17	0097 through 0100
18	0101
19	0103 through 0105
21	0117
23	0118 through 0121
24	0122
25	0124
26	0125 through 0134
65	0138 through 0147
66	0148 and 0149
67	0150 and 0169
68	0170
69	0177 through 0188
70	0190
71	0192 through 0199
72	0200 through 0206
73	0207 through 0218
74	0219 through 0230
75	0231 through 0236
76	0239 through 0247
77	0248 through 0260
78	0261 through 0271
79	0272 through 0277
80	0278 and 0279
81	0280 through 0286
82	0288 through 0303
83	0304 through 0308
84	0309
88	0310 through 0316

Step Number	Cards (Identifier S05)
89	0317 through 0326
90	0329
91	0330 through 0348
92	0350
93	0351 through 0355

SUBROUTINE GRAPH3

The function of Subroutine GRAPH3 is to linearly interpolate input data at streamline radii. The subroutine is called by Subroutine SOLVE and Subroutine INTERP. The following steps in the STEPWISE ANALYSIS PROCEDURE involve Subroutine GRAPH3. Card identifier: S06.

Step Number
15
20
22
27
28

SUBROUTINE LSQLNE

The function of the routine is to supervise the evaluation of the coefficients of a second-order polynomial curve yielding a least-squares fit to the streamlines. The subroutine is called by Subroutine SOLVE, and calls Subroutine LSQFIT. Step 16 of the STEPWISE ANALYSIS PROCEDURE is performed by the subroutine. Card identifier: S07.

In general, the standard procedure used in the subroutine involves fitting a least-squares parabola through a group of NPTS points of the Y versus EX curve. The values of DYDX(LM,J) and D2YDX2(LM,J) are then calculated using the coefficients of the parabola and the value of EX(LM), where LM refers to the index of the middle point of the group. The equations are as follows:

DYDX(LM,J)=ACO(2)+2.0·ACO(3)·EX(LM)

D2YDX2(LM,J)=2.0·ACO(3)

where ACO(2) and ACO(3) are the second and third coefficients of the least-squares parabola computed by Subroutine LSQFIT. The procedure is repeated increasing LM in increments of one until all of the derivatives have been calculated. The value of NPTS can be either 5, 7, 9, or 11 whichever is specified in the input to the program.

The standard fitting procedure described above cannot be used, without some modification, to calculate derivatives at a number of stations near the beginning or end of the machine. For example, with NPTS=7 station 2 cannot be treated in the normal manner since there are less than three preceding stations and the procedure requires that the station under consideration be the middle of a group of NPTS stations.

It can be seen that the first and last stations at which the standard fitting procedure can be used are I1 and I2, respectively, defined as follows:

$$I1 = \frac{NPTS+1}{2}$$

$$I2 = NX - \frac{NPTS-1}{2}$$

For stations whose indices are less than I1 or greater than I2, the fitting procedure must be modified. This is done by setting up "dummy" stations at the beginning and end of the machine in sufficient number to satisfy the condition that there be (NPTS-1)/2 stations on either side of the one being considered. The values of the dependent variable at dummy stations placed before the first axial station are set equal to Y(1,J). The corresponding values of the independent variable are calculated using the same increments in axial distance, EX, to the left of the first axial station as those to the right. For example, the value of the independent variable used at the first dummy station to the left of axial station 1 would be, EX(1)- EX(2)-EX(1). A similar procedure is used for stations at the end of the machine. That is, for dummy stations placed after the last axial station, the values of the dependent variable are Y(NX,J). The corresponding values of the independent variable are calculated using the same increments to the right as those to the left of the last axial station whose index is less than I1 or greater than I2.

It is possible, in some cases, that portions of the Y versus X curve may be horizontal, that is, that Y is constant with X for a number of axial stations in the machine. This condition is most commonly encountered when the dependent variable is the streamline radius and a straight hub or tip is used for the inlet or outlet sections of the machine. Theoretically, the calculated coefficients of a least-squares parabola, fit through a group of points forming a horizontal line, should all be zero except the first which should be identical to the constant value of the dependent variable, Y. Due to the round-off error inherent in digital-computer calculations, it is highly probable that the parabola coefficients, evaluated by a computer, will all be nonzero. The use of the coefficients to calculate derivatives, which are known to be zero for a constant value of Y, will lead to inaccurate results. Therefore, a procedure is used to determine whether any portions of the Y versus X curve are horizontal and to set the values of DYDX and D2YDX2 to zero for axial stations falling within those portions of the curve.

Starting from the first axial station, I=1, the values of

$$\text{DELTY} = \frac{1}{\text{YAV}} \cdot \frac{Y(I+1,J)-Y(I,J)}{EX(I+1,J)-EX(I)}$$

where

$$\text{YAV} = \frac{Y(I+1,J)+Y(I)}{2.0}$$

are calculated and compared with a small number, chosen as 10^{-5} to allow for some round-off error in the calculations. From the former of the two equations above, it can be seen that DELTY is the fractional change in the dependent variable per unit axial distance between stations I and I+1. If the value of DELTY is less than 10^{-5}, the Y versus X curve between these two stations is, for all practical purposes, horizontal. Therefore, the values of DYDX(I,J) and D2YDX2(I,J) are set equal to zero. The procedure is continued until the station at which the curve is no longer horizontal is found. The last station from the inlet to the machine for which the curve is horizontal is indicated as K1 and represents the <u>first</u> station for which a least-squares parabolic-fitting procedure may be used to evaluate derivatives.

A similar checking procedure is used for axial stations, KNX, starting from the last station of the machine, KNX=NX, and calculating values of

$$\text{DELTY} = \frac{1}{\text{YAV}} \cdot \frac{Y(KNX,J)-Y(KNX-1,J)}{EX(KNX)-EX(KNX-1)}$$

where

$$\text{YAV} = \frac{(KNX,J)+Y(KNX-1,J)}{2.0}$$

For stations where DELTY 10^{-5}, the values of DYDX(KNX,J) and D2YDX2(KNX,J) are set equal to zero. The last value of KNX is the index of the last axial station from the end of the machine for which the curve is horizontal. The axial statation, KNX, is then the <u>last</u> one for which a least-squares parabola-fitting procedure may be <u>used</u> to evaluate derivatives.

The values of K1 and KNX are then compared with the values of I1 and I2 in order to determine which types of fitting procedure are required and for which axial stations each type of procedure is to be used.

When the derivatives for all axial stations from 2 to NX-1 have been calculated, K1 and KNX are compared with 1 and NX, respectively. If K1=1 the value of D2YDX2(1,J) is set equal to zero and the value of

$$DYDX(1,J) = \frac{Y(2,J)-Y(1,J)}{EX(2)-EX(1)}$$

is calculated. If KNX=NX, the value of D2YDX2(NX,J) is set equal to zero and the value of

$$DYDX(NX,J) = \frac{Y(NX,J)-Y(NX-1,J)}{EX(NX)-EX(NX-1)}$$

is calculated. The foregoing corresponds to specifying the boundary conditions at the inlet and outlet of the machine.

Finally, the streamline slope angles and curvatures are determined. For passes after the first, the curvature is set equal to the mean of the value used on the previous pass, and that prescribed by the streamline slopes and rates of change of slope just obtained.

SUBROUTINE LSQFIT

The function of Subroutine LSQFIT is to determine the coefficients of a second-order least-squares fitting polynomial for the streamlines. The subroutine is called by Subroutine LSQLNE. Step 16 of the STEPWISE ANALYSIS PROCEDURE involves the use of the subroutines. Card identifier: S08.

SUBROUTINE SEARCH

The function of Subroutine SEARCH is to estimate the flow required in a sector of the compressor to give the specified downstream static pressure. The subroutine is called by Subroutine DISTOR, and calls Function BIG. A flow diagram for the subroutine is shown on Figure 9. Step 95 of the STEPWISE ANALYSIS PROCEDURE is performed by Subroutine SEARCH. Card identifier: S09.

FUNCTION BIG

The function of Function BIG is to determine the value of the larger of two quantities. The function is called by Subroutine SEARCH, and is involved in Step 95 of the STEPWISE ANALYSIS PROCEDURES. Card identifier: S10.

FUNCTION SGIN

The function of Function SGIN is the return 1.0 or -1.0, in accord with the sign of the input quantity. The function is called by Subroutine SOLVE and Subroutine INTERP. Steps 62 and 82 involve the use of the function. Card identifier: S11.

SUBROUTINE RATE

The function of Subroutine RATE is to determine the radial derivatives of various quantities required for the solution to the momentum equation. The subroutine is called by Subroutine SOLVE. Step 71 of the STEPWISE ANALYSIS PROCEDURE involves the use of the subroutine. Card identifier: S12.

SUBROUTINE INTERP

The function of Subroutine INTERP is to determine the specification of the blade sections formed by intersection of the streamlines with the blade rows and hence to supervise the prediction of the blade section performances. The subroutine is called by Subroutine SOLVE, and calls Subroutine GRAPH3 and Subroutine CASCDE.

The following relates the subroutine to the STEPWISE ANALYSIS PROCEDURE:

Step Number	Cards (Identifier S13)
27	0017 through 0060
28	0061 through 0078
63	0080 through 0084
64	0085 and 0086

SUBROUTINE CASCDE

Subroutine CASCDE is the principal subroutine used to predict blade section performance. The subroutine is called by Subroutine INTERP, and calls Subroutine OPTANG, Subroutine STLANG, and Subroutine MCDEVN. A flow diagram for the subroutine is shown on Figure 10.

The following relates the subroutine to the STEPWISE ANALYSIS PROCEDURE:

Step Number	Cards (Identifier S14)
29	0021 through 0034
30	0035 through 0041
33	0045 through 0053

Step Number	Cards (Identifier S14)
34	0057 through 0073
41	0076 through 0091
42	0095
43	0097 through 0107
44	0110 through 0114
45	0115 through 0121
46	0123 through 0160
47	0161 through 0172
48	0173 through 0178
49	0180 through 0183
51	0186 through 0190
52	0193 and 0194
53	0196 through 0206
54	0208 through 0215
59	0217
60	0219 through 0232
61	0233 through 0241
62	0242 through 0312

SUBROUTINE GRAPH2

The function of Subroutine GRAPH2 is to interpolate between choking and stalling angles (determined by Subroutine STLANG) to find the values corresponding to the solidity of the cascade section under consideration. The subroutine is called by Subroutine STLANG. Step 36 of the STEPWISE ANALYSIS PROCEDURE is performed by Subroutine GRAPH2. Card identifier: S15.

SUBROUTINE STLANG

The principal function of Subroutine STLANG is to determine the choking and stalling inlet angles of the cascade section under consideration, by determining these quantities for the cascade of interest, but at solidities of 0.5, 1.0, and 1.5, and interpolating between these solidities (in Subroutine GRAPH2) for the desired solidity. Subroutine STLANG also initiates the determination of the relative outlet flow angle from the cascade (by Subroutine OUTANG). Subroutine CASCDE calls Subroutine STLANG. Step 35 of the STEPWISE ANALYSIS PROCEDURE is performed by the subroutine. Card identifier: S16.

SUBROUTINE OUTANG

The function of Subroutine OUTANG is to determine the relative outlet flow angle from a specified blade section. The subroutine is called by Subroutine STLANG.

The following relates the subroutine to the STEPWISE ANALYSIS PROCEDURE:

Step Number	Cards (Identifier S17)
37	0004 through 0011
38	0016 through 0020
39	0021 and 0022
40	0023 through 0058

SUBROUTINE OPTANG

The function of Subroutine OPTANG is to determine for a specified blade section the minimum loss inlet angle. The subroutine is called by Subroutine CASCDE. Step 32 of the STEPWISE ANALYSIS PROCEDURE is performed by the subroutine. Card identifier: S18.

SUBROUTINE MCDEVN

The function of Subroutine MCDEVN is to determine the increase in the minimum loss exit flow angle that occurs when the relative inlet Mach number exceeds the critical value.

The following relates the subroutine to the STEPWISE ANALYSIS PROCEDURE:

Step Number	Cards (Identifier S19)
55	0022
56	0023 through 0047
57	0051
58	0052 through 0058

SUBROUTINE BLTHIC

The function of Subroutine BLTHIC is to reestimate during each velocity profile iteration the annulus wall boundary-layer thickness at a computing station. The subroutine is called by Subroutine SOLVE.

The following relates the subroutines to the STEPWISE ANALYSIS PROCEDURE:

Step Number	Cards (Identifier S20)
85	0016 through 0026
86	0027 through 0060
87	0061

SUBROUTINE PRINT

The function of Subroutine PRINT is to output (onto the standard output unit normally) the velocity diagram results for each sector analysis. The output is programmed to print up to 60 lines on a page, and to skip to a new page whenever a column of results would be separated from its heading by the 60 line limit. The subroutine called by Subroutine DISTOR. Step 98 of the STEPWISE ANALYSIS PROCEDURE is performed by the routine. Card identifier: S21.

SUBROUTINE PUNCH

The function of Subroutine PUNCH is to output onto the "L-Series" units the results of each sector analysis, for use subsequently by Subroutine THETA. The subroutine called by Subroutine DISTOR. Step 99 of the STEPWISE ANALYSIS PROCEDURE is performed by the subroutine. Card identifier: S22.

SUBROUTINE THETA

The function of Subroutine THETA is to reestimate the angular coordinates of the streamlines defining the sector centerlines, the angle of inclination of these centerlines against the local meridional plane, the angular width of each sector, the circumferential gradients of static pressure and streamline slope, and the change in angular momentum along streamlines in blade-free spaces. The subroutine is called by the MAIN program, and calls Subroutine NEWRAP, Subroutine SMITH, Subroutine OVERAL, Subroutine FOUR, Subroutine GEORGE, Subroutine EDWIN, Function SERVUS, Function VBARF, Function DERIV, Function XMEAN, Function WBARF, and Function FINDY. (Calls to OVLTWO and OTHREE are calls to alternative entry points of Subroutine OVERAL.) Figure 11 shows a flow diagram for Subroutine THETA.

The principal inputs to Subroutine THETA are arrays giving for every meshpoint defined by a streamline number, station number, and sector number, the radius, static pressure, static temperature, streamline slope, meridional component of velocity, and absolute whirl angle.

The principal outputs are the quantities whose reestimation is the function of the subroutine indicated above.

The following relates the subroutine to the STEPWISE ANALYSIS PROCEDURE:

Step Number	Cards (Identifier S23)
102	0052 through 0062
103	0064 through 0081
104	0085 through 0089
105	0092 through 0101
106	0103 through 0106

Step Number	Cards (Identifier S23)
107	0107
108	0108 through 0113
109	0115 through 0117
110	0118 through 0121
111	0122 and 0123
112	0128 through 0146
114	0152 through 0162
116	0167 through 0185
117	0189 through 0202
118	0206 through 0217
119	0221 through 0223
120	0227 through 0231
121	0235 through 0253
122	0257 through 0266
123	0268
124	0270
125	0274 through 0282
126	0283 through 0293
127	0297 through 0306
132	0379 through 0386
134	0416 through 0427
135	0432 through 0452
136	0453 and 0454
137	0456 through 0479
138	0480 and 0481
139	0483 through 0486
140	0487 through 0499
141	0501 through 0532
142	0565 through 0583
143	0585
144	0588 through 0604
145	0606 through 0608
146	0611
147	0614 through 0629
148	0630
149	0632 through 0650

SUBROUTINE OVERAL

The function of Subroutines OVERAL is to determine (and print) the overall performance of the compressor, as implied by the results of the current cycle of calculations. The subroutine is called by Subroutine THETA. Note that the subroutine contains two ENTRY statements, so that calls to OVERAL, OVLTWO, and OTHREE are all calls to this subroutine. The following relates the subroutine to the STEPWISE ANALYSIS PROCEDURE:

Step Number	Cards (Identifier S24)
113	0020 through 0040
115	0041 through 0065
150	0066 through 0079

FUNCTION SERVUS

The function of Function SERVUS to determine the ratio of any sector width to the width of the same sector at the first computing station. Subroutine THETA calls the function, and the function calls Function DELTA. Function SERVUS is involved in Step 120 of the STEPWISE ANALYSIS PROCEDURE. Card identifier: S25.

FUNCTION FINDY

Function FINDY is designed to linearly interpolate (or extrapolate) at the reference radii the data read in by Subroutine THETA at the sector streamline radii. The function is called by Subroutine THETA and is involved in Step 116 of the STEPWISE ANALYSIS PROCEDURE. Card identifier: S26.

FUNCTION DELTA

The function of Function DELTA is to determine the angular separation of two sector centerlines. The function is called by Subroutine FOUR, Function SERVUS, and Function DERIV. The function is involved in Steps 120, 128, and 134 of the STEPWISE ANALYSIS PROCEDURE. Card identifier: S27.

FUNCTION DERIV

Function DERIV is used to determine the circumferential gradients of static pressure and streamline slope angle. The function is called by Subroutine THETA, and calls Function DELTA. Steps 128, 134, and 137 of the STEPWISE ANALYSIS PROCEDURE involve the use of the function. Card identifier: S28.

SUBROUTINE FOUR

The function of Subroutine FOUR is to determine the coefficients of a Fourier series representing a piece-wise linear function specified by the points at the intersections of the straight-line regions. Subroutine THETA calls Subroutine FOUR, which calls Function DELTA. Subroutine FOUR performs Step 128 of the STEPWISE ANALYSIS PROCEDURE. Card identifier: S29.

FUNCTION XMEAN

Function XMEAN is used to obtain the circumferential mean of quantities assumed to vary linearly between specified points. Subroutine THETA calls Function XMEAN, which calls Function DELTA. Steps 127 and 133 of the STEPWISE ANALYSIS PROCEDURE involve the use of Function XMEAN. Card identifier: S30.

SUBROUTINE EDWIN

The function of Subroutine EDWIN is to sum the Fourier series defining the static pressure and static pressure gradient in regions where the perturbation solution is obtained. The subroutine is called by Subroutine THETA, and calls Function VBARF, Function VBARDF, and Function WBARF. Subroutine EDWIN is involved in Step 133 of the STEPWISE ANALYSIS PROCEDURE. Card identifier: S31.

SUBROUTINE NEWRAP

The function of Subroutine NEWRAP is to determine the coefficients of a function approximating the (circumferential) mean meridional and tangential velocity variations in the axial direction. The subroutine is called by Subroutine THETA, and is involved in Step 129 of the STEPWISE ANALYSIS PROCEDURE. Card identifier: S32.

SUBROUTINE CPOUR

The function of Subroutine CPOUR is to control overflows generated by Subroutine CSIMEQ, which calls Subroutine CPOUR. Subroutine CPOUR is involved in Step 130 of the STEPWISE ANALYSIS PROCEDURE. Card identifier: S33.

FUNCTION VBARF

The function of VBARF is to calculate the mean tangential velocity at a particular axial location according to the function defined by Subroutine NEWRAP. Function VBARF is called by Subroutine THETA, Subroutine EDWIN, and Subroutine SMITH. Steps 137, 133, and 130 of the STEPWISE ANALYSIS PROCEDURE involve Function VBARF. Card identifier: S34.

FUNCTION VBARDF

Function VBARDF is used to determine the gradient in the axial direction of the mean tangential velocity according to the function defined by Subroutine NEWRAP. Function VBARDF is called by Subroutine EDWIN and Subroutine SMITH. Steps 130 and 133 of the STEPWISE ANALYSIS PROCEDURE involve Function VBARDF. Card identifier: S35.

FUNCTION WBARF

The function of Function WBARF is to calculate the mean axial velocity at a particular axial location according to the function defined by Subroutine NEWRAP. Function WBARF is called by Subroutine THETA, Subroutine EDWIN, and Subroutine SMITH. Steps 130, 131, and 133 of the STEPWISE ANALYSIS PROCEDURE involve Function WBARF. Card identifier: S36.

SUBROUTINE CSIMEQ

The function of Subroutine CSIMEQ is to solve a set of simultaneous linear algebraic equations having complex coefficients by means of a Gauss-Jordan triangular reduction. Rows are interchanged to maximize the diagonal elements during the reduction. Since these elements are used as divisors, errors are thereby minimized. Also, a zero check is used to bypass unnecessary operations. Subroutine CSIMEQ is called by Subroutine SMITH, and calls Subroutine CPOUR. Step 130 of the STEPWISE ANALYSIS PROCEDURE involves Subroutine CSIMEQ. Card identifier: S37.

SUBROUTINE GEORGE

The function of Subroutine GEORGE is to sum the Fourier series defining the perturbation tangential and axial velocities in regions where the perturbation solution is obtained. The subroutine is called by Subroutine THETA. Step 131 involves Subroutine GEORGE. Card identifier: S38.

SUBROUTINE SMITH

The function of Subroutine SMITH is to compute the Fourier coefficients which define the perturbation solution to the flow in regions where this is obtained, that is, the compressor inlet and outlet ducts. Subroutine SMITH is called Subroutine THETA, and calls Subroutine CSIMEQ. Function VBARF, Function VBARDF, and Function WBARF. Step 130 of the STEPWISE ANALYSIS PROCEDURE is performed by Subroutine SMITH. Card identifier: S39.

SUBROUTINE PCTURE

Subroutine PCTURE supervises the plotting of the sector centerline locations and conditions. Subroutine PCTURE is called by the MAIN Program, and calls Subroutine PLOT and Subroutine MPLOT. Step 151 of the STEPWISE ANALYSIS PROCEDURE involves Subroutine PCTURE. Card identifier: S40.

SUBROUTINE PLOT

Subroutine PLOT creates polar plots showing the location of each sector centerline at every computing station. The subroutine is called by Subroutine PCTURE. Step 151 of the STEPWISE ANALYSIS PROCEDURE involves Subroutine PLOT. Card identifier: S41.

SUBROUTINE MPLOT

Subroutine MPLOT creates Cartesian plots showing the variations of conditions with radius along sector centerlines at each computing station. The subroutine is called by Subroutine PCTURE. Step 151 of the STEPWISE ANALYSIS PROCEDURE involves Subroutine MPLOT. Card identifier: S42.

NOMENCLATURE OF FORTRAN

The FORTRAN variables are listed according to the COMMON block containing them, or, if they are not in a COMMON block, by the name of subroutine in which they occur. (All COMMON is named.)

COMMON/DATA/

Fortran Symbol	Use
CPI	Specific heat of working fluid, Btu per lb deg R
GASK	Gas constant of working fluid, ft lb per lb deg R
HUBLOC	Fraction of annulus blockage area initially placed on hub
IFBL	Indicator specifying annulus wall boundary-layer calculation option
IFRPM	Indicator controlling output during first cycle
IFSIMP	Indicator controlling equilibrium and curve-fitting options
IFTYPE	Indicator specifying station lean angle and solution type option
IGV	Set by program to 1 if number of sector is greater than 1, otherwise set to 2

Fortran Symbol	Use
IMASS	Set to 1 in program
INCPO	Increment in pass number for printing of cascade analysis results
INCWRI	Increment in pass number for printing of velocity triangle results
IPUNCH	Set to zero in program
IRAD	Set to zero in program for first cycle, 1 for subsequent cycles
ITAPE	Current input unit numbers from "J-Series"
ITER	Indicator controlling velocity print-out option
NBETA	Number of blading data radii
NLINES	Number of streamlines on each sector centerline
NPASS	Maximum number of passes per cycle
NPLOT	First pass during which cascade analysis results are printed
NPTS	Indicator controlling output from Subroutine PUNCH
NTAPE	Standard output unit number
NTOPO	Number of radii at which inlet conditions are specified
NWRIT	First pass during which velocity triangle results are printed
NX	Total number of computing stations
TOLCX	Tolerance for continuity calculation at each station, as a fraction of flow
VIS1	First coefficient in expression = VIS1+ VIS2 to give viscosity for Reynolds number calculations, lb per ft sec

Fortran Symbol	Use
VIS2	See above, lb per ft sec deg R
ANGLN (I)	Station I lean angle, deg
BBP(I,J)	Blading data radii, Station I, data line J, ft
BETA	Blade camber angle, or outlet angle, deg
BLAK(I,J)	Blade thickness/chord ratio, Station I, data line J
BLAM(I)	Fraction of annulus blocked by boundary layer at Station I
CLOSS(I,J)	Blade row total pressure loss coefficients, Station I, data line J
COMENT(K)	Title for sector analysis
CO(I,J)	Additional loss factor, Station I, data line J
DELM(J)	Fraction of flow between hub and streamline J
ESTFLO	Estimated flow in sector, on whole compressor basis, lbs per sec
EXPLOS(I)	Exponent in loss/incidence relationship, Station I
FLOWI	Specified downstream static pressure, lbs per sq ft
IBETA2(I)	Indicator specifying cascade analysis option, blade preceding Station I
IFCAX(I)	Indicator specifying cascade analysis option
IFLVS(I)	Indicator specifying cascade analysis option
IFMACH(I)	Indicator specifying cascade analysis option
IFMLOS(I)	Indicator specifying cascade analysis option
IFPROF(I)	Indicator specifying cascade analysis option
IFREYL(I)	Indicator specifying cascade analysis option
IFREYN(I)	Indicator specifying cascade analysis option

Fortran Symbol	Use
IFTHIC(I)	Indicator specifying cascade analysis option
ILOSS(I)	Indicator specifying cascade analysis option
IMACHI(I)	Indicator specifying solution type
KSUMRY	Set to 1 in program
RPM2(I)	Speed of rotation of blade row preceding Station I, revolutions per min
RHP(I)	Hub radius, Station I, ft
RSP(I)	Casing radius, Station I, ft
ROT	Not used
SIGMA(I,J)	Cascade solidity, Station I, data line J
STAG(I,J)	Blade stagger angle, or inlet angle, Station I, data line J, deg
WIDTH	Estimated flow range of characteristic, as a fraction of flow
X(I)	Axial coordinate of intersection of computing Station I with axis, ft
XMACH(I)	Multiplying factor in Mach number deviation, correction, blade preceding Station I
XBAR(I,J)	Distance to maximum camber point of blade from leading edge, as fraction of chord, Station I, data line J
DELR(J)	Fraction of annulus height between hub and streamline J in inlet

COMMON /CALC/

Fortran Symbol	Use
FLOW	Current estimate of flow in sector, lbs per sec
ICASE	Set to 1 in program
KRPM	Not used

Fortran Symbol	Use
L	Current pass number
NCASE	Set to 1 in program
NDIV	Indicator recording convergence of sector analysis
NDUM	Not used
NMASS	Set to 1 in program
NPLET	Records NPLOT
NROT	Set to 1 in program
NWRET	Records NWRIT
ALPHA(I,J)	Streamline slope angle, Station I, streamline J, radians
CMT(I,J)	Meridional velocity, Station I, streamline J, ft per sec
PO(I,J)	Total pressure, Station I, streamline J, lbs per sq ft
RCB(I,J)	Curvature of streamline, Station I, streamline J, 1/ft
STATT(I,J)	Static temperature, Station I, streamline J, deg R
TO(I,J)	Total temperature, Station I, streamline J, deg R
WMT(I,J)	Meridional velocity on previous pass, ft per sec, also rate of change of slope of streamlines, Station I, streamline J

COMMON /WENT/

Fortran Symbol	Use
GJ	64.35 x 778.0, being 2.g.J, ft^2 lbs per sec^2 BTU
GJCP	Not used
GR2	64.35 x GASK, being 2.g.R, ft^2 per sec^2 deg R

Fortran Symbol	Use
CXM(I,J)	Tangent of flow angle defined as tangential velocity/meridional velocity, Station I, streamline J
Y(K)	Various cascade analysis parameters

COMMON /NEW/

Fortran Symbol	Use
RDIS(I,J)	Asymmetric flow parameter data radii, Station I, data line J, ft
XMU(I,J)	Angle μ, Station I, data line J, radians
DPDTH(I,J)	$\frac{1}{\rho}\frac{\partial p}{\partial \theta}$, Station I, data line J, ft^2/sec^1
DFDTH(I,J)	$\frac{\partial \phi}{\partial \theta}$, Station I, data line J
ARATIO(I,J)	Sector angular width/sector inlet angular width, Station I, data line J

COMMON /RESULT/

Fortran Symbol	Use
DAMP	Relaxation factor for streamline location re-estimation
R(I,J)	Radius of streamline location, Station I, streamline J, ft

COMMON /TOGOSS/

Fortran Symbol	Use
BET1(I,J)	Relative inlet angle to blade row, Station I, streamline J, deg
CLOS(I,J)	Blade row total pressure loss coefficient, Station I, streamline J
CR(I,J)	Not used
CUPONS(I,J)	Cascade solidity, blade preceding, Station I, streamline J

Fortran Symbol	Use
RM1(I,J)	Relative inlet Mach number, blade following Station I, streamline J
STAGG(I,J)	Minimum loss inlet angle, blade row following Station I, streamline J, deg
STATP(I,J)	Static pressure, Station I, streamline J, lbs per sq ft
TANBET(I,J)	Tangent of relative outlet flow angle, blade row preceding Station I, streamline J

COMMON /GEN/

Fortran Symbol	Use
TITLE(K)	Case identification, alphanumeric
NTHETA	Number of sectors
LP	Counter on major iteration loop (cycle number of job)
NLOOPS	Total number of cycles
IFPRIN	Indicator controlling printout for DISTORT
R32(N)	Angular coordinate of sector centerline, deg, radians

COMMON /SECTOR/

Fortran Symbol	Use
IFFULL	Indicator governing reading of input
EF(N)	Flow in sector N, lbs per sec
CARD(K,N)	Sector identification, alphanumeric
I1	Total number of axial computing stations (NX in /DATA/)
I2	NLINES in /DATA/
I3	NBETA in /DATA/

Fortran Symbol	Use
I4	NTOPO in /DATA/
I5	IMASS in /DATA/
I6	IFSIMP in /DATA/
I7	NPASS in /DATA/
I8	IFBL in /DATA/
I9	NPTS in /DATA/
I10	IGV in /DATA/
I11	IFRAD in /DATA/
I12	IFRPM in /DATA/
I13	ITER in /DATA/
I14	Set to 1 in program
I15	NPLOT in /DATA/
I16	INCPO in /DATA/
I17	NWRIT in /DATA/
I18	INCWRI in /DATA/
I19	IRAD in /DATA/
I20	IPUNCH in /DATA/
I21	IFTYPE in /DATA/
I22	KSUMRY in /DATA/
R1	TOLCX in /DATA/
R2	HUBLOC in /DATA/
R3	GASK in /DATA/
R4	VIS1 in /DATA/
R5	VIS2 in /DATA/
R6	CP1 in /DATA/

Fortran Symbol	Use
I23(I)	IBETA2 in /DATA/
I24(I)	IFTHIC in /DATA/
I25(I)	IFCAX in /DATA/
I26(I)	IFMACH in /DATA/
I27(I)	IFREYN in /DATA/
I28(I)	ILOSS in /DATA/
I29(I)	IFMLOS in /DATA/
I30(I)	IFLVSI in /DATA/
I31(I)	IFPROF in /DATA/
I32(I)	IFREYL in /DATA/
R13(I,J)	BBP in /DATA/
R14(I,J)	STAG in /DATA/
R15(I,J)	BETA in /DATA/
R16(I,J)	SIGMA in /DATA/
R17(I,J)	BLAK in /DATA/
R18(I,J)	ZBAR in /DATA/
R19(I,J)	CLOSS in /DATA/
R20(I,J)	CO in /DATA/
R21(J)	DELR in /DATA/
R22(I)	BLAM in /DATA/
R23	FLOWI in /DATA/
R24	ESTFLO in /DATA/
R25	WIDTH in /DATA/
R26(I)	X in /DATA/
R27(I)	RHP in /DATA/

Fortran Symbol	Use
R28(I)	RSP in /DATA/
R29(I)	RPM2 in /DATA/
I33(I)	IMACHI in /DATA/
R30(I)	ANGLN in /DATA/
EXPLOS(I)	EXPLOS in /DATA/
XMACH(I)	XMACH in /DATA/

COMMON /ARRAY/

Fortran Symbol	Use
RADIN(J,N,I)	Streamline radius, streamline J, sector N, Station I, ft
PIN(J,N,I)	Static pressure, streamline J, sector N, Station I, lbs/ft^2
TIN(J,N,I)	Static temperature, streamline J, sector N, Station I, deg R
VMIN(J,N,I)	Meridional velocity, streamline J, sector N, Station I, ft/s
TANAIN(J,N,I)	Tangent of absolute gas angle, streamline J, sector N, Station I
PHIN(J,N,I)	Slope angle of streamline, streamline J, sector N, Station I

COMMON /AUXIL/

Fortran Symbol	Use
RREF(J,I)	Circumferential average of radii, streamline J, Station I, ft
ZREF(J,I)	Axial distance corresponding to RREF(J,I), ft
THETAS(N,I)	Damped θ, sector N, Station I, radian
THETAP(N,I)	Damped θ of previous cycle (LP-1), sector N, Station I, radian

Fortran Symbol	Use
SEARAT(N,I)	Sector width ratio, sector N, Station I
PREV(N,I)	Unmodified α or $\frac{1}{\rho}\cdot\frac{\partial p}{\partial \theta}$, sector N, Station I
DPDTHS(N,I)	$\frac{1}{\rho}\cdot\frac{\partial p}{\partial \theta}$ after mixing, sector N, Station I, ft^2/s^2
DRCTHS(N,I)	$\int_{I-1}^{I}\frac{1}{\rho}\cdot\frac{1}{C_m}\cdot\frac{\partial p}{\partial \theta}\,dm$, sector N, Station I, ft^2/s
PTOTAL(N)	$\frac{P_{TOTAL}}{\rho}$, total pressure, divided by density, sector N, ft^2/s^2
PSTAT(N)	$\frac{P_{STATIC}}{\rho}$, static pressure, divided by density, sector N, ft^2/s^2
RADUM(J)	Dummy array for radius storage, sector N
PDUM(J)	Dummy array for pressure storage, sector N
TDUM(J)	Dummy array for temperature storage, sector N
VMDUM(J)	Dummy array for meridional velocity storage, sector N
PHIDUM(J)	Dummy array for storage of streamline slope angle, sector N
TANDUM(J)	Dummy array for tangent of absolute air angle, sector N
VS(N)	Circumferential velocity (perturbation), sector N, ft/s
WS(N)	Meridional velocity (perturbation), sector N, ft/s
DPREV(N,I)	Static pressure gradient $\frac{1}{\rho}\cdot\frac{\partial p}{\partial \theta}$ of previous cycle (LP-1) sector N, Station I, ft^2/s^2

COMMON /ONE/

Fortran Symbol	Use
NF	Number of Fourier terms
CNZ(M)	Fourier sine coefficient for total pressure at z = XLI

Fortran Symbol	Use
DNZ(M)	Fourier cos coefficient for total pressure at z = XL1
XL1	Nondimensional axial location of total pressure boundary condition
CNF(M)	Fourier sin coefficient for either static pressure or flow angle at z = XL2
DNF(M)	Fourier cos coefficient for either static pressure or flow angle at z = XL2
XL2	Nondimensional axial location of second boundary condition
IND2	If 0, boundary condition at z = XL2 gives static pressure; if 1, flow angle
ENF(M)	Fourier sin coefficient for either static pressure or flow angle at z = XL3
FNF(M)	Fourier cos coefficient for either static pressure or flow angle at z = XL3
XL3	Nondimensional axial location of third boundary condition
IND3	If 0, boundary condition at z = XL3 gives static pressure, if 1, flow angle
WZ	Wo in mean axial velocity equation: $W = W_0 + \hat{W}_1 e^{k_1 z} + \hat{W}_2 e^{k_2 z}$
VZ	Vo in mean tangential velocity equation: $V = V_0 + \hat{V}_1 e^{k_1 z} + \hat{V}_2 e^{k_2 z}$
WCAP1	$\hat{W}_1$ in mean axial velocity equation
VCAP1	$\hat{V}_1$ in mean tangential velocity equation
XK1	k_1 in mean velocity equations
WCAP2	$\hat{W}_2$ in mean axial velocity equation
VCAP2	$\hat{V}_2$ in mean tangential velocity equation
XK2	k_2 in mean velocity equations
SZ	Flow angle tangent: $S_0 = \frac{V_z}{W_z}$

Fortran Symbol	Use
EP	$\frac{1}{W_z}$
VMQ	$\sqrt{-1} = j$
AMQ(M)	Coefficient of $e^{mz} \cdot e^{jmy}$ in perturbation stream function solution
BMQ(M)	Coefficient of $e^{-jmS_0 z} \cdot e^{jmy}$ in perturbation stream function solution
CMQ(M)	Coefficient of $e^{-mz} \cdot e^{jmy}$ in perturbation stream function solution
DMQ(M,I)	Coefficient of $e^{(k_i + m)z} \cdot e^{jmy}$ in perturbation stream function solution, $i = 1$ or 2
EM2(M,I)	Coefficient of $e^{(k_i - jmS_0)z} \cdot e^{jmy}$ in perturbation stream function solution, $i = 1$ or 2
FMQ(N,K)	Coefficient of $e^{(k_i - m)z} \cdot e^{jmy}$ in perturbation stream function solution, $i = 1$ or 2

COMMON /OV/

Fortran Symbol	Use
IZR1	First relative axial station number of current tape
IZR2	Last relative axial station number of current tape
IZA1	First absolute station number of current tape
GUMBO	Exponent
LOG1	Standard output unit number

COMMON /HUGE/

Fortran Symbol	Use
P(J,N,I)	Static pressure (except at Station 1, total pressure), streamline J, sector N, Station I, lbs per sq ft; also θ, deg

Fortran Symbol	Use
T(J,N,I)	Static temperature (except at Station I, total temperature), streamline J, sector N, Station I, deg R
VM(J,N,I)	Meridional velocity, streamline J, sector N, station I, ft per sec
TANALP(J,N,I)	Tangent of flow angle defined as tangential velocity/meridional velocity, streamline J, sector N, Station I
RAD(J,I)	Radius of streamlines, last sector, streamline J, Station I, ft

MAIN PROGRAM

Fortran Symbol	Use
R32(N)	Angular location of centerline of sector N in inlet, deg
TITLE(K)	Title for job
HEAD(K)	Variable heading NEW CASE or CONTINUATION
J1	Number of last station for which data is on units JTAPE1 and LTAPE1
J21	Number of first station for which data is on units JTAPE2 and LTAPE2
J22	Number of last station for which data is on units JTAPE2 and LTAPE2
J31	Number of first station for which data is on units JTAPE3 and LTAPE3
J32	Number of last station for which data is on units JTAPE3 and LTAPE3
J41	Number of first station for which data is on units JTAPE4 and LTAPE4
J42	Number of last station for which data is on units JTAPE4 and LTAPE4
J5	Number of first station for which data is on units JTAPE5 and LTAPE5

Fortran Symbol	Use
ITAPE	Standard input unit number
NTAPE	Standard output unit number
JTAPE1	Intermediate storage unit number
JTAPE2	Intermediate storage unit number
JTAPE3	Intermediate storage unit number
JTAPE4	Intermediate storage unit number
JTAPE5	Intermediate storage unit number
LTAPE1	Intermediate storage unit number
LTAPE2	Intermediate storage unit number
LTAPE3	Intermediate storage unit number
LTAPE4	Intermediate storage unit number
LTAPE5	Intermediate storage unit number
KTAPE1	Intermediate storage unit number
KTAPE2	Intermediate storage unit number
K1	KTAPE1 on odd cycles, KTAPE2 on even cycles
K2	KTAPE2 on odd cycles, KTAPE1 on even cycles
NTHETA	Number of sectors
NLOOPS	Number of cycles of computation
IFSTAR	Indicator specifying job to be NEW CASE or CONTINUATION
IFPRIN	Indicator controlling output on cycles after the first
RFAC1	Static pressure gradient relaxation factor
RFAC2	Streamline angular relocation relaxation factor

Fortran Symbol	Use
RFAC3	Increment by which perturbation cycle is discarded per cycle
RFAC4	Not used
IFAC1	First cycle for which perturbation solution is eliminated by amount RFAC3
IFAC2	Number of cycles of perturbation
LP	Current cycle number
LDNM	Current sector number
KKK	Stores KI during unit switching

SUBROUTINE INPUT

Fortran Symbol	Use
RPM2(I)	See /DATA/
IBETA2(I)	See /DATA/
IFTHIC(I)	See /DATA/
IFCAX(I)	See /DATA/
IFMACH(I)	See /DATA/
IFREYN(I)	See /DATA/
ILOSS(I)	See /DATA/
IFMLOS(I)	See /DATA/
IFLVSI(I)	See /DATA/
IFPROF(I)	See /DATA/
IFREYL(I)	See /DATA/
EXPLOS(I)	See /DATA/
XMACH(I)	See /DATA/
BBP(I,J)	See /DATA/
STAG(I,J)	See /DATA/

Fortran Symbol	Use
BETA(I,J)	See /DATA/
SIGMA(I,J)	See /DATA/
BLAK(I,J)	See /DATA/
ZBAR(I,J)	See /DATA/
CLOSS(I,J)	See /DATA/
CO(I,J)	See /DATA/
DELM(J)	DELR in /DATA/
BLAM(I)	See /DATA/
X(I)	See /DATA/
RHP(I)	See /DATA/
RSP(I)	See /DATA/
IMACHI(I)	See /DATA/
ANGLN(I)	See /DATA/
COMENT(K,N)	Titles for N sectors
BHP(J,N)	J inlet condition data radii for sector N, ft
TOCO(J,N)	Inlet total temperature, data line J, sector N, deg R
POCO(J,N)	Inlet total pressure, data line J, sector N, lb per sq ft
AFCO(J,N)	Inlet whirl angle, data line J, sector N, deg
NX	See /DATA/
NLINES	See /DATA/
NBETA	See /DATA/
NTOPO	See /DATA/
IFSIMP	See /DATA/

Fortran Symbol	Use
NPASS	See /DATA/
IFBL	See /DATA/
IFRPM	See /DATA/
ITER	See /DATA/
NPLOT	See /DATA/
INCPO	See /DATA/
NWRIT	See /DATA/
INCWRI	See /DATA/
IFTYPE	See /DATA/
TOLCX	See /DATA/
HUBLOC	See /DATA/
GASK	See /DATA/
CP	See /DATA/
VIS1	See /DATA/
VIS2	See /DATA/
I	Station number
IB	IBETA2 at Station I, plus 1
IL	ILOSS at Station I, plus 1
FLOWI	See /DATA/
ESTFLO	See /DATA/
WIDTH	See /DATA/
N	Sector number
J1	See MAIN Program
J21	See MAIN Program
J22	See MAIN Program

Fortran Symbol	Use
J31	See MAIN Program
J32	See MAIN Program
J41	See MAIN Program
J42	See MAIN Program
J5	See MAIN Program
II	Used in computing J22
ICUS1	Set to 1 in program
ICUS2	Set to 0 for sector 1, 1 for subsequent sectors. Becomes NPTS in /DATA/, IFFULL in /SECTOR/
ICUS3	Set to 2 in program
ICUS4	Set to 0 in program
ICUS5	Set to 1 if number of sectors is greater than 1, otherwise set to 2. Becomes IGV in /DATA/
NUNIT	Current unit number from "J-Series"
II1	Current first station number for NUNIT
II2	Current last station number for NUNIT
II3	Assign location for transfer of control during writing of "J-Series" units
RCUS1	Set to 0.0, 1.0, 2.0, ... in program. Becomes RDIS in /NEW/
RCUS2	Set to 1.0 in program
RCUS3	Set to 0.0 in program

SUBROUTINE DISTOR

Fortran Symbol	Use
BLOCK(I)	Computed blockage at Station I, as a fraction of annulus area

Fortran Symbol	Use
ISLOPE	Indicator used in logic of Subroutine SEARCH
WPREV	Estimate of flow in sector made previously, lbs per sec
LL	Indicator used in logic of Subroutine SEARCH
IABORT	Indicator set by Subroutine READIN if certain data errors are detected
ISOLUT	Indicator used in logic of Subroutine SEARCH
NP	Maximum possible number of entries to Subroutine SEARCH
K	Index running from 1 to NP
KK	Pass number corresponding to K
PCALC	Latest calculated mean exit static pressure, lbs per sq ft
I	Station number

SUBROUTINE READIN

Fortran Symbol	Use
IFSUM	Set to 1 in program
KSUMAX	Set to 1 in program
I	Station number
J	Streamline or data line number
II1	First station for which data is on current unit ITAPE
II2	Last station for which data is on current unit ITAPE
LVS	IFLVSI at current station
IFMA	IFMACH at current station
NUNIT	Number of unit to be read next
N	Index, held to 1 in program

Fortran Symbol	Use
NPOINT	Indicator to select streamline curve fit
AA	Constant used in streamline radial relocation relaxation factor calculation
NZM1	NX-1
ASPECT	Square of aspect ratio of annulus between computing stations
DAMP2	Interim value of streamline relocation relaxation factor
RPM	Set to 0.0 in program
PRATIO	Set to 0.0 in program
ILOS	ILOSS at current station
IBET	IBETA2 at current station
GR	64.35 x GASK, being 2.g.R, ft^2 per sec^2 deg R
NTUB	NLINES - 1
SBLOC	1.0 - HUBLOC

SUBROUTINE SOLVE

Fortran Symbol	Use
F1(J)	$(1 + \tan^2\mu + \tan^2\gamma)^{-\frac{1}{2}}$, at streamline J
F2(J)	$\bar{A}\cos\phi - \bar{C}\sin\phi$, at streamline J
F3(J)	$(\bar{A}\sin\phi + \bar{C}\cos\phi)\frac{dC_m}{ds}$, at streamline J
F4(J)	$\frac{\bar{B}}{\rho r}\cdot\frac{\partial p}{\partial\theta}$ at streamline J
DP(J)	Interpolated value of $\frac{1}{\rho}\cdot\frac{\partial p}{\partial\theta}$ at streamline J
DF(J)	Interpolated value of $\frac{\partial\phi}{\partial\theta}$ at streamline J
XM(J)	Interpolated value of μ at streamline J

Fortran Symbol	Use
XINT(J)	Interpolated value of $\int_{I-1}^{I} \frac{1}{\rho}\frac{\partial p}{\partial \theta}\frac{1}{C_m}\,dm$ at streamline J
CONFAC(J)	Interpolated value of ARATIO midway between streamlines J and (J+1)
RADFAC(K)	Not used
U1VW1(J)	Product of blade speed and tangential velocity at blade unit, streamline J, ft^2 per sec^2
AFA(J)	Tangent of flow angle defined as tangential velocity/meridional velocity, streamline J
CPUP(J)	Specific heat, streamline J, BTU per lb deg R
CPDN(J)	Specific heat, streamline J, BTU per lb deg R
CPMN(J)	Specific heat, streamline J, BTU per lb deg R
WATE(J)	Specific weight, at streamline J, lbs per ft^3
RM(J)	Mach number controlling solution type at streamline J
W(J)	Flow between hub and streamline J, lbs per sec
P2RP2D(J)	Ratio of actual to ideal total pressure at streamline J
DPDR(J)	$\frac{\partial P}{\partial r}$ at streamline J, lbs per ft^3
DTDR(J)	$\frac{\partial T}{\partial r}$ at streamline J, deg R per ft
DTBDR(J)	$\frac{\partial}{\partial r}(\tan\beta)$ at streamline J, 1 per ft
RX(J)	Streamline radius at streamline J, ft
ARG(J)	Dummy array used with Subroutine RATE
AX(J)	$\left\{-(\bar{A}\cos\phi - \bar{C}\sin\phi)\left(\frac{1}{r_m} + \frac{\tan\alpha}{r}\cdot\frac{\partial\phi}{\partial\theta}\right) + A\left(\tan\alpha\,\frac{d\tan\alpha}{dr} + \frac{\tan^2\alpha}{r}\right) - (\bar{A}\sin\phi + \bar{C}\cos\phi)\frac{dC_m}{ds}\cdot\frac{\cos\alpha}{C_m}\right\}\cos^2\alpha$ at streamline J

Fortran Symbol	Use
BX(J)	$\left\{\bar{A}\left(gJc_p\frac{\partial T}{\partial r} - \frac{t}{T}\left(gJc_p\frac{\partial T}{\partial r} - \frac{1}{\rho t}\frac{\partial P}{\partial r}\right)\right) - \frac{\tan\mu}{\rho r}\bar{A}\frac{\partial p}{\partial \theta}\right\}$ at streamline J
DX(J)	Calculated meridional velocity, at streamline J, ft per sec
VELFAC(K)	Not used
RCAL(J)	Calculated radius of streamline J, ft
AFEPH(J)	Not used
TIRTI(J)	Ratio of relative total to static temperature at blade inlet streamline J
TANA(J)	$\frac{\tan\beta}{\left(\frac{1}{VR}+1\right)}$
U2(J)	Blade outlet speed at streamline J, ft per sec
VFAC(J)	Velocity ratio correction factor for blade at streamline J
J	Streamline number
BHPI	Radius of streamline, for Subroutine GRAPH3, ft
DM	Meridional velocity ratio, also $\frac{ds}{dm}$ for $\frac{\partial C_m}{\partial s}$ calculation
K	Indicator to record if Subroutine SEARCH will be entered on current pass
NP	Maximum number of entries to Subroutine SEARCH
KK	Pass number corresponding to current entry number to Subroutine SEARCH
NPOINT	Number of points for least-squares fitting of streamlines
I	Station number
NTUB	NLINES - 1
JM	NLINES/2
JLINE	NLINES/2 + 1, the starting streamline for the momentum equation integration

Fortran Symbol	Use
VEL	First pass estimate of meridional velocity in inlet, ft per sec
IFTAN	Indicator specifying option to perform axial velocity ratio correction to deflection for blade row
IBE	IBETA2 for blade row
RMID	Mean of two adjacent streamline radii, ft, also $\frac{ds}{dm}$ for $\frac{dC_m}{ds}$ calculation
GAMMA	Ratio of specific heats, γ
GF	$\frac{\gamma - 1}{2}$
UI	Blade row inlet speed, ft per sec
VF	Velocity ratio correction factor
SAMUP	$\frac{\gamma}{\gamma - 1}$
NITER	Maximum number of iterations at each station
X4	Tangent of station lean angle, also for momentum equation integration, also $\frac{1 - e^{A dl}}{A}$ for momentum equation integration
CBAR	$\bar{C} = \frac{\tan \mu}{\sqrt{1 + \tan^2 \gamma + \tan^2 \mu}}$
X1	Used in calculating $\frac{dC_m}{ds}$
X2	Used in calculating $\frac{dC_m}{ds}$
X3	Used in calculating $\frac{dC_m}{ds}$
IISOL	Records solution validity during iteration at a station
VLAST	Value of meridional velocity computed on hub during previous iteration
TRAT	Ratio of static to total temperatures
JJ	Streamline number during reestimation of streamline radii

Fortran Symbol	Use
AFF	Tangent of flow angle defined as tangential velocity/meridional velocity
WR	Flow-weighted mean Mach number
WR2	WR squared
RATIO	Number by which meridional velocities are multiplied to give estimate for next iteration
WRX	$1-WR^2$, limited to not less than 0.1 (absolute value)
WRAT	Ratio of current to previous meridional velocity estimate on hub
WMXRAT	Ratio of maximum possible to specified flows

SUBROUTINE GRAPH3

Fortran Symbol	Use
XIN	Independant variable
XOUT	Dependant variable
X(I,J)	Table of independant variables
Y(I,J)	Table of dependant variables
NPOINT	Number of values in above table (J dimention)
I	Station number to be used in table searching
K	Dummy table number used in searching procedure

SUBROUTINE LSQLNE

Fortran Symbol	Use
EY(I)	Axial coordinate of intersection of computing, Station I with axis, ft
Y(I,J)	Streamline radius, Station I, streamline J, ft

Fortran Symbol	Use
JAY	Streamline number
NPTS	Number of points for least squares fitting procedure
NEX	Total number of computing stations
DYDX(I,J)	Tangent of streamline slope angle, also streamline slope angle, radians, Station I, streamline J
D2YDX2(I,J)	Rate of change of streamline slope, also streamline curvature, 1/ft, Station I, streamline J
ANGLN(I)	Lean angle of Station I, deg
RCB(I,J)	Curvature of streamlines at Station I, streamline J, 1/ft
LL	Pass number
EX(I)	Axial coordinate of intersection of computing Station I with streamline under consideration, ft
U(K)	Selected streamline radii for curve-fitting, ft
V(K)	Selected axial coordinates for curve-fitting, ft
ACO(K)	Coefficients of second order polynomial giving least squares fit of streamline
J	Streamline number
NX	Total number of computing stations
I	Station number
NX1	NX - 1
I1	Integer relating number of points for fitting procedure to calculation
I2	Integer relating number of points for fitting procedure to calculation

Fortran Symbol	Use
MID	I1
MID1	I1-1
YAV	Average streamline radius between two computing stations, ft
DELTY	Change in streamline radius between two computing stations, normalized with respect to average streamline radius and distance between stations
L	Station number
B	2 times third polynomial coefficient

SUBROUTINE LSQFIT

Fortran Symbol	Use
Y(I)	Selected streamline radii for curve-fitting, ft
X(I)	Selected axial coordinates for curve-fitting, ft
NPTS	Number of points for least-squares curve-fitting procedure
A(K)	Coefficients of second-order polynomial procedure by curve-fitting procedure
N	NPTS

SUBROUTINE SEARCH

Fortran Symbol	Use
ISOLUT	Indicator recording validity of solution previously obtained in Subroutine SOLVE
ISLOPE	Indicator recording possibility of finding a satisfactory exit static pressure
LL	Indicator recording occurrance of previous valid solutions
WPREV	Previous estimate of sector flow, lbs per sec

Fortran Symbol	Use
PCALC	Latest calculated downstream static pressure, lbs per sq ft
STEP	Increment by which flow is incremented in search for valid solution, lbs per sec
PPREV	Previously calculated downstream static pressure, lbs per sq ft
NTUB	NLINES - 1
SLOPE	Slope of characteristic defined by two latest valid solutions
WTEMP	Temporary storage for sector flow, lbs per sec
WX	Temporary storage for sector flow, lbs per sec

FUNCTION BIG

Fortran Symbol	Use
X	Quantity examined for size relative to Y
Y	Quantity examined for size relative to X

FUNCTION SGIN

Fortran Symbol	Use
X	Quantity whose sign is examined

SUBROUTINE RATE

Fortran Symbol	Use
X(I)	Radii at which dependent variable is specified, ft
Y(I)	Dependent variable
IS	Number of first streamline at which gradient is required
IE	Number of last streamline at which gradient is required

Fortran Symbol	Use
DYDX(I)	Gradient of dependent variable at streamline I
I	Streamline number
DX1	Finite difference increment of independent variable increment
DY1	Finite difference increment of dependent variable
DXIE	Finite difference increment of independent variable
DYIE	Finite difference increment of dependent variable

SUBROUTINE INTERP

Fortran Symbol	Use
I	Number of computating station at blade row trailing edge
J	Streamline number of blade section under examination
VF	Number used to correct axial velocity ratio
BHP1	Streamline radius at blade outlet, ft
BHP2	Streamline radius at blade inlet, ft
BHP3	Mean streamline radius, ft
COSA1	Cosine of streamline slope angle, blade outlet
COSA2	Cosine of streamline slope angle, blade inlet
IBE	IBETA2 for blade row
TEMP1	Interpolated blade stagger angle at blade inlet, degrees, or interpolated blade outlet angle, degrees
TEMP2	Interpolated blade camber angle at blade inlet, degrees, or interpolated blade

Fortran Symbol	Use
	inlet angle, degrees
TEMP3	Interpolated point of maximum camber at inlet radius
ABARN	Function of point of maximum camber at inlet radius
ABARD	Function of point of maximum camber at blade outlet
TANB1N	Tangent of angle between blade inlet and chord line
B1N	Blade inlet angle, radians
TANB2D	Tangent of angle between blade outlet and chord line
B2D	Blade outlet angle, radians
ABAR	Function of point of maximum camber at mean radius
TB1N	Blade section camber angle, radians
RB1N	Angle between blade inlet and chord line, radians
COSA	Mean of cosines of streamline inlet and outlet slope angles
SIGN	Plus a minus one, according to sign of relative inlet angle

SUBROUTINE CASCDE

Fortran Symbol	Use
STAR	Blade section stagger angle, degrees
THETA	Blade section camber angle, degrees
THICK	Blade section thickness/chord ratio, degrees
ZEE	Blade section point of maximum camber as a fraction of chord length

Fortran Symbol	Use
I	Number of station at blade trailing edge
J	Number of streamline on which blade section data is specified
WSPEC	Specified loss coefficient (if applicable)
VF	Number used to correct axial velocity ratio
WORD1(K)	Array used to store Hollerith characters
WORD3(K)	Array used to store Hollerith characters
WORD4(K)	Array used to store Hollerith characters
KBETA2	IBETA2 for blade row
KTHIC	IFTHIC for blade row
KCAX	IFCAX for blade row
KMACHD	IFMACH for blade row
KREYD	IFREYN for blade row
KMLOS	IFMLOS for blade row
KLUSI	IFLVSI for blade row
KPROF	IFPROF for blade row
KREYL	IFREYL for blade row
KLOS	ILOSS for blade row
SOLID	CUPONS for blade section (solidity)
RMN	RMN for blade section (relative inlet Mach number)
NRATIO	Indicator recording whether axial velocity is in, above, or below normal range
B2	Relative outlet flow angle, degrees

Fortran Symbol	Use
B1	BET1 for blade section (relative inlet flow angle), degrees
FRSTB2	Relative outlet flow angle corrected for axial velocity ratio and thickness/chord ratio
WDES	Minimum, low speed loss coefficient
DL	Final loss coefficient
NCHOKE	Indicator controlling logic in Subroutine STLANG
FCOEFL	Blade camber angle expressed as lift coefficient
ATTACK	Minimum loss angle of attack, degrees
B1DES	Low speed minimum loss relative inlet angle, degrees
X2	Blade inlet angle, degrees
X3	Blade inlet angle assuming maximum camber occurs at mid chord, degrees
X1	Function of point of maximum camber
NRATIO	Axial velocity ratio
EX	Store various angles in radians
BE1	Minimum loss inlet angle for equivalent two-dimensional cascade, also actual inlet angle for equivalent two-dimensional cascade, degrees
BE2	Equivalent two-dimensional minimum loss outlet flow angle, also equivalent actual two-dimensional outlet flow angle, degrees
BCHOKE	Low speed choking inlet angle
BSTALL	Low speed stalling inlet angle
SHIFT	Correction to BSTALL, BCHOKE, B1DES for extreme solidity

Fortran Symbol	Use
FACINE	Correction factor for BSTALL, BCHOKE, B1DES for thickness/chord ratio
XEWSTL	BSTALL corrected for thickness/chord ratio
X4	B1DES corrected for thickness/chord ratio
XEWCHK	BCHOKE corrected for thickness/chord ratio
B1DESE	Minimum loss inlet angle for equivalent two-dimensional cascade, degrees
B2DESE	Minimum loss outlet angle for equivalent two-dimensional cascade, degrees
B2DIN	Minimum loss outlet angle corrected for axial velocity ratio, degrees
EMM	"m" in expression to correct deflection for thickness/chord ratio
BEB	"x" in expression to correct deflection for thickness/chord ratio
BEE	"b" in expression to correct deflection for thickness/chord ratio
D10	$\theta m/b$
CAY	$625\ t/c + 37.5\ t/c^2$
DOT	Correction relative outlet angle for thickness/chord ratio
B2DES	Minimum loss outlet flow angle corrected for axial velocity ratio and thickness/chord ratio, degrees
B1RAD	Low speed minimum loss inlet angle, radians
B2RAD	Minimum loss outlet flow angle corrected for axial velocity, ratio and thickness/chord ratio, radians
VMAX	Ratio of maximum surface velocity to

Fortran Symbol	Use
	upstream velocity
CP	$VMAX^2 - 1.0$
CRITM	Critical relative inlet Mach number for blade section
DEQDES	Design diffusion factor
WSTHRD	Function of minimum loss coefficient
B1DCOR	Minimum loss inlet angle corrected for Mach number, degrees
RANGE	Difference between corrected stall and design, or design and choking angles, degrees
B1SCOR	Stalling inlet angle, corrected for Mach number, degrees
B1CCOR	Choking inlet angle, corrected for Mach number, degrees
XPLOS	Exponent in loss/incidence relationship
WOPT	Minimum loss coefficient corrected for actual Mach number
S	Nondimensional incidence
CLMR	Loss coefficient corrected for Mach number and incidence
BE1DES	Inlet angle equivalent to actual for two-dimensional cascade, degrees
BSTLLL	Low speed stalling angle, degrees
B2STA	Equivalent two-dimensional relative outlet flow angle at stall, degrees
BUNCOR	Low speed outlet angle corrected for axial velocity ratio
RMP	Effective relative inlet Mach number
GAMMA	Ratio of specific heats

Fortran Symbol	Use
B2NEW	Minimum loss outlet angle corrected for Mach number effect
RENN	Blade section Reynolds number
RR	Log (RENN) -10.82
RRR	RR -1.48
B2MR	Relative outlet flow angle corrected for axial velocity ratio, thickness/chord ratio, and Mach number
WREN	Increment on loss coefficient due to Reynolds number
DVR	Increment on deviation angle due to Reynolds number, degrees
CHECK	Number indication relative magnitudes of Mach number and Reynolds number deviation corrections
BCD1	Used to store Hollerith characters
BCD2	Used to store Hollerith characters
BCD3	Used to store Hollerith characters
BCD4	Used to store Hollerith characters

SUBROUTINE GRAPH2

Fortran Symbol	Use
D1	Angle for interpolation, solidity of 0.5, degrees
D2	Angle for interpolation, solidity of 1.0, degrees
D3	Angle for interpolation, solidity of 1.5, degrees
S	Solidity
Y	Interpolated angle, degrees

SUBROUTINE STLANG

Fortran Symbol	Use
BIN	Relative inlet flow angle for equivalent two dimensional cascade, degrees
STAR	Blade section stagger anle, degrees
SOLID	Cascade solidity
COEFFL	Blade section camber angle expressed as a lift coefficient
ZEE	Point of maximum camber
NCHOKE	Indicator specifying if choke and stall angles are required
BEXIT	Relative outlet flow angle, degrees
BCHOKE	Choking inlet angle, degrees
BSTALL	Stalling inlet angle, degrees
B1OPT	Minimum loss inlet angle, degrees
J	Streamline number
RM1	Relative inlet Mach number
THETA	Blade section camber angle, degrees
S	Blade section stagger angle, degrees
DELTAC	Increment in lift coefficient between successive curves fitted in the subroutine
CLIFT	Lift coefficient during search between curves for appropriate curves for interpolation
NVM	Records number of curve examined in search
I	Index in searching loop
INDIC	Records whether one or two values have been obtained for subsequent interpolation

Fortran Symbol	Use
D1	First coefficient in choking angle curve equation
D2	Second coefficient in choking angle curve equation
D3	Third coefficient in choking angle curve equation
E1	First coefficient in stalling angle curve equation
E2	Second coefficient in stalling angle curve equation
E3	Third coefficient in stalling angle curve equation
AE15	Choking angle, solidity of 1.5, camber above required value, degrees
AS15	Stalling angle, solidity of 1.5, camber above required value, degrees
BE15	Choking angle, solidity of 1.5, camber below required value, degrees
BS15	Stalling angle, solidity of 1.5, camber below required value, degrees
COUT15	Choking angle, solidity of 1.5, degrees
SOUT15	Stalling angle, solidity of 1.5, degrees
AC10	Choking angle, solidity of 1.0, camber above required value, degrees
AS10	Stalling angle, solidity of 1.0, camber above required value, degrees
BC10	Choking angle, solidity of 1.0, camber below required value, degrees
BS10	Stalling angle, solidity of 1.0, camber below required value, degrees
COUT1	Choking angle, solidity of 1.0, degrees

Fortran Symbol	Use
SOUT1	Stalling angle, solidity to 1.0, degrees
AC05	Choking angle, solidity of 0.5, camber above required value, degrees
AS05	Stalling angle, solidity of 0.5, camber above required value, degrees
BC05	Choking angle, solidity of 0.5, camber below required value, degrees
BS05	Stalling angle solidity of 0.5, camber below required value, degrees
COUT05	Choking angle, solidity of 0.5, degrees
SOUT05	Stalling angle, solidity of 0.5, degrees
COUT	Choking angle, degrees
SOUT	Stalling angle, degrees
BLEAV	Relative outlet flow angle, degrees

SUBROUTINE OUTANG

Fortran Symbol	Use
STAR	Blade section stagger angle, degrees
THETA	Blade section camber angle, degrees
SOLID	Cascade solidity
B1STAR	Relative inlet flow angle for equivalent two-dimensional cascade, degrees
ZEE	Point of maximum camber
BLEAV	Relative outlet flow angle for equivalent two-dimensional cascade, degrees
B1OPT	Minimum loss inlet angle, degrees
J	Streamline number

Fortran Symbol	Use
RMI	Relative inlet Mach number
DBDA(N)	Rate of change of outlet angle with inlet angle for four inlet angles
RC	Blade section stagger angle, radians
EMI	"m" in expression to determine deviation angle
W	"w" is expression to determine deviation angle
EM	"m" is expression to determine deviation angle
DELTA	Deviation angle, degrees
RTH	Blade section camber angle, radians
ABAR	Function of point of maximum camber
TANB2D	Tangent of angle between blade outlet and chordline
FUNCY	"f" is expression to correct relative outlet flow
GUTCY	Correction to relative outlet flow angle, degrees
B2DIN	Relative outlet flow angle
ABI	Absolute value of relative inlet flow angle, degrees
DB	Rate of change of outlet angle with inlet angle at actual inlet angle
N	Records which of four rates of charge of outlet angle with inlet angle are used for interpolation
D	One half rate of change of outlet angle with inlet angle at zero inlet angle
DBSTAR	Minimum loss deflection angle, degrees

SUBROUTINE OPTANG

Fortran Symbol	Use
SOLID	Cascade section solidity
COEFFL	Blade section camber angle expressed as a lift coefficient
ATTACK	Minimum loss angle of attack, degrees
A123(I,J)	Coefficients of curves defining minimum loss angle of attack, coefficient orders 1
M	Records lift coefficients various lift coefficient J during search
DELTAC	Increment between lift coefficients of successive curves
CLIFT	Records lift coefficient during search
A	Minimum angle of attack at lift coefficient above required value
B	Minimum loss angle of attack at lift coefficient below required value
I	Index for coefficient order
J	Index for curve number

SUBROUTINE MCDEVN

Fortran Symbol	Use
CRITM	Blade section critical Mach number
B1RAD	Minimum loss inlet angle, radians
B2RAD	Low speed minimum loss outlet angle, radians
I	Number of station at blade row outlet
J	Streamline number
RMP	Effective relative inlet Mach number

Fortran Symbol	Use
B2NEW	Minimum loss outlet angle corrected for Mach number effect, degrees
GAMMA	Ratio of specific heats,
C1	Cosine (B1RAD)
C2	Cosine (B2RAD)
EXP	$\frac{\gamma+1}{2(\gamma+1)}$
EXPO	$\frac{\gamma-1}{2}$
M	Number of station at blade row inlet
V1CRIT	Critical inlet velocity function
A2	Proportional to streamtube outlet area
A1	Proportional to streamtube inlet area
G	First approximation to critical outlet Mach number
GUESS	Defined estimate of critical outlet Mach number
NCOUNT	Records number of iterations for critical outlet Mach number
EMNEW	Recalculated critical outlet Mach number
DIFF	Difference between successive estimates of critical outlet Mach number
RATIO	Critical velocity ratio across cascade
V1	Actual inlet velocity function
V2	Actual outlet velocity function
EM2	Actual outlet Mach number
COSB2	Cosine of relative outlet flow angle corrected for Mach number effect

SUBROUTINE BLTHIC

Fortran Symbol	Use
I	Station number
BLOCK(I)	Fraction of annulus blocked at station I
HUBI(I)	Integral on hub of $C_m^4 dz$ from transition to station I
CASEI(I)	Integral on casing of $C_m^4 dz$ from transition to station I
THETHB(I)	Momentum thickness on hub at station I, ft
THETCS(I)	Momentum thickness on casing at station I, ft
DXHUB	Axial distance between current and previous computing stations on hub, ft
DXCASE	Axial distance between current and previous computing stations on casing, ft
THETA	Newly calculated momentum thickness, ft
H	Shape factor
DELTA	Displacement thickness, ft
ANNUL	Annulus height, ft
RHUB	Radius of limit of displacement thickness on hub, ft
RCASE	Radius of limit of displacement thickness on casing, ft
GAP	Fraction of annulus height free of boundary layer
RH2	$RHUB^2$
RI12	$(\text{Hub radius})^2$
RRR	Fraction of annulus area free of boundary layer prior to current calculation
DELHUB	Displacement thickness on hub, ft

Fortran Symbol	Use
DELCSE	Displacement thickness on case, ft

SUBROUTINE PRINT

Fortran Symbol	Use
VABS(J)	Absolute velocity at streamline J, ft per sec
GAMMA(J)	Ratio of specific heats at streamline J
XABS(J)	Absolute Mach number at streamline J
DENS(J)	Density at streamline J, lbs per cubic ft
RN(J)	Total pressure ratio from preceding to current station or streamline J
TN(J)	$\frac{\Delta T}{T}$ from preceding to current station on streamline J
RO(J)	Total pressure ratio from inlet to current station on streamline J
TO(J)	$\frac{\Delta T}{T}$ from inlet to current station on streamline J
GAMIN(J)	Ratio of specific heats at inlet on streamline J
GAMUP(J)	Ratio of specific heats at preceding station on streamline J
GOJ(J)	Mean ratio of specific heats between inlet and current station on streamline
GNJ(J)	Mean ratio of specific heats between preceding and current station
G	32.175, being the acceleration due to gravity, ft per sec per sec
CJ	778.0 being Joules equivalent, ft lb per Btu
LNCT	Records lines printed for page control
I	Station number

Fortran Symbol	Use
J	Streamline number
VTAN	Tangential component of velocity, ft per sec
ANG	Absolute whirl angle, defined atan (tangential meridional velocities)
RC	Radius of curvature of streamlines in meridional projection, ft
SLOPE	Streamline slope angle in meridional projection, degrees
A2R	Relative outlet flow angle, degrees
V1R	Relative inlet velocity, ft per sec
V2R	Relative outlet velocity, ft per sec
X2R	Relative outlet Mach number
DHN	De Haller number
DFF	Diffusion factor
DPQ	Static pressure rise coefficient
U1	Blade speed at inlet, ft per sec
U2	Blade speed at outlet, ft per sec
CUMRN	Mass-flow-weighted mean total pressure ratio from preceding to current station
CUMRO	Mass-flow-weighted mean total pressure ratio from inlet to current station
CUMTO	Mass-flow-weighted mean $\frac{\Delta T}{T}$ from inlet to current station
CUMTN	Mass-flow-weighted mean $\frac{\Delta T}{T}$ from preceding to current station
GAMMN	Mass-flow-weighted mean ratio of specific heats between preceding and current station
GAMMO	Mass-flow-weighted mean ratio of specific heats between inlet and current stations

Fortran Symbol	Use
CUMEN	Isentropic efficiency between preceding and current stations, mean for all radii
CUMEO	Isentropic efficiency between inlet and current stations, mean for all radii
EN	Isentropic efficiency between preceding and current stations
EO	Isentropic efficiency between inlet and current stations

SUBROUTINE PUNCH

Fortran Symbol	Use
J1	See MAIN Program
J21	See MAIN Program
J22	See MAIN Program
J31	See MAIN Program
J32	See MAIN Program
J41	See MAIN Program
J42	See MAIN Program
J5	See MAIN Program
LTAPE1	See MAIN Program
LTAPE2	See MAIN Program
LTAPE3	See MAIN Program
LTAPE4	See MAIN Program
LTAPE5	See MAIN Program
NTHET	Current sector number
IFSUM	Set to 1 in program
NUNIT	Number of unit on which data is currently being written
II1	Number of first station for which principal data is written onto unit NUNIT

Fortran Symbol	Use
I12	Number of last station for which principal data is written onto unit NUNIT
I	Station number

SUBROUTINE THETA

Fortran Symbol	Use
J1	See MAIN Program
J21	See MAIN Program
J22	See MAIN Program
J31	See MAIN Program
J32	See MAIN Program
J41	See MAIN Program
J42	See MAIN Program
J5	See MAIN Program
LTAPE1	See MAIN Program
LTAPE2	See MAIN Program
LTAPE3	See MAIN Program
LTAPE4	See MAIN Program
LTAPE5	See MAIN Program
JTAPE1	See MAIN Program
JTAPE2	See MAIN Program
JTAPE3	See MAIN Program
JTAPE4	See MAIN Program
NTAPE	See MAIN Program
KTAPE1	K1 in MAIN Program
KTAPE2	K2 in MAIN Program

Fortran Symbol	Use
LTAPE	Current input unit number from "L-Series"
JTAPE	Current output unit number from "J-Series"
IZALE	Number of station at compressor inlet
IZATE	Number of station at compressor outlet
IZA1	Number of first station currently being processed
IZA2	Number of last station currently being processed
RADIAN	180.0/3.1415926
GZERO	32.174
WORKJ	778.17
RFAC1	Static pressure gradient relaxation factor
RFAC2	Sector centerline relocation relaxation factor
RFAC3	Increment for elimination of perturbation solution
IFAC1	Cycle number when elimination of perturbation solution commences
IFAC2	Number of iterations through perturbation procedure, plus 1
N	Subscript indicating sector number
I	Subscript indicating station number
J	Subscript indicating stream surface number
XNTHET	Number of sectors
ROMEAN	Mean density at one station, one radius
IZA	Station number (absolute)
TAN30	Tangent of station lean angle

Fortran Symbol	Use
KATZ	Indicator noting whether calculation is being made in inlet duct, compressor, or outlet duct
ROME1	Mean density at boundary of perturbation solution
ROME2	Mean density at boundary of perturbation solution
CMEAN1	Mean meridional velocity at boundary of perturbation solution
CMEAN2	Mean meridional velocity at boundary of perturbation solution
AMEAN1	Mean tangent of flow angle at boundary of perturbation solution
CTHET1	Mean tangential velocity component at boundary of perturbation solution
CTHET2	Mean tangential velocity component at boundary of perturbation solution
AZERO	Constant coefficient in Fourier series describing nonuniformity at boundary of perturbation solution
RFAC5	Current "mix" factor in elimination of perturbation solution
PMEAN	Mean static pressure at one station, one radius
NIZA1	First station number for stations to be processed next
NIZA2	Last station number for stations to be processed next
NEWJ	Output unit number for next stations to be processed
NEWL	Input unit number for next stations to be processed

SUBROUTINE OVERAL

Fortran Symbol	Use
DELW(N,J)	Flow through streamtube defined by streamline J and J+1, sector N, lbs per sec
PINLET(J,N)	Inlet total pressure, streamline J, sector N, lbs per sq ft
TINLET(J,N)	Inlet total temperature, streamline J, sector N, deg R
R(I)	Mean total pressure ratio between inlet and station I
EI(I)	Mean isentropic efficiency between inlet and station I
EQ(I)	Mean polytropic efficiency between inlet and station I
TZ(I)	Mean $\frac{\Delta T}{T}$ between station I and inlet
NLINES	Number of streamlines
NTUB	NLINES -1
TOTALW	Total flow through compressor, lbs per second
N	Sector number
NP1	N+1
NM1	N-1
WIDTH	Sector width x 2, radians
I	Station number index
IZA	Absolute station number
SPEED	Speed of rotation of blade row, rev per min

FUNCTION SERVUS

Fortran Symbol	Use
THETA(N,I)	Sector centerline location at sector N, station I

Fortran Symbol	Use
THETA1(N)	Sector centerline location at sector N, inlet station
NTHETA	Number of sectors

FUNCTION FINDY

Fortran Symbol	Use
X	Input independent variable
ARGX(I)	Table of independent variable
ARGY(I)	Table of dependent variable

FUNCTION DELTA

Fortran Symbol	Use
THETA2	Location of sector centerline
THETA1	Location of sector centerline
DELTA	Difference between THETA2 and THETA1

FUNCTION DERIV

Fortran Symbol	Use
P(J,N,I)	Static pressure, stream surface J, sector N, station I
THETA(N,I)	Sector centerline location, sector N, station I
DERIV	Circumferential static pressure gradient,

SUBROUTINE FOUR

Fortran Symbol	Use
BANG	Slope of function between two points
NN	Index for Fourier coefficient number
XN	Fourier coefficient number
P(N)	Value of function at sector N

Fortran Symbol	Use
THETA(N,I)	Location of centerline at sector N, station I
AZERO	Constant coefficient
AN(NN)	Coefficient of NN cosine term in Fourier series
BN(NN)	Coefficient of NN sine term in Fourier series
NMAX	Number of terms in Fourier series

FUNCTION XMEAN

Fortran Symbol	Use
P(J,N,I)	Value of function at stream surface J, sector N, station I
THETA(N,I)	Sector centerline location at sector N, station I

SUBROUTINE EDWIN

Fortran Symbol	Use
Y(I)	locations (sector centerlines) at which perturbation static pressure is to be calculated
Z	Nondimensional axial location at which perturbation static pressure is to be calculated
P(I)	Perturbation static pressure/density for the i^{th} sector
DP(I)	Perturbation static pressure circumferential gradient for the i^{th} sector
NT	Number of sectors at which pressure is to be calculated
DPMQ	Fourier summation for static pressure circumferential gradient
PMQ	Fourier summation for perturbation static pressure/density

Fortran Symbol	Use
DETQ	e^{-jmSoz}
M	Fourier summation index
XM	M
A	e^{mz} or cos $m\Theta i$
WBAR	Mean axial velocity at z = Z
VBAR	Mean tangential velocity at z = Z
VBARD	Axial derivative of mean tangential velocity at z = Z
XK	Either XK1 (K_1) or XK2 (K_2) from COMMON/ONE/
B	e^{kz} or sin $m\Theta i$

SUBROUTINE NEWRAP

Fortran Symbol	
W1	Mean axial or tangential velocity at z = Z1
W2	Unused
W3	Unused
W4	Mean axial or tangential velocity at z = Z4
Z1	Nondimensional axial location corresponding to W1
Z2	Unused
Z3	Unused
Z4	Nondimensional axial location corresponding to W4
ND	If 0, W1 and W4 are axial velocities; if 1, W1 and W4 are tangential velocities
A	e^{k1Z1}

Fortran Symbol	Use
B	$e^{k_2 Z_1}$
C	$e^{k_1 Z_4}$
D	$e^{k_2 Z_4}$
E	W1-WZ or W1-VZ
F	W4-WZ or W4-VZ

SUBROUTINE CPOUR

Fortran Symbol	Use
R	Number to be normalized to form $R' \cdot 10^L$
L	Power of 10 in normalized form
S	$\|R\|$

FUNCTION VBARF

Fortran Symbol	Use
Z	Nondimensional axial location
VBARF	Mean tangential velocity at $z = Z$

FUNCTION VBARDF

Fortran Symbol	Use
Z	Nondimensional axial location
VBARDF	Axial derivative of mean tangential velocity at $z = Z$

FUNCTION WBARF

Fortran Symbol	Use
Z	Nondimensional axial location
WBARF	Mean axial velocity at $z = Z$

SUBROUTINE CSIMEQ

Fortran Symbol	Use
A(I,J)	Coefficient matrix augmented at constant

Fortran Symbol	Use
	vectors
X(I,K)	Solution vectors
DET	Normalized determinant of coefficient matrix: determinant = DET*10**LEXP
NR	Number of unknowns
NS	Number of solutions desired
NDEX	If < 0, only determinant calculated; if = 0, only solution vectors; if > 0, both
LSGN	0 unless matrix is singular
LEXP	Normalized power of 10 of determinant: determinant = DET*10**LEXP
NRD	Dimension of A in calling program
NCD	Dimension of A in calling program
NSD	Dimension of X in calling program
R	Dummy variable used locally
NC	NR+NS: number of columns in A matrix
I	Row index
S	Dummy variable used locally
J	Column index
K	Subsidiary row index
T	Dummy variable used locally
L	Resubstitution summation index

SUBROUTINE GEORGE

Fortran Symbol	Use
Y(I)	locations (sector centerlines) at which perturbation axial and tangential velocities are to be calculated

Fortran Symbol	Use
Z	Nondimensional axial location at which perturbation velocities are to be calculated
U(I)	Perturbation axial velocity for the i^{th} sector
V(I)	Perturbation tangential velocity for the i^{th} sector
NT	Number of sectors at which velocity is to be calculated
ALPHMQ	Fourier summation for perturbation axial velocity
BETAMQ	Fourier summation for perturbation tangential velocity
DETQ	e^{-jmS_0z}
M	Fourier summation index
XM	M
A	e^{mz} or $\cos m\theta_i$
XK	Either XK1 or XK2 from COMMON/ONE/
B	$e^{k_1 z}$ or $\sin m\theta_i$

SUBROUTINE SMITH

Fortran Symbol	Use
COEMQ(I,J)	Boundary condition coefficient matrix (retained)
SOLMQ(I,J)	Solution vector of simultaneous boundary condition equations
BOEMQ(I,J)	Boundary condition coefficient matrix (not retained)
HMQ	Complex Fourier coefficient corresponding to boundary condition at z = XL1 and CNX and DNZ

Fortran Symbol	Use
PMQ	Complex Fourier coefficient corresponding to boundary condition at z = XL2 and CNF and DNF
TMQ	Complex Fourier coefficient corresponding to boundary condition at z = XL3 and ENF and FNF
AQQ	$-jm = -m\sqrt{-1}$
BQQ	Used variously as a complex quantity
DETQ	Unused dummy argument (see CSIMEQ subroutine)
M	Fourier index
XM	M
XZ	Nondimensional axial location, either XL1, XL2, or XL3
A	EXP (XM*XZ)
IND	Either IND2 or IND3 from COMMON/ONE/
I	Index of boundary condition equation; I=1, 2, or 3
WBAR	Mean axial velocity at z = XZ
VBAR	Mean tangential velocity at z = XZ
VBARD	Axial derivative of mean tangential velocity at z = XZ
J	Matrix index
LSGN	Unused dummy argument (see CSIMEQ subroutine)
LEXP	Unused dummy argument (see CSIMEQ subroutine)
XK	Either XK1 or XK2 from COMMON/ONE/
WC	Either WCAP1 or WCAP2 from COMMON/ONE/
VC	Either VCAP1 or VCAP2 from COMMON/ONE/

SUBROUTINE PCTURE

Fortran Symbol	Use
NTAPE	See MAIN Program
JTAPE1	See MAIN Program
JTAPE2	See MAIN Program
JTAPE3	See MAIN Program
JTAPE4	See MAIN Program
JTAPE5	See MAIN Program
LTAPE1	See MAIN Program
LTAPE2	See MAIN Program
LTAPE3	See MAIN Program
LTAPE4	See MAIN Program
LTAPE5	See MAIN Program
J1	See MAIN Program
J21	See MAIN Program
J22	See MAIN Program
J31	See MAIN Program
J32	See MAIN Program
J41	See MAIN Program
J42	See MAIN Program
J5	See MAIN Program
K1	K2 in MAIN Program
NTHETA	Number of sectors
TITLE(K)	Title for job
IFPRIN	Indicator controlling output options

Fortran Symbol	Use
R32(N)	Angular location of sector N in inlet, degrees
CP	Specific heat, Btu per lb deg R
C1	180/3.1415926535
C2	2.0*32.175*778.0*CP, being $2gJC_p$, ft^2 per sec^2 deg R
JTAPE	Current unit from "L-Series"
II	Number of stations for which there is data on unit JTAPE
III	Absolute station number
DUM	Dummy variable used in skip-reading
N	Sector number
IFFULL	Indicator indicating external data on unit LTAPE1 (for each sector)
I1	Total number of stations
I2	Number of streamlines
I3	Number of blading data radii
I21	Indicator specifying if station lean angles and solution types are detailed
GASK	Gas constant, ft lbm per lb deg R
I	Station number
I23	Indicator specifying presence or otherwise of blades
IFMACH	Indicator specifying cascade analysis option
IFLVSI	Indicator specifying cascade analysis option
J	Streamline number
IDUM	Dummy variable used in skip-reading

Fortran Symbol	Use
N1	Indicator noting first storage location in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N2	Indicator noting first storage locations in triple-subscripted arrays (which are equivalent to singly subscripted arrays) holding values applying at hub streamline
N3	Indicator noting first storage location in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N4	Indicator noting first storage location in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N5	Indicator noting first storage location in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N6	Indicator noting first storage location in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N7	Indicator noting first storage location in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N8	Indicator noting first storage location in triple-subscripted arrays (which are equivalenced to single subscripted arrays) holding values applying at hub streamline

Fortran Symbol	Use
N9	Indicator noting first storage locations in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N10	Indicator noting first storage locations in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N11	Indicator noting first storage locations in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N12	Indicator noting first storage locations in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N13	Indicator noting first storage locations in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N14	Indicator noting first storage locations in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N15	Indicator noting first storage locations in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N16	Indicator noting first storage locations in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline

Fortran Symbol	Use
N17	Indicator noting first storage locations in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N18	Indicator noting first storage locations in triple-subscripted arrays (which are equivalenced to singly subscripted arrays) holding values applying at hub streamline
N19	Indicator noting first storage locations in triple-subscripted arrays (which are equivalenced to single subscripted arrays) holding values applying at hub streamline
C3	
C4	Plus or minus one in order to control conversion from total to static temperature, or vice versa
TT	Static or total temperature, deg R
JTAPE1	See MAIN Program
JTAPE2	See MAIN Program
JTAPE3	See MAIN Program
JTAPE4	See MAIN Program
JTAPE5	See MAIN Program
LOG2	Standard output unit number
LOG4	K2 in MAIN Program
NTHETA	See MAIN Program
TITLE(K)	See MAIN Program
J1	See MAIN Program
J21	See MAIN Program
J22	See MAIN Program

Fortran Symbol	Use
J31	See MAIN Program
J32	See MAIN Program
J41	See MAIN Program
J42	See MAIN Program
J5	See MAIN Program
CP	Specific heat, Btu per lb deg R
R(J)	Approximate radius of stream surface J, ft
Y(N,J)	Vertical coordinate of point for polar plot, ft
SYMBOL(N)	Array storing Hollerith characters used to plot N sector centerlines
LINE(L)	Array printed to create one line of plot
X1	Store Hollerith character
DASH	Store Hollerith character
CROSS	Store Hollerith character
BLANK	Store Hollerith character
LOG3	Current unit number from "J-Series"
NX	Total number of stations
NLINES	Number of streamlines
NBETA	Number of blading data radii
J	Streamline number
IEND	Number of stations for which data is on unit number LOG3
III	Current station number
I1	1
I	Station number

Fortran Symbol	Use
N	Sector number
II	Station number
IBETA2	Indicator specifying pressure or otherwise of blades
IFMACH	Indicator specifying cascade analyses option
IFLVSI	Indicator specifying cascade analyses option
YINC	Vertical increment represented by one line of plot, ft
YSTART	Upper limit of vertical scale represented by current line of plot, ft
SCALE	Scale of plot produced, ft per inch
KLINE	Line number of plot
L	Location number in line of plot

SUBROUTINE MPLOT

Fortran Symbol	Use
IX	Number of values of horizontal coordinate to be plotted
IY	Number of values of vertical coordinate to be plotted for each value of horizontal coordinate
LOGI	Standard output unit number
X(I)	Value of horizontal coordinate, point I
YI(I)	First value of vertical coordinate, point I
Y2(I)	Second value of vertical coordinate, point I
Y3(I)	Third value of vertical coordinate, point I

Fortran Symbol	Use
Y4(I)	Fourth value of vertical coordinate, point I
Y5(I)	Fifth value of vertical coordinate, point I
Y6(I)	Sixth value of vertical coordinate, point I
Y7(I)	Seventh value of vertical coordinate, point I
Y8(I)	Eight value of vertical coordinate, point I
Y9(I)	Ninth value of vertical coordinate, point I
Y10(I)	Tenth value of vertical coordinate, point I
Y11(I)	Eleventh value of vertical coordinate, point I
Y12(I)	Twelfth value of vertical coordinate, point I
Y13(I)	Thirteenth value of vertical coordinate, point I
Y14(I)	Fourteenth value of vertical coordinate, point I
Y15(I)	Fifteenth value of vertical coordinate, point I
Y16(I)	Sixteenth value of vertical coordinate, point I
Y17(I)	Seventeenth value of vertical coordinate, point I
Y18(I)	Eighteenth value of vertical coordinate, point I
SYMBOL(N)	Average storing Hollerith characters used to plot eighteen curves
LINE(L)	Array printed to create one line of plot

Fortran Symbol	Use
XNUM(K)	Array of numbers describing horizontal scale
DASH	Stores Hollerith character
CROSS	Stores Hollerith character
BLANK	Stores Hollerith character
XI	Stores Hollerith character
YMIN	Minimum value of vertical coordinate
XMIN	Minimum value of horizontal coordinate
YMAX	Maximum value of vertical coordinate
XMAX	Maximum value of horizontal coordinate
I	Number of point currently being processed
YH	Maximum value on vertical scale
YL	Minimum value on vertical scale
XH	Maximum value of horizontal scale
XL	Minimum value of horizontal scale
MX	Normalizing factor to put horizontal scale maximum value in the range 1 to 10
MY	Normalizing factor to put vertical scale maximum value in the range 1 to 10
YINC	Vertical increment represented by one line of plot
YINC2	YINC/2.0
XRANGE	Difference between maximum and minimum values of horizontal scale
KLINE	Line number of plot
L	Location number in line of plot
YNUM	Number describing vertical scale

FORTRAN PROGRAM LISTING

```
C     MAIN PROGRAM   NREC PROJECT 1126A
      DIMENSION R32(18),TITLE(12),HEAD(4),J1(1),J21(1),J22(1),J31(1),J32
     1(1),J41(1),J42(1),J5(1)
      DATA(HEAD(I),I=1,4)/6HNEW CA,6HCONTIN,6HSE    ,6HUATION/
      ITAPE=5
      NTAPE=6
      JTAPE1=1
      JTAPE2=2
      JTAPE3=3
      JTAPE4=4
      JTAPE5=7
      LTAPE1=8
      LTAPE2=9
      LTAPE3=10
      LTAPE4=11
      LTAPE5=12
      KTAPE1=13
      KTAPE2=14
      K1=KTAPE1
      K2=KTAPE2
      READ(ITAPE,10)TITLE,NTHETA,NLOOPS,IFSTAR,IFPRIN,RFAC1,RFAC2,RFAC3,
     1RFAC4,IFAC1,IFAC2,(R32(N),N=1,NTHETA)
10    FORMAT(12A6,/,4I6,/,4F8.4,/,2I6,/,(9F8.4))
      IF(NTHETA.GT.1)REWIND K1
      IF(NTHETA.GT.1)REWIND K2
      WRITE(NTAPE,20)TITLE,NTHETA,NLOOPS,HEAD(IFSTAR+1),HEAD(IFSTAR+3),I
     1FPRIN,RFAC1,RFAC2,RFAC3,RFAC4,IFAC1,IFAC2,(R32(N),N=1,NTHETA)
20    FORMAT(1H1,31X,70HPROGRAM TO ESTIMATE AXIAL COMPRESSOR PERFORMANCE
     1 WITH INLET DISTORTION/32X,70H************************************
     2**********************************,///,10X,11HJOB TITLE =,12A6,/,1
     30X,19HNUMBER OF SECTORS =,I3,/,10X,18HNUMBER OF CYCLES =,I3,/,10X,
     410HTHIS IS A ,2A6,/,10X,8HIFPRIN =,I3,54H (CONTROLS OUTPUT PRINTIN
     5G FOR CYCLES AFTER THE FIRST)/,10X,7HRFAC1 =,F8.5,8H RFAC2 =,F8.5,
     68H RFAC3 =F8.5,8H RFAC4 =,F8.5,8H IFAC1 =,I3,8H IFAC2 =I3,/,10X,41
     7HSECTOR CENTER-LINE LOCATIONS AT THE INLET,F9.1,/,(54X,F6.1))
      IF(IFSTAR.EQ.0)CALL INPUT(NTHETA,ITAPE,NTAPE,JTAPE1,JTAPE2,JTAPE3,
     1JTAPE4,JTAPE5,LTAPE1,LTAPE2,LTAPE3,LTAPE4,LTAPE5,KTAPE1,KTAPE2)
      DO 100 LP=1,NLOOPS
      IF(NTHETA.EQ.1)GO TO 22
      REWIND LTAPE1
      WRITE(LTAPE1,25)TITLE,NTHETA,LP,NLOOPS,IFPRIN,RFAC1,RFAC2,RFAC3,RF
     1AC4,IFAC1,IFAC2,(R32(N),N=1,NTHETA)
25    FORMAT(12A6,/,4I6,/,4F8.5,/,2I6,/,(9F8.4))
22    CONTINUE
      WRITE(NTAPE,30)LP
30    FORMAT(34H1ENTERING NTHETA DISTORTS ON CYCLE,I3)
      DO 40 LDUM= 1,NTHETA
40    CALL DISTOR (J1,J21,J22,J31,J32,J41,J42,J5,LTAPE1,LTAPE2,LTAPE3,LT
     1APE4,LTAPE5,NTAPE,JTAPE1,JTAPE2,JTAPE3,JTAPE4,JTAPE5,LDUM)
      IF(NTHETA.EQ.1)STOP
      IF(J21.EQ.0)GO TO 70
      IF(J31.EQ.0)GO TO 65
      IF(J41.EQ.0)GO TO 60
      REWIND JTAPE4
      REWIND LTAPE4
60    REWIND JTAPE3
      REWIND LTAPE3
65    REWIND JTAPE2
      REWIND LTAPE2
70    REWIND JTAPE1
```

```
      REWIND LTAPE1                                                     S01*0060
      REWIND JTAPE5                                                     S01*0061
      REWIND LTAPE5                                                     S01*0062
      WRITE(NTAPE,50)LP                                                 S01*0063
50    FORMAT(24H1ENTERING THETA ON CYCLE,I3)                            S01*0064
      CALL THETA(J1,J21,J22,J31,J32,J41,J42,J5,LTAPE1,LTAPE2,LTAPE3,LTAPS01*0065
     1E4,LTAPE5,JTAPE1,JTAPE2,JTAPE3,JTAPE4,JTAPE5,NTAPE,K1,K2,IFSTAR)  S01*0066
      CALL PCTURE(NTAPE,JTAPE1,JTAPE2,JTAPE3,JTAPE4,JTAPE5,LTAPE1,LTAPE2S01*0067
     1,LTAPE3,LTAPE4,LTAPE5,J1,J21,J22,J31,J32,J41,J42,J5,K2,NTHETA,TITLS01*0068
     2E,IFPRIN,R32)                                                     S01*0069
      KKK=K1                                                            S01*0070
      K1=K2                                                             S01*0071
      K2=KKK                                                            S01*0072
      REWIND K1                                                         S01*0073
      REWIND K2                                                         S01*0074
100   CONTINUE                                                          S01*0075
      STOP                                                              S01*0076
      END                                                               S01*0077
```

```
      SUBROUTINE INPUT(NTHETA,ITAPE,NTAPE,JTAPE1,JTAPE2,JTAPE3,JTAPE4,JT
     1APE5,LTAPE1,LTAPE2,LTAPE3,LTAPE4,LTAPE5,KTAPE1,KTAPE2)
      DIMENSION RPM2(30),IBETA2(30),IFTHIC(30),IFCAX(30),IFMACH(30),IFRE
     1YN(30),ILOSS(30),IFMLOS(30),IFLVSI(30),IFPROF(30),IFREYL(30),EXPLO
     2S(30),XMACH(30),BBP(30,15),STAG(30,15),BETA(30,15),SIGMA(30,15),BL
     3AK(30,15),ZBAR(30,15),CLOSS(30,15),CO(30,15),DELM(15),BLAM(30),X(3
     40),RHP(30),RSP(30),IMACHI(30),ANGLN(30),COMENT(12,18),BHP(15,18),T
     5OCO(15,18),POCO(15,18),AFCO(15,18)
      READ(ITAPE,10)NX,NLINES,NBETA,NTOPO,IFSIMP,NPASS,IFBL,IFRPM,ITER,N
     1PLOT,INCPO,NWRIT,INCWRI,IFTYPE,TOLCX,HUBLOC,GASK,CP,VIS1,VIS2
10    FORMAT(12I6,/,2I6,/,6F8.4)
      WRITE(NTAPE,15)NX,NLINES,NBETA,NTOPO,IFSIMP,NPASS,IFBL,IFRPM,ITER,
     1NPLOT,INCPO,NWRIT,INCWRI,IFTYPE,TOLCX,HUBLOC,GASK,CP,VIS1,VIS2
15    FORMAT(/,10X,20HNUMBER OF STATIONS =,I3,/,10X,23HNUMBER OF STREAML
     1INES =,I3,/,10X,30HNUMBER OF BLADING DATA RADII =,I3,/,10X,38HNUMB
     *ER OF INLET CONDITION DATA RADII =,I3,/,10X,8HIFSIMP =,I3,57H (2 -
     $S.R.E.  .NE.2 -L.S.Q. STREAMLINES,NPOINT = IFSIMP+2),    /,10X,36H
     3MAXIMUM NUMBER OF PASSES PER CYCLE =,I3,/,10X,6HIFBL =,I3,67H (1 -
     4BLOCKAGE HELD AT DATA VALUES  2 -ANNULUS WALL B.L. CALCULATED),/,1
     50X,7HIFRPM =,I3,43H (CONTROLS OUTPUT PRINTING FOR FIRST CYCLE),/,1
     60X,6HITER =,I3,62H (1 -PRINT ALL VELOCITIES DURING ITERATIONS  2 -
     7NORMAL OPTION),/,10X,7HNPLOT =,I3,54H (FIRST PASS DURING WHICH CAS
     8CADE ANALYSIS IS PRINTED),/,10X,7HINCPO =,I3,22H (INCREMENT FOR AB
     9OVE),/,10X,7HNWRIT =,I3,60H (FIRST PASS DURING WHICH VELOCITY TRIA
     ANGLE DATA IS PRINTED),/,10X,8HINCWRI =,I3,22H (INCREMENT FOR ABOVE
     B),/,10X,8HIFTYPE =,I3,102H (0 -ALL STATIONS UPRIGHT,ALL SOLUTIONS 
     CSUBSONIC  1 -STATION LEAN ANGLES AND SOLUTION TYPES SPECIFIED),/,1
     D0X,22HCONTINUITY TOLERANCE =,F7.4,/,10X,35HFRACTION OF INLET BLOCK
     EAGE ON HUB =,F7.4,/,10X,14HGAS CONSTANT =,F9.4,/,10X,15HSPECIFIC H
     FEAT =,F8.5,/,10X,29HFIRST VISCOSITY COEFFICIENT =,E10.3,/,10X,30HS
     GECOND VISCOSITY COEFFICIENT =,E10.3)
      WRITE(NTAPE,18)
18    FORMAT(1H1,9X,46HSTATION-TO-STATION CHANGES ARE PRESCRIBED THUS,//
     1)
      DO 100 I=2,NX
      RPM2(I)=0.0
      READ(ITAPE,20)IBETA2(I),IFTHIC(I),IFCAX(I),IFMACH(I),IFREYN(I),ILO
     1SS(I),IFMLOS(I),IFLVSI(I),IFPROF(I),IFREYL(I)
20    FORMAT(10I6)
      IB=IBETA2(I)+1
      IF(IB.EQ.7)GO TO 80
      IF(IFLVSI(I).EQ.1)READ(ITAPE,25)EXPLOS(I)
25    FORMAT(F8.4)
      IF(IFMACH(I).EQ.3)READ(ITAPE,25)XMACH(I)
      READ(ITAPE,25)RPM2(I)
      READ(ITAPE,30)(BBP(I,J),STAG(I,J),BETA(I,J),SIGMA(I,J),BLAK(I,J),
     1ZBAR(I,J),CLOSS(I,J),CO(I,J),J=1,NBETA)
30    FORMAT(8F8.4)
      WRITE(NTAPE,33)I,RPM2(I)
      WRITE(NTAPE,32)IBETA2(I),IFTHIC(I),IFCAX(I),IFMACH(I),IFREYN(I),IL
     1OSS(I),IFMLOS(I),IFLVSI(I),IFPROF(I),IFREYL(I)
32    FORMAT(/,24X,8HIBETA2 =,I1,9H IFTHIC =,I1,8H IFCAX =,I1,9H IFMACH 
     1=,I1,9H IFREYN =,I1,8H ILOSS =,I1,9H IFMLOS =,I1,9H IFLVSI =,I1,9H
     2 IFPROF =,I1,9H IFREYL =,I1,//)
      GO TO(750,35,40,750,65,40,750),IB
33    FORMAT(/18X7HSTATION,I3,59H FOLLOWS A BLADE DESCRIBED BY THE FOLLO
     1WING AND ROTATING AT,F9.1,4H RPM,//)
35    WRITE(NTAPE,36)(BBP(I,J),STAG(I,J),CLOSS(I,J),J=1,NBETA)
36    FORMAT(24X,34HRADIUS    RELATIVE  FLOW      LOSS,/,38X,5HANGLE,9X,
     111HCOEFFICIENT,//,(23X,F7.4,F13.2,F17.5))
```

```
      GO TO 100
40    WRITE(NTAPE,42)
42    FORMAT(24X,99HRADIUS    STAGGER    CAMBER    SOLIDITY   THICK/CHORD
     1  MAX.CAM.PT.    LOSS COEFF.    ADDIT.LOSS FACTOR,//)
      IF(IB.EQ.6)GO TO 60
44    IL=ILOSS(I)+1
      FO SO(750,45,50,55,750),IL
45    WRITE(NTAPE,46)(BBP(I,J),STAG(I,J),BETA(I,J),SIGMA(I,J),BLAK(I,J),
     1ZBAR(I,J),CLOSS(I,J),J=1,NBETA)
46    FORMAT(23X,F7.4,F11.2,F10.2,F10.3,F11.4,F13.3,F16.5)
      GO TO 100
50    WRITE(NTAPE,52)(BBP(I,J),STAG(I,J),BETA(I,J),SIGMA(I,J),BLAK(I,J),
     1ZBAR(I,J),CO(I,J),J=1,NBETA)
52    FORMAT(23X,F7.4,F11.2,F10.2,F10.3,F11.4,F13.3,F32.3)
      GO TO 100
55    WRITE(NTAPE,56)(BBP(I,J),STAG(I,J),BETA(I,J),SIGMA(I,J),BLAK(I,J),
     1ZBAR(I,J),CLOSS(I,J),CO(I,J),J=1,NBETA)
56    FORMAT(23X,F7.4,F11.2,F10.2,F10.3,F11.4,F13.3,F16.5,F17.3)
      GO TO 100
60    WRITE(NTAPE,61)(BBP(I,J),STAG(I,J),BETA(I,J),SIGMA(I,J),CLOSS(I,J)
     1,J=1,NBETA)
61    FORMAT(23X,F7.4,F11.2,F10.2,F10.3,F40.5)
      GO TO 100
65    WRITE(NTAPE,66)
66    FORMAT(38X,12HBLADE-ANGLES,/,24X,99HRADIUS       INLET     OUTLET
     1SOLIDITY  THICK/CHORD   MAX.CAM.PT.   LOSS COEFF.   ADDIT.LOSS FAC
     2TOR,//)
      GO TO 44
80    WRITE(NTAPE,82)I
82    FORMAT(/,18X,7HSTATION,I3,27H FOLLOWS A BLADE FREE SPACE)
100   CONTINUE
      READ(ITAPE,105)(DELM(J),J=1,NLINES)
105   FORMAT(9F8.5)
      READ(ITAPE,105)(BLAM(I),I=1,NX)
      READ(ITAPE,108)FLOWI,ESTFLO,WIDTH
108   FORMAT(F8.4)
      READ(ITAPE,110)(X(I),RHP(I),RSP(I),I=1,NX)
110   FORMAT(3F8.4)
      IF(IFTYPE.NE.0)GO TO 125
      DO 115 I=1,NX
      IMACHI(I)=0
115   ANGLN(I)=0.0
      GO TO 135
125   READ(ITAPE,126)(IMACHI(I),I=1,NX)
126   FORMAT(12I6)
      READ(ITAPE,128)(ANGLN(I),I=1,NX)
128   FORMAT(9F8.4)
135   WRITE(NTAPE,140)(I,X(I),RHP(I),RSP(I),ANGLN(I),BLAM(I),IMACHI(I),I
     1=1,NX)
140   FORMAT(1H1,9X,59HANNULUS GEOMETRY SPECIFICATION AND SOLUTION TYPE
     1INDICATORS,//,18X,66HSTATION     AXIAL       HUB      CASING     LEA
     2N    BLOCK      IMACHI,/,18X,88HNUMBER    LOCATION     RADIUS    RAD
     3IUS    ANGLE    -AGE        (0 -SUBSONIC  1 -SUPERSONIC),//,(19X,I3,F
     413.4,F10.4,F11.4,F9.3,F9.4,I8))
      WRITE(NTAPE,141)FLOWI,ESTFLO,WIDTH,(DELM(J),J=1,NLINES)
141   FORMAT(/,10X,28HDOWNSTREAM STATIC PRESSURE =,F9.2,17H ESTIMATED FL
     1OW =,F7.2,23H CHARACTERISTIC WIDTH =,F6.3,//,10X,50HFRACTIONS OF I
     2NLET BETWEEN HUB AND EACH STREAMLINE,//,(10X,9F10.4))
      DO 142 N=1,NTHETA
142   READ(ITAPE,143)(COMENT(K,N),K=1,12),(BHP(J,N),TOCO(J,N),POCO(J,N),
```

```
     1AFCO(J,N),J=1,NTOPO)
 143  FORMAT(12A6,/,(4F8.4))
      WRITE(NTAPE,145)
 145  FORMAT(/,10X,23HSECTOR INLET CONDITIONS)
      DO 155 N=1,NTHETA
      WRITE(NTAPE,150)N,(COMENT(K,N),K=1,12),(BHP(J,N),TOCO(J,N),POCO(J,
     1N),AFCO(J,N),J=1,NTOPO)
 150  FORMAT(/,10X,6HSECTOR,I3,9H COMENT =,12A6,//24X,38HRADIUS     TOTAL
     1           TOTAL      FLOW,/,32X,31HTEMPERATURE   PRESSURE    ANGLE,/
     2/,(23X,F7.4,F11.2,F13.1,F9.2))
 155  CONTINUE
      DO 160 I=2,NX
      J1=I
      IF(IBETA2(I+1).NE.6)GO TO 170
 160  CONTINUE
 170  J21=J1+1
      DO 180 I=1,NX
      II=NX-I+1
      J22=II-1
      IF(IBETA2(II).NE.6)GO TO 190
 180  CONTINUE
 190  IF(J22-J21.GT.5)GO TO 200
      IF(J22.EQ.J21-1)GO TO 195
      J31=0
      J32=0
      J41=0
      J42=0
      J5=J22+1
      GO TO 260
 195  J21=0
      J22=0
      J31=0
      J32=0
      J41=0
      J42=0
      J5=J1+1
      GO TO 260
 200  J32=J22
      J22=J21+5
      J31=J22+1
      IF(J32-J31.GT.5)GO TO 230
      J41=0
      J42=0
      J5=J32+1
      GO TO 260
 230  J42=J32
      J32=J31+5
      J41=J32+1
      J5=J42+1
 260  CONTINUE
      ICUS1=1
      WRITE(NTAPE,262)ITAPE,NTAPE,KTAPE1,KTAPE2,JTAPE1,ICUS1,J1,JTAPE2,J
     121,J22,JTAPE3,J31,J32,JTAPE4,J41,J42,JTAPE5,J5,NX,LTAPE1,ICUS1,J1,
     2LTAPE2,J21,J22,LTAPE3,J31,J32,LTAPE4,J41,J42,LTAPE5,J5,NX
 262  FORMAT(1H1,9X,52HTHE FOLLOWING UNIT USAGE IS ESTABLISHED FOR THIS
     1JOB,//,18X,4HUNIT,8X,4HUNIT,8X,17HNOTES ABOUT USAGE,/,18X,4HNAME,8
     2X,6HNUMBER,//,18X,5HITAPE,I9,10X,10HSTD. INPUT,/,18X,5HNTAPE,I9,10
     3X,11HSTD. OUTPUT,/,18X,6HKTAPE1,I8,10X,16HUSED IN THETA AS,/,18X,6
     4HKTAPE2,I8,10X,17HTEMPORARY STORAGE,/,18X,6HJTAPE1,I8,10X,25HDISTO
     5RT DATA FOR STATIONS,I3,5H THRU,I3,/,18X,6HJTAPE2,I8,35X,I3,I8,/,1
```

```
     68X,6HJTAPE3,I8,35X,I3,I8,/,18X,6HJTAPE4,I8,35X,I3,I8,/,18X,6HJTAPE
     75,I8,35X,I3,I8,/,18X,6HLTAPE1,I8,10X,25HTHETA DATA FOR STATIONS
     8I3,5H THRU,I3,/,18X,6HLTAPE2,I8,35X,I3,I8,/,18X,6HLTAPE3,I8,35X,I3
     9,I8,/,18X,6HLTAPE4,I8,35X,I3,I8,/,18X,6HLTAPE5,I8,35X,I3,I8,//,66X
     A,12H(ZEROS ABOVE,/,67X,13HINDICATE THAT ,/,67X,13HTHE UNIT WILL ,/
     B,67X,18HNOT BE REFERENCED))
      IF(J21.EQ.0)GO TO 2641
      IF(J31.EQ.0)GO TO 264
      IF(J41.EQ.0)GO TO 263
      REWIND JTAPE4
263   REWIND JTAPE3
264   REWIND JTAPE2
2641  REWIND JTAPE1
      REWIND JTAPE5
      ICUS2=0
      ICUS3=2
      ICUS4=0
      ICUS5=2
      IF(NTHETA.GT.1)ICUS5=1
      DO 700 N=1,NTHETA
      NUNIT=JTAPE1
      WRITE(NUNIT,265)(COMENT(K,N),K=1,12),NX,NLINES,NBETA,NTOPO,ICUS1,
     1IFSIMP,NPASS,IFBL,ICUS2,ICUS5,ICUS3,IFRPM,ITER,ICUS1,NPLOT,INCPO,NS
     2WRIT,INCWRI,ICUS4,ICUS4,IFTYPE,ICUS1,TOLCX,HUBLOC,GASK,CP,VIS1,VIS
     32,(BHP(J,N),TOCO(J,N),POCO(J,N),AFCO(J,N),J=1,NTOPO)
265   FORMAT(1X,12A6,/,1X,I5,4I6,/,1X,I5,4I6,/,1X,I5,10I6,/,1X,I5,/,1X,F
     17.4,2F8.4,F8.5,8X,2F8.4,/,(1X,F7.3,2F8.2,F8.4,16X))
      II1=2
      II2=J1
      ASSIGN 300 TO II3
      GO TO 500
300   IF(J21.EQ.0)GO TO 420
      WRITE(NUNIT,310)JTAPE2
310   FORMAT(I6)
      NUNIT=JTAPE2
      II1=J21
      II2=J22
      ASSIGN 325 TO II3
      GO TO 500
325   IF(J31.EQ.0)GO TO 420
      WRITE(NUNIT,310)JTAPE3
      NUNIT=JTAPE3
      II1=J31
      II2=J32
      ASSIGN 350 TO II3
      GO TO 500
350   IF(J41.EQ.0)GO TO 420
      WRITE(NUNIT,310)JTAPE4
      NUNIT=JTAPE4
      II1=J41
      II2=J42
      ASSIGN 420 TO II3
      GO TO 500
420   WRITE(NUNIT,310)JTAPE5
      NUNIT=JTAPE5
      II1=J5
      II2=NX
      ASSIGN 450 TO II3
      GO TO 500
450   WRITE(NUNIT,460)(DELM(J),J=1,NLINES)
```

```
460   FORMAT(1X,F7.4,8F8.4)
      WRITE(NUNIT,460)(BLAM(I),I=1,NX)
      WRITE(NUNIT,465)FLOWI,ESTFLO,WIDTH,(X(I),RHP(I),RSP(I),I=1,NX)
465   FORMAT(1X,F7.1,/,1X,F7.2,/,1X,F7.4,/,(1X,F7.4,8F8.4))
      WRITE(NUNIT,470)(RPM2(I),I=2,NX)
470   FORMAT(1X,F7.1,8F8.1)
      IF(IFTYPE.EQ.0)GO TO 700
      WRITE(NUNIT,475)(IMACHI(I),I=1,NX)
475   FORMAT(1X,I5,11I6)
      WRITE(NUNIT,460)(ANGLN(I),I=1,NX)
      GO TO 700
500   WRITE(NUNIT,505)II1,II2
505   FORMAT(1X,I5,I6)
      DO 680 I=II1,II2
      WRITE(NUNIT,510)IBETA2(I),IFTHIC(I),IFCAX(I),IFMACH(I),IFREYN(I),I
     1LOSS(I),IFMLOS(I),IFLVSI(I),IFPROF(I),IFREYL(I)
510   FORMAT(1X,I5,9I6)
      IF(IBETA2(I).EQ.6)GO TO 660
      IF(IFLVSI(I).EQ.1)WRITE(NUNIT,515)EXPLOS(I)
515   FORMAT(1X,F7.4)
      IF(IFMACH(I).EQ.3)WRITE(NUNIT,515)XMACH(I)
      WRITE(NUNIT,520)(BBP(I,J),STAG(I,J),BETA(I,J),SIGMA(I,J),BLAK(I,J)
     1,ZBAR(I,J),CLOSS(I,J),CO(I,J),J=1,NBETA)
520   FORMAT(8F8.4)
660   RCUS1=0.0
      RCUS2=1.0
      RCUS3=0.0
      DO 670 J=1,NLINES
      WRITE(NUNIT,665)RCUS1,RCUS2,RCUS3
665   FORMAT(9X,F7.4,24X,2F8.4)
670   RCUS1=RCUS1+1.0
680   CONTINUE
      GO TO II3,(300,325,350,420,450)
700   ICUS2=1
      GO TO 800
750   WRITE(NTAPE,770)
      STOP
770   FORMAT(63H1 JOB STOPPED BECAUSE INDICATOR OUT OF RANGE - SUBROUTIN
     1E INPUT)
800   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DISTOR (J1,J21,J22,J31,J32,J41,J42,J5,LTAPE1,LTAPE2,LTAS03*0000
     1PE3,LTAPE4,LTAPE5,NTAP,JTAPE1,JTAPE2,JTAPE3,JTAPE4,JTAPE5,NTHET)  S03*0001
      COMMON/DATA/CP1,CP2,GASK,HUBLOC,IFBL,IFCYL,IFRAD,IFRPM,IFSIMP,IFSTS03*0002
     1AT,IFTYPE,IGV,IMASS,INCPO,INCWRI,IPUNCH,IRAD,IROT,ITAPE,ITER,MTAPES03*0003
     2,NBETA,NCURVE,NLINES,NPASS,NPLOT,NPTS,NSTAGE,NTAPE,NTOPO,NWRIT,NX,S03*0004
     3TOLCX,VIS1,VIS2,ANGLN(30),BBP(30,15),BETA(30,15),BLAK(30,15),BLAM(S03*0005
     430),CLOSS(30,15),COMENT(12),CO(30,15),DELM(15),ESTFLO(1),EXPLOS(30S03*0006
     5),FLOWI(1,1),IBETA2(30),IFCAX(30),IFLVSI(30),IFMACH(30),IFMLOS(30)S03*0007
     6,IFPROF(30),IFREYL(30),IFREYN(30),IFTHIC(30),ILOSS(30),IMACHI(30),S03*0008
     7KSUMRY(1),RPM2(30),RHP(30),RSP(30),ROT(1),SIGMA(30,15),STAG(30,15)S03*0009
     8,WIDTH(1),X(30),XMACH(30),ZBAR(30,15),DELR(15)                    S03*0010
      COMMON/CALC/FLOW,ICASE,KRPM,L,NCASE,NDIV,NDUM,NMASS,NPLET,NROT,NWRS03*0011
     1ET,ALPHA(30,15),CMT(30,15),PO(30,15),RCB(30,15),STATT(30,15),TO(30S03*0012
     2,15),WMT(30,15)                                                   S03*0013
      COMMON/WENT/GJ,GJCP,GR2,CXM(30,15),Y(9)                           S03*0014
      COMMON/NEW/RDIS(30,15),XMU(30,15),DPDTH(30,15),DFDTH(30,15),ARATIOS03*0015
     1(30,15)                                                           S03*0016
      DIMENSION BLOCK(30)                                               S03*0017
      NTAPE=NTAP                                                        S03*0018
      PCALC=0.0                                                         S03*0019
      GJ=64.35*778.0                                                    S03*0020
    1 ICASE=1                                                           S03*0021
      NMASS=1                                                           S03*0022
      NROT=1                                                            S03*0023
    3 L=1                                                               S03*0024
      ISLOPE=0                                                          S03*0025
      WPREV=0.0                                                         S03*0026
      LL=0                                                              S03*0027
      IABORT=0                                                          S03*0028
      CALL READIN(IABORT,J1,J21,J22,J31,J32,J41,J42,J5,NTHET,JTAPE1,JTAPS03*0029
     1E2,JTAPE3,JTAPE4,JTAPE5)                                          S03*0030
      IF(IABORT.EQ.1)STOP                                               S03*0031
    4 CALL SOLVE(ISOLUT,BLOCK)                                          S03*0032
      NP=NPASS/3+1                                                      S03*0033
      DO 8 K=1,NP                                                       S03*0034
      KK=3*K                                                            S03*0035
      IF(L.EQ.KK)GO TO 12                                               S03*0036
    8 CONTINUE                                                          S03*0037
      GO TO 20                                                          S03*0038
   12 CALL SEARCH(ISOLUT,ISLOPE,LL,WPREV,PCALC)                         S03*0039
   20 IF(ABS(PCALC/FLOWI(NMASS,NROT)-1.0).GT.0.002)NDIV=2               S03*0040
      IF(NDIV.EQ.1.AND.IFBL.EQ.2.AND.(IFRPM.EQ.2.OR.IFRPM.EQ.3.OR.IFRPM.S03*0041
     1EQ.5.OR.IFRPM.EQ.6))WRITE(NTAPE,25)L,(I,BLOCK(I),I=1,NX)          S03*0042
   25 FORMAT(1H1,20X,75HRESULTS FROM ANNULUS WALL BOUNDARY LAYER CALCULAS03*0043
     1TIONS PERFORMED DURING PASS,I3,//,50X,18HSTATION   BLOCKAGE,//,(52S03*0044
     2X,I3,5X,F7.4))                                                    S03*0045
      IF((IFRPM.EQ.2.OR.IFRPM.EQ.3.OR.IFRPM.EQ.5.OR.IFRPM.EQ.6).AND.(L.ES03*0046
     1Q.NPASS.OR.L.EQ.NWRIT.OR.NDIV.EQ.1))CALL PRINT                    S03*0047
      IF(IGV.EQ.1.AND.(L.EQ.NPASS.OR.NDIV.EQ.1))CALL PUNCH(J1,J21,J22,J3S03*0048
     11,J32,J41,J42,J5,LTAPE1,LTAPE2,LTAPE3,LTAPE4,LTAPE5,NTHET)        S03*0049
      IF(L.EQ.NPLOT)NPLOT=NPLOT+INCPO                                   S03*0050
      L=L+1                                                             S03*0051
      IF(L.LE.NPASS.AND.NDIV.EQ.2)GO TO 4                               S03*0052
      RETURN                                                            S03*0053
      END                                                               S03*0054
```

```
      SUBROUTINE READIN(IABORT,J1,J21,J22,J31,J32,J41,J42,J5,NTHET,JTAPE
     11,JTAPE2,JTAPE3,JTAPE4,JTAPE5)
      COMMON/DATA/CP1,CP2,GASK,HUBLOC,IFBL,IFCYL,IFRAD,IFRPM,IFSIMP,IFST
     1AT,IFTYPE,IGV,IMASS,INCPO,INCWRI,IPUNCH,IRAD,IROT,ITAPE,ITER,MTAPE
     2,NBETA,NCURVE,NLINES,NPASS,NPLOT,NPTS,NSTAGE,NTAPE,NTOPO,NWRIT,NX,
     3TOLCX,VIS1,VIS2,ANGLN(30),BBP(30,15),BETA(30,15),BLAK(30,15),BLAM(
     430),CLOSS(30,15),COMENT(12),CO(30,15),DELM(15),ESTFLO(1),EXPLOS(30
     55),FLOWI(1,1),IBETA2(30),IFCAX(30),IFLVSI(30),IFMACH(30),IFMLOS(30)
     6,IFPROF(30),IFREYL(30),IFREYN(30),IFTHIC(30),ILOSS(30),IMACHI(30),
     7KSUMRY(1),RPM2(30),RHP(30),RSP(30),ROT(1),SIGMA(30,15),STAG(30,15)
     8,WIDTH(1),X(30),XMACH(30),ZBAR(30,15),DELR(15)
      COMMON/WENT/GJ,GJCP,GR2,CXM(30,15),Y(9)
      COMMON/CALC/FLOW,ICASE,KRPM,L,NCASE,NDIV,NDUM,NMASS,NPLET,NROT,NWR
     1ET,ALPHA(30,15),CMT(30,15),PO(30,15),RCB(30,15),STATT(30,15),TO(30
     2,15),WMT(30,15)
      COMMON/RESULT/DAMP,R(30,15)
      COMMON/NEW/RDIS(30,15),XMU(30,15),DPDTH(30,15),DFDTH(30,15),ARATIO
     1(30,15)
C*****READ IN INPUT DATA
      ITAPE=JTAPE1
      IF(NTHET.EQ.1)REWIND ITAPE
      READ(ITAPE,3)COMENT
C*****THIS CARD CONTAINS COMPRESSOR GEOMETRY SPECIFICATIONS
11    READ(ITAPE,100)NX,NLINES,NBETA,NTOPO,IMASS
      IROT=1
      NCASE=IMASS*IROT
C*****THIS CARD CONTAINS INDICATORS SPECIFYING THE COMPUTING METHOD
C                 TO BE USED
      READ(ITAPE,100)IFSIMP,NPASS,IFBL,NPTS,IGV
C*****THIS CARD CONTAINS INDICATORS SPECIFYING INPUT/OUTPUT OPTIONS
      READ(ITAPE,100)IFRAD,IFRPM,ITER,IFSUM,NPLOT,INCPO,NWRIT,INCWRI,IRA
     1D,IPUNCH,IFTYPE
C*****THIS CARD CONTAINS INDICATORS FOR THOSE CASES FOR WHICH COMPLETE
C         VELOCITY TRIANGLE DATA ARE DESIRED.  (WHEN IFSUM=2 THIS CARD
C         IS NOT REQUIRED).
      KSUMAX=IMASS*IROT
      GO TO(31,32),IFSUM
 31   READ(ITAPE,100)(KSUMRY(I),I=1,KSUMAX)
      GO TO 34
 32   DO 33 I=1,KSUMAX
 33   KSUMRY(I)=0
 34   NPLET=NPLOT
      NWRET=NWRIT
      READ (ITAPE,26)TOLCX,HUBLOC,GASK,CP1,CP2,VIS1,VIS2
      VIS1=VIS1*1.0E-6
      VIS2=VIS2*1.0E-6
      DO 341 J=1,NLINES
341   BBP(1,J)=0.0
      READ(ITAPE,35)(BBP(1,J),BLAK(1,J),XMU(1,J),ARATIO(1,J),DPDTH(1,J),
     1DFDTH(1,J),J=1,NTOPO)
35    FORMAT(4F8.4,F8.0,F8.4)
      DO 36 J=1,NLINES
36    R(1,J)=BBP(1,J)
      J21=0
      J22=0
      J31=0
      J32=0
      J41=0
      J42=0
38    FORMAT(2I6)
```

```
39    READ(ITAPE,38)III,II2
      IF(ITAPE.EQ.JTAPE1)J1=II2
      IF(ITAPE.EQ.JTAPE2)J21=II1
      IF(ITAPE.EQ.JTAPE2)J22=II2
      IF(ITAPE.EQ.JTAPE3)J31=II1
      IF(ITAPE.EQ.JTAPE3)J32=II2
      IF(ITAPE.EQ.JTAPE4)J41=II1
      IF(ITAPE.EQ.JTAPE4)J42=II2
      IF(ITAPE.EQ.JTAPE5)J5=II1
      DO 60 I=II1,II2
      READ(ITAPE,100)IBETA2(I),IFTHIC(I),IFCAX(I),IFMACH(I),IFREYN(I),
     1ILOSS(I),IFMLOS(I),IFLVSI(I),IFPROF(I),IFREYL(I)
      IF(IBETA2(I).EQ.1.OR.IBETA2(I).EQ.5)GO TO 45
      IF(IBETA2(I).EQ.6)GO TO 53
      LVS=IFLVSI(I)
      IFMA=IFMACH(I)
      IF(LVS.EQ.1)READ(ITAPE,26)EXPLOS(I)
      IF(IFMA.EQ.3)READ(ITAPE,26)XMACH(I)
 45   DO 50 J=1,NBETA
 50   READ(ITAPE,26)BBP(I,J),STAG(I,J),BETA(I,J),SIGMA(I,J),BLAK(I,J),
     1ZBAR(I,J),CLOSS(I,J),CO(I,J)
      READ(ITAPE,51)(R(I,J),RDIS(I,J),XMU(I,J),DPDTH(I,J),DFDTH(I,J),ARA
     1TIO(I,J),J=1,NLINES)
51    FORMAT(6F8.4)
C CLOSS - DCMDTHETA
      IF(IBETA2(I).NE.5)GO TO 60
      DO 52 J=1,NBETA
 52   ZBAR(I,J)=0.5
      GO TO 60
53    READ(ITAPE,54)(R(I,J),RDIS(I,J),XMU(I,J),DPDTH(I,J),DFDTH(I,J),ARA
     1TIO(I,J),BLAK(I,J),J=1,NLINES)
54    FORMAT(7F8.4)
60    CONTINUE
      IF(II2.EQ.NX)GO TO 68
      READ(ITAPE,61)NUNIT
61    FORMAT(I6)
      ITAPE=NUNIT
      IF(NTHET.EQ.1)REWIND ITAPE
      GO TO 39
68    CONTINUE
      READ(ITAPE,26)(DELR(J),J=1,NLINES)
 75   READ (ITAPE,26)(BLAM(I),I=1,NX)
 225  DO 226 N=1,IROT
 226  READ (ITAPE,26)(FLOWI(I,N),I=1,IMASS)
C FLOWI IS THE DESIRED STATIC-TO-TOTAL PRESSURE RATIO
      READ(ITAPE,26)(ESTFLO(I),I=1,IROT)
      READ(ITAPE,26)(WIDTH(I),I=1,IROT)
 234  READ (ITAPE,26)(X(I),RHP(I),RSP(I),I=1,NX)
      READ(ITAPE,26)(RPM2(I),I=2,NX)
273   IF(IFTYPE.EQ.0)GO TO 275
      READ(ITAPE,100)(IMACHI(I),I=1,NX)
      READ(ITAPE,26)(ANGLN(I),I=1,NX)
      GO TO 301
275   DO 277 I=1,NX
      ANGLN(I)=0.0
277   IMACHI(I)=0
C*****WRITE OUT INPUT DATA
301   IF(NX.GT.30.OR.NLINES.GT.15.OR.NBETA.GT.15)GO TO 5001
      NPOINT=(IFSIMP+1)/2
      GO TO(4000,4001,4002,4003,4004),NPOINT
```

```
4000  AA=0.1667
      GO TO 4005
4001  AA=0.02383
      GO TO 4005
4002  AA=0.007933
      GO TO 4005
4003  AA=0.00361
      GO TO 4005
4004  AA=0.00194
4005  DAMP=0.0
      NZM1=NX-1
      DO 4109 I=1,NZM1
      ASPECT=((RSP(I)-RHP(I))/(X(I+1)-X(I)+(TAN(ANGLN(I+1)/57.296))*
     1(RSP(I+1)+RHP(I+1))/2.-(TAN(ANGLN(I)/57.296))*(RSP(I)+RHP(I))/2.))
     2**2
      DAMP2=AA*ASPECT
      IF(DAMP2-DAMP)4109,4109,4108
4108  DAMP=DAMP2
4109  CONTINUE
      FLOW=ESTFLO(NROT)
      RPM=0.0
      PRATIO=0.0
      NPLOT=NPLET
      NWRIT=NWRET
      IF(IFRPM.GT.4)GO TO 565
      WRITE(NTAPE,300)
      WRITE (NTAPE,4)COMENT
      WRITE (NTAPE,307)
      WRITE(NTAPE,410)(BBP(1,J),BLAK(1,J),XMU(1,J),ARATIO(1,J),DPDTH(1,J
     1),DFDTH(1,J),J=1,NTOPO)
410   FORMAT(30X,4F12.5,F12.1,F12.5)
      DO 494 I=2,NX
      ILOS=ILOSS(I)
      IBET=IBETA2(I)
      IF(IBET.EQ.6)GO TO 429
      IF(IBET.NE.5)GO TO 4281
      ILOSS(I)=1
 4281 WRITE(NTAPE,310)I
      IF(IBET.EQ.1.AND.ILOS.NE.1)GO TO 5000
      GO TO(430,440,445,450,440),IBET
  429 WRITE(NTAPE,338)I
      DO 4291 J=1,NLINES
4291  WRITE(NTAPE,457)RDIS(I,J),XMU(I,J),DPDTH(I,J),DFDTH(I,J),ARATIO(I,
     1J),BLAK(I,J)
      GO TO 494
430   WRITE(NTAPE,309)
      DO 435 J=1,NBETA
 435  WRITE(NTAPE,28)BBP(I,J),STAG(I,J),CLOSS(I,J),CO(I,J)
      GO TO 493
440   WRITE(NTAPE,321)
      GO TO 454
445   WRITE(NTAPE,323)
      IABORT=1
      GO TO 454
450   WRITE(NTAPE,322)
 454  DO 455 J=1,NBETA
455   WRITE(NTAPE,29)BBP(I,J),STAG(I,J),BETA(I,J),SIGMA(I,J),BLAK(I,J),
     1ZBAR(I,J),CLOSS(I,J),CO(I,J)
493   WRITE(NTAPE,338)I
      DO 456 J=1,NLINES
```

```
456    WRITE(NTAPE,457)RDIS(I,J),XMU(I,J),DPDTH(I,J),DFDTH(I,J),ARATIO(I,
      1J)
457    FORMAT(30X,2F12.5,F12.1,3F12.5)
494    CONTINUE
       IF((IRAD.EQ.2.AND.ICASE.EQ.1).OR.(IRAD.EQ.0))GO TO 565
       WRITE(NTAPE,8)
       DO 563 I=1,NX
       WRITE(NTAPE,9)I,(J,J=1,NLINES)
563    WRITE(NTAPE,10)(R(I,J),J=1,NLINES)
565    CONTINUE
       GR=64.35*GASK
       GR2=GR/2.0
611    IF((IRAD.EQ.2.AND.ICASE.NE.1).OR.(IRAD.EQ.1))GO TO 5200
       NTUB=NLINES-1
       SBLOC=1.0-HUBLOC
       DELM(1)=RHP(1)**2
       DELM(2)=RSP(1)**2
       DELM(3)=DELM(2)-DELM(1)
       R(1,1)=SQRT(DELM(1)+BLAM(1)*HUBLOC*DELM(3))
       R(1,NLINES)=SQRT(DELM(2)-BLAM(1)*SBLOC*DELM(3))
       DAMP2=RSP(1)-RHP(1)
       ASPECT=R(1,1)**2
       AA=R(1,NLINES)**2-ASPECT
       DO 614 J=2,NTUB
       R(1,J)=RHP(1)+DELR(J)*DAMP2
614    DELM(J)=(R(1,J)**2-ASPECT)/AA
       DO 616 I=2,NX
       AA=RHP(I)**2
       DAMP2=RSP(I)**2
       ASPECT=DAMP2-AA
       R(I,1)=SQRT(AA+BLAM(I)*HUBLOC*ASPECT)
       R(I,NLINES)=SQRT(DAMP2-BLAM(I)*SBLOC*ASPECT)
       AA=R(I,1)**2
       DAMP2=R(I,NLINES)**2-AA
       DO 616 J=2,NTUB
616    R(I,J)=SQRT(AA+DELM(J)*DAMP2)
       GO TO 5200
 5000  WRITE(NTAPE,336)
       IABORT=1
       GO TO 5200
 5001  WRITE(NTAPE,337)
       IABORT=1
3      FORMAT(1X,12A6)
4      FORMAT(1H0,34X,12A6,///)
8      FORMAT(////,30X,65HTHE FOLLOWING INITIAL STREAMLINE ESTIMATES ARE
      1MADE FOR THIS CASE,///)
9      FORMAT(/30X62HSTREAMLINE NUMBERS AND RADII (TO 3 DECIMAL PLACES) F
      1OR STATIONI3,/3X,I3,14I8)
10     FORMAT(1X,15F8.3)
 26    FORMAT(9F8.4)
   27  FORMAT(30X7F12.4)
 28    FORMAT(40X,7F12.4)
 29    FORMAT(10X,8F12.4)
30     FORMAT(28X,6F12.4)
  100  FORMAT (12I6)
101    FORMAT(13I6)
300    FORMAT(1H1,40X,58HSECTOR CALCULATION FOR CIRCUMFERENTIAL DISTORTIO
      1N ANALYSIS)
307    FORMAT(///,59X,16HINLET CONDITIONS,//,35X,69HRADIUS     TOT.TEMP.
      1 TOT.PRESS.  WHIRL ANG.  DP/DTHETA   DPHI/DTHETA,/)
```

```
308   FORMAT(30X,6F12.5)                                                  S04*0240
309   FORMAT(46X,6HRADIUS6X,5HANGLE,3X,12HLOSS COEFF. ,1X,11HADD. LOSSESS04*0241
     1)                                                                   S04*0242
  310 FORMAT(///35X57HTHE BLADE EXIT ANGLES AND LOSS COEFFICIENTS FOR BLS04*0243
     1ADE ROWI4/35X42HARE INTERPOLATED FROM THE DATA GIVEN BELOW//)       S04*0244
 321  FORMAT(16X,6HRADIUS,5X,7HSTAGGER,6X,6HCAMBER,5X,8HSOLIDITY,3X,      S04*0245
     146HTHICKNESS   MAX CAM PT  LOSS COEFF ADD. LOSSES)                  S04*0246
 322  FORMAT(15X,6HRADIUS,3X,12HBLD INL ANGL,1X,11HBLD LV ANGL,2X,        S04*0247
     158HSOLIDITY   THICKNESS   MAX CAM PT  LOSS COEFF ADD  LOSSES)       S04*0248
323   FORMAT(1H0,15X,35HINVALID VALUE OF IBETA2 ENCOUNTERED)              S04*0249
 336  FORMAT(1H0,14X,89H*****ILOSS IS GREATER THAN UNITY WHEN IBETA2=1.   S04*0250
     1CALCULATION FOR THIS CASE TERMINATED*****)                          S04*0251
337   FORMAT(1H0,5X,66HDATA ERROR - NX GREATER THAN 30,OR NLINES OR NBETS04*0252
     1A GREATER THAN 15)                                                  S04*0253
338   FORMAT(///50X,10HAT STATION,I3,20H THE FOLLOWING APPLY,//,35X,70HRS04*0254
     1ADIUS    MU ANGLE    DP/DTHETA    DPHI/DTHETA    ARATIO    INT(DPDTHS04*0255
     2)DM,/)                                                              S04*0256
 5200 RETURN                                                              S04*0257
      END                                                                 S04*0258
```

```
      SUBROUTINE SOLVE(ISOLUT,BLOCK)
      COMMON/DATA/CP1,CP2,GASK,HUBLOC,IFBL,IFCYL,IFRAD,IFRPM,IFSIMP,IFST
     1AT,IFTYPE,IGV,IMASS,INCPO,INCWRI,IPUNCH,IRAD,IROT,ITAPE,ITER,MTAPE
     2,NBETA,NCURVE,NLINES,NPASS,NPLOT,NPTS,NSTAGE,NTAPE,NTOPO,NWRIT,NX,
     3TOLCX,VIS1,VIS2,ANGLN(30),BBP(30,15),BETA(30,15),BLAK(30,15),BLAM(
     430),CLOSS(30,15),COMENT(12),CO(30,15),DELM(15),ESTFLO(1),EXPLOS(30
     5),FLOWI(1,1),IBETA2(30),IFCAX(30),IFLVSI(30),IFMACH(30),IFMLOS(30)
     6,IFPROF(30),IFREYL(30),IFREYN(30),IFTHIC(30),ILOSS(30),IMACHI(30),
     7KSUMRY(1),RPM2(30),RHP(30),RSP(30),ROT(1),SIGMA(30,15),STAG(30,15)
     8,WIDTH(1),X(30),XMACH(30),ZBAR(30,15),DELR(15)
      COMMON/WENT/GJ,GJCP,GR2,CXM(30,15),Y(9)
      COMMON/CALC/FLOW,ICASE-KRPM,L,NCASE,NDIV,NDUM,NMASS,NPLET,NROT,NWR
     1ET,ALPHA(30,15),CMT(30,15),PO(30,15),RCB(30,15),STATT(30,15),TO(30
     2,15),WMT(30,15)
      COMMON/RESULT/DAMP,R(30,15)
      COMMON/TOGOSS/BET1(30,15),CLOS(30,15),CR(30,15),CUPONS(30,15),RM1(
     130,15),STAGG(30,15),STATP(30,15),TANBET(30,15)
      COMMON/NEW/RDIS(30,15),XMU(30,15),DPDTH(30,15),DFDTH(30,15),ARATIO
     1(30,15)
      DIMENSION F1(15),F2(15),F3(15),F4(15),BLOCK(30),DP(15),DF(15),XM(1
     15),XINT(15),CONFAC(15)
      DIMENSION RADFAC(3),U1VW1(15),AFA(15),CPUP(15),CPDN(15),CPMN(15)
     1,WATE(15),RM(15),W(15),P2RP2D(15),DPDR(15),DTDR(15),DTBDR(15),RX(1
     25),ARG(15),AX(15),BX(15),DX(15),VELFAC(3),RCAL(15),ALEPH(15),T1RT1
     3(15),TANA(15),U2(15),VFAC(15)
      EQUIVALENCE (RM(1),RCAL(1)),(AX(1),RX(1)),(BX(1),ARG(1))
      DO 20 J=1,NLINES
      RCB(1,J)=0.0
      RCB(NX,J)=0.0
      BHP1=R(1,J)
      CALL GRAPH3(BHP1,DP(J),BBP,DPDTH,NTOPO,1)
      CALL GRAPH3(BHP1,TO(1,J),BBP,BLAK,NTOPO,1)
      CALL GRAPH3(BHP1,PO(1,J),BBP,XMU,NTOPO,1)
      CALL GRAPH3(BHP1,DM,BBP,ARATIO,NTOPO,1)
      CALL GRAPH3(BHP1,DF(J),BBP,DFDTH,NTOPO,1)
      XM(J)=0.0
      CONFAC(J)=1.0
20    AFA(J)=TAN(DM/57.296)
      K=0
      NP=NPASS/3+1
      DO 21 J=1,NP
      KK=3*J
      IF(L.EQ.KK)K=1
21    CONTINUE
      IF(IFSIMP.EQ.2)GO TO 35
      NPOINT=IFSIMP+2
      DO 22 J=1,NLINES
22    CALL LSQLNE(X,R,J,NPOINT,NX,ALPHA,WMT,ANGLN,RCB,L)
      IF(L.GT.1)GO TO 40
23    WRITE(NTAPE,25)
25    FORMAT(1H1)
      DO 30 I=1,NX
      DO 30 J=1,NLINES
30    STATT(I,J)=TO(1,1)
      GO TO 40
35    IF(L.GT.1)GO TO 40
      DO 38 I=1,NX
      DO 38 J=1,NLINES
      ALPHA(I,J)=0.0
38    RCB(I,J)=0.0
```

```
      GO TO 23
40    IF(L.NE.NPLOT.OR.IFRPM.EQ.1.OR.IFRPM.EQ.4.OR.IFRPM.GE.7)GO TO 49
      DO 45 I=2,NX
      IF(IBETA2(I).EQ.1.OR.IBETA2(I).EQ.6)GO TO 45
      WRITE(NTAPE,46)
      WRITE(NTAPE,47)
      WRITE(NTAPE,48)
      GO TO 49
45    CONTINUE
46    FORMAT(1H1/118H *****************FOR THE INTERMEDIATE OUTPUT PRINT
     1ED BELOW, THE SYMBOLS HAVE THE FOLLOWING MEANINGS*****************
     2*//1X26HB1DES  -DESIGN INLET ANGLE33X27HBCHOKE -CHOKING INLET ANGL
     3E/1X28HBSTALL -STALLING INLET ANGLE31X53HB1DESE -EQUIVALENT TWO-DI
     4MENSIONAL DESIGN INLET ANGLE/1X52HB2DESE -EQUIVALENT TWO-DIMENSION
     5AL DESIGN EXIT ANGLE7X59HB2DAX  -DESIGN EXIT FLOW ANGLE CORRECTED 
     6FOR AXIAL VELOCITY/1X54HB2DES  -DESIGN EXIT FLOW ANGLE CORRECTED F
     7OR THICKNESS5X46HBE1    -EQUIVALENT TWO-DIMENSIONAL INLET ANGLE/1X
     845HBE2    -EQUIVALENT TWO-DIMENSIONAL EXIT ANGLE14X52HB2AXVL -EXIT
     9 FLOW ANGLE CORRECTED FOR AXIAL VELOCITY)
47    FORMAT(1X47HB2THIC -EXIT FLOW ANGLE CORRECTED FOR THICKNESS12X3HRM
     114X21H-RELETIVE MACH NUMBER/1X59HVMAX   -RATIO OF MAXIMUM SUCTION 
     2SURFACE VEL TO INLET VEL  28HCRITM  -CRITICAL MACH NUMBER/1X49HB2M
     3    -EXIT FLOW ANGLE CORRECTED FOR MACH NUMBER10X31HREYN.  -SECTIO
     4N REYNOLDS NUMBER/54H B2     -EXIT FLOW ANGLE CORRECTED FOR REYNOL
     5DS NUMBER6X31HDEQDES -DESIGN DITFUSION FACTOR/44H WDES   -UNCORREC
     6TED DESIGN LOSS COEFFICIENT16X57HWOPT   -DESIGN LOSS COEFFICIENT C
     7ORRECTED FOR MACH NUMBER2X/1X38HB1SCOR -CORRECTED STALLING INLET A
     8NGLE21X37HB1CCOR -CORRECTED CHOKING INLET ANGLE)
48    FORMAT(1X3HCLI4X41H-LOSS COEFFICIENT CORRECTED FOR INCIDENCE11X2HC
     1L5X47H-LOSS COEFFICIENT CORRECTED FOR REYNOLDS NUMBER//9X101H*****
     2NOTE - THE SIGNS OF THE FLOW ANGLES ARE SUCH THAT THE INLET FLOW A
     3NGLES ARE ALWAYS POSITIVE*****///)
49    W(1)=0.0
      NTUB=NLINES-1
      JM=NLINES/2
      JLINE=JM+1
      ISOLUT=0
      IF(L.NE.1)GO TO 70
      VEL=FLOW*GASK*TO(1,1)/(3.142*(R(1,NLINES)**2-R(1,1)**2)*PO(1,1))
      DO 50 J=1,NLINES
50    CMT(1,J)=VEL
70    DO 800 I=1,NX
      IFTAN=0
      IF(L.NE.1.OR.I.EQ.1)GO TO 80
      DO 75 J=1,NLINES
75    CMT(I,J)=(CMT(I-1,1)+CMT(I-1,NLINES))/2.0
80    IBE=IBETA2(I)
      IF(I.EQ.1)IBE=6
      IF(I.EQ.1)GO TO 200
      DO 83 J=1,NTUB
      RMID=(R(I,J)+R(I,J+1))/2.0
83    CALL GRAPH3(RMID,CONFAC(J),RDIS,ARATIO,NLINES,I)
      DO 85 J=1,NLINES
      BHP1=R(I,J)
      CALL GRAPH3(BHP1,DP(J),RDIS,DPDTH,NLINES,I)
      CALL GRAPH3(BHP1,DF(J),RDIS,DFDTH,NLINES,I)
85    CALL GRAPH3(BHP1,XM(J),RDIS,XMU,NLINES,I)
      IF(IBE.NE.6)GO TO 115
      DO 88 J=1,NLINES
      CALL GRAPH3(BHP1,XINT(J),RDIS,BLAK,NLINES,I)
```

```
      TO(I,J)=TO(I-1,J)
 88   PO(I,J)=PO(I-1,J)
      GO TO 200
115   CONTINUE
      DO 170 J=1,NLINES
      CPUP(J)=CP1+CP2*STATT(I-1,J)
      GAMMA=1.0/(1.0-GASK/(778.0*CPUP(J)))
      GF=2.0/(GAMMA-1.0)
      U1=RPM2(I)*3.14159/30.0*R(I-1,J)
      U2(J)=U1*R(I,J)/R(I-1,J)
      T1RT1(J)=1.0+CMT(I-1,J)**2*(1.0+(U1/CMT(I-1,J)-CXM(I-1,J))**2)/(GJ
     1*CPUP(J)*STATT(I-1,J))
      U1VW1(J)=CMT(I-1,J)*U1*CXM(I-1,J)
      RM1(I-1,J)=SQRT(GF*(T1RT1(J)-1.0))
      BET1(I-1,J)=ATAN(CXM(I-1,J)-U1/CMT(I-1,J))*57.296
      CALL INTERP(I,J,VF)
      VFAC(J)=VF
      SAMUP=CPUP(J)*778.0/GASK
      IF(IBE.EQ.1.OR.IBE.GE.5)GO TO 170
      IF(IFCAX(I).EQ.2)GO TO 170
      IFTAN=1
      DM=1.0
      IF(L.EQ.1)GO TO 168
      DM=CMT(I,J)/CMT(I-1,J)
      DM=DM*(1.0+VF*(1.0-DM)/DM)
      IF(DM.GT.1.25)DM=1.25
      IF(DM.LT.0.8)DM=0.8
168   TANA(J)=TANBET(I,J)/(1.0/DM+1.0)
170   P2RP20(J)=1.0-CLOS(I,J)*(1.0-T1RT1(J)**(-SAMUP))*(1.0+U1**2*((R(I,
     1J)/R(I-1,J))**2-1.0)/(GJ*CPUP(J)*T1RT1(J)*STATT(I-1,J)))**(-SAMUP)
200   NITER=15
      IF(L.EQ.1)NITER=10
      X4=TAN(ANGLN(I)/57.296)
      DO 210 J=1,NLINES
      WMT(I,J)=CMT(I,J)
      F1(J)=1.0/SQRT(1.0+(TAN(XM(J)))**2+X4**2)
      CBAR=X4*F1(J)
      F2(J)=F1(J)*COS(ALPHA(I,J))-CBAR*SIN(ALPHA(I,J))
      F3(J)=0.0
      IF(L.EQ.1.OR.I.EQ.1.OR.I.EQ.NX)GO TO 210
      X1=R(I,J)*X4
      X3=X(I+1)-X(I)+R(I+1,J)*TAN(ANGLN(I+1)/57.296)-X1
      X2=X(I)-X(I-1)+X1-R(I-1,J)*TAN(ANGLN(I-1)/57.296)
      DM=SQRT((1.0+((CXM(I,J)+CXM(I-1,J))/2.0)**2)*(1.0+((TAN(ALPHA(I,J)
     1)+TAN(ALPHA(I-1,J)))/2.0)**2))
      RMID=SQRT((1.0+((CXM(I+1,J)+CXM(I,J))/2.0)**2)*(1.0+((TAN(ALPHA(I+
     11,J))+TAN(ALPHA(I,J)))/2.0)**2))
      F3(J)=(CMT(I+1,J)-WMT(I-1,J))/(X3*RMID+X2*DM)*(F1(J)*SIN(ALPHA(I,J
     1))+CBAR*COS(ALPHA(I,J)))
210   F4(J)=F1(J)*TAN(XM(J))/R(I,J)*DP(J)
      DO 700 KK=1,NITER
      IISOL=0
      VLAST=CMT(I,1)
      DO 270 J=1,NLINES
      CPDN(J)=CP1+CP2*STATT(I,J)
      IF(I.EQ.1)GO TO 270
      IF(IBE.EQ.6)GO TO 230
      CPMN(J)=(CPUP(J)+CPDN(J))/2.0
      IF(IFTAN.EQ.0)GO TO 220
      DM=CMT(I,J)/CMT(I-1,J)
```

```
      DM=DM*(1.0+VFAC(J)*(1.0-DM)/DM)
      IF(DM.GT.1.25)DM=1.25
      IF(DM.LT.0.8)DM=0.8
      TANBET(I,J)=(TANA(J)*(1.0+1.0/DM)+TANBET(I,J))/2.0
220   AFA(J)=TANBET(I,J)+U2(J)/CMT(I,J)
      TO(I,J)=(U2(J)*CMT(I,J)*AFA(J)-U1VW1(J))/(GJ/2.0*CPMN(J))+TO(I-1,J
     1)
      PO(I,J)=PO(I-1,J)*(TO(I,J)/TO(I-1,J))**(778.0*CPMN(J)/GASK)*P2RP2D
     1(J)
      GO TO 270
230   AFA(J)=(CXM(I-1,J)*CMT(I-1,J)*R(I-1,J)-XINT(J))/(CMT(I,J)*R(I,J))
270   CONTINUE
      DO 280 J=1,NLINES
      RX(J)=R(I,J)
280   ARG(J)=PO(I,J)
      CALL RATE(RX,ARG,1,NLINES,DPDR)
      DO 300 J=1,NLINFS
300   ARG(J)=TO(I,J)
      CALL RATE(RX,ARG,1,NLINES,DTDR)
      CALL RATE(RX,AFA,1,NLINES,DTBDR)
      DO 400 J=1,NLINES
      AX(J)=-(F2(J)*(RCB(I,J)+AFA(J)/R(I,J)*DF(J))-F1(J)*(AFA(J)*DTBDR(J
     1)+AFA(J)**2/R(I,J))+F3(J)*SQRT(1.0+AFA(J)**2)/CMT(I,J))/(1.0+AFA(J
     2)**2)
      TRAT=1.0-CMT(I,J)**2*(1.0+AFA(J)**2)/(TO(I,J)*GJ*CPDN(J))
400   BX(J)=(F1(J)*(GJ/2.0*CPDN(J)*DTDR(J)-TRAT*(GJ/2.0*CPDN(J)*DTDR(J)-
     1GASK*32.175*TO(I,J)/PO(I,J)*DPDR(J)))-F4(J))/(1.0+AFA(J)**2)
      DX(JLINE)=CMT(I,JLINE)**2
      DO 450 J=JLINE,NTUB
      X1=AX(J+1)+AX(J)
      X3=BX(J+1)+BX(J)
      X4=(R(I,J+1)-R(I,J))/(F1(J)+F1(J+1))*2.0
      X2=1.0
      IF(X1.EQ.0.0)GO TO 450
      X2=-X1*X4
      IF(X2.GT.88.0)X2=88.0
      X2=EXP(X2)
      X4=(1.0-X2)/X1
450   DX(J+1)=DX(J)*X2+X3*X4
      DO 452 JJ=1,JM
      J=JLINE-JJ
      X1=AX(J+1)+AX(J)
      X3=BX(J+1)+BX(J)
      X4=(R(I,J)-R(I,J+1))/(F1(J)+F1(J+1))*2.0
      X2=1.0
      IF(X1.EQ.0.0)GO TO 452
      X2=-X1*X4
      IF(X2.GT.88.0)X2=88.0
      X2=EXP(X2)
      X4=(1.0-X2)/X1
452   DX(J)=DX(J+1)*X2+X3*X4
      DO 550 J=1,NLINES
      IF(DX(J).GE.1.0)GO TO 465
      DX(J)=1.0
      IISOL=1
      IF(ISOLUT.EQ.0.AND.K.EQ.1.AND.KSUMRY(ICASE).EQ.1)WRITE(NTAPE,455)I
     1,J,L,KK
455   FORMAT(1X,33HVELOCITY VALUE LIMITED AT STATIONI3,11H STREAMLINEI3,
     15H PASSI3,16H CONTINUITY LOOPI3)
465   DX(J)=SQRT(DX(J))*0.25+CMT(I,J)*0.75
```

```
      IF(DX(J)/CMT(I,J).LE.1.4)GO TO 480
      DX(J)=CMT(I,J)*1.4
      IF(KSUMRY(ICASE).EQ.1.AND.KK.GE.14)WRITE(NTAPE,475)I,J,L,KK
475   FORMAT(1X,46HVELOCITY CHANGE BY MOMENTUM LIMITED AT STATIONI3,11H
     1STREAMLINEI3,5H PASSI3,16H CONTINUITY LOOPI3)
480   IF(DX(J)/CMT(I,J).GT.0.6)GO TO 490
      DX(J)=CMT(I,J)*0.6
      IF(KSUMRY(ICASE).EQ.1.AND.KK.GE.14)WRITE(NTAPE,475)I,J,L,KK
490   IF(I.NE.1.AND.IBE.EQ.6)GO TO 491
      IF(I.EQ.1)GO TO 4905
      AFF=TANBET(I,J)+U2(J)/DX(J)
      IF(RPM2(I).EQ.0.0)GO TO 492
      TO(I,J)=(U2(J)*DX(J)*AFF-U1VW1(J))/(GJ/2.0*CPMN(J))+TO(I-1,J)
      PO(I,J)=PO(I-1,J)*(TO(I,J)/TO(I-1,J))**(778.0*CPMN(J)/GASK)*P2RP2D
     1(J)
      GO TO 492
4905  AFF=AFA(J)
      TANBET(I,J)=AFF
      GO TO 492
491   AFF=(CMT(I-1,J)*R(I-1,J)*CXM(I-1,J)-XINT(J))/(DX(J)*R(I,J))
      TANBET(I,J)=0.0
492   STATT(I,J)=TO(I,J)-DX(J)**2*(1.0+AFF**2)/(GJ*CPDN(J))
      IF(STATT(I,J).GE.0.001)GO TO 495
      STATT(I,J)=0.001
      IF(KSUMRY(ICASE).EQ.1.AND.KK.GE.14)WRITE(NTAPE,493)I,J,L,KK
493   FORMAT(1X,40HSTATIC TEMPERATURE RESTRAINED AT STATIONI3,11H STREAM
     1LINEI3,5H PASS I3,16H CONTINUITY LOOPI3)
495   GAMMA=1.0/(1.0-GASK/(778.0*CPDN(J)))
      STATP(I,J)=PO(I,J)*(STATT(I,J)/TO(I,J))**(GAMMA/(GAMMA-1.0))
      WATE(J)=STATP(I,J)/(GASK*STATT(I,J))
550   RM(J)=DX(J)/SQRT((GAMMA*32.175*GASK*STATT(I,J))/(1.0+TANBET(I,J)**
     12))
      WR=0.0
      DO 600 J=1,NTUB
      W(J+1)=W(J)+(DX(J)*COS(ALPHA(I,J))+DX(J+1)*COS(ALPHA(I,J+1)))*(WAT
     1E(J+1)+WATE(J))*(R(I,J+1)**2-R(I,J)**2)*0.7853975*CONFAC(J)
600   WR=WR+(RM(J+1)+RM(J))*(W(J+1)-W(J))/2.0
      WR=WR/W(NLINES)
      IF((IMACHI(I).EQ.0.AND.WR.LT.1.0).OR.(IMACHI(I).EQ.1.AND.WR.GE.1.0
     1))GO TO 630
      GF=(GAMMA-1.0)/2.0
      WR2=WR**2
      RATIO=SQRT((1.0+GF*WR2)/(WR2**2*(1.0+GF/WR2)))
      IF(ISOLUT.EQ.0.AND.K.EQ.1.AND.KSUMRY(ICASE).EQ.1)WRITE(NTAPE,610)I
     1,L,KK
610   FORMAT(1X,39HMACH NUMBER SWITCH PERFORMED AT STATIONI3,5H PASSI3,1
     16H CONTINUITY LOOPI3)
      GO TO 680
630   WRX=1.0-WR**2
      IF(WRX.EQ.0.0)WRX=1.0
      IF(ABS(WRX).GT.0.1)GO TO 640
      WRX=0.1*SGIN(1.0-WR)
640   WRAT=FLOW/W(NLINES)
      RATIO=WRAT**(1.0/WRX)
      IF(RATIO.GT.1.2)GO TO 660
      IF(RATIO.LT.0.8)GO TO 670
      GO TO 680
660   RATIO =1.2
      IF(KSUMRY(ICASE).EQ.1.AND.KK.GE.14)WRITE(NTAPE,665)I,L,KK
      GO TO 680
```

```
665   FORMAT(1X,42HVELOCITY CHANGE BY FLOW LIMITED AT STATIONI3, 5H PASSS05*0300
     1I3,16H CONTINUITY LOOPI3)                                         S05*0301
670   RATIO=0.8                                                         S05*0302
      IF(KSUMRY(ICASE).EQ.1.AND.KK.GE.14)WRITE(NTAPE,665)I,L,KK         S05*0303
680   DO 690 J=1,NLINES                                                 S05*0304
      CMT(I,J)=DX(J)*RATIO                                              S05*0305
      IF(KSUMRY(ICASE).EQ.1.AND.ITER.EQ.1)WRITE(NTAPE,695)I,J,L,KK,CMT(IS05*0306
     1,J)                                                               S05*0307
690   CONTINUE                                                          S05*0308
      IF(IFBL.EQ.2.AND.L.GT.1)CALL BLTHIC(I,BLOCK)                      S05*0309
      VRAT=CMT(I,1)/VLAST                                               S05*0310
      IF(ABS(WRAT-1.0).LE.TOLCX.AND.ABS(VRAT-1.0).LE.0.002)GO TO 705    S05*0311
695   FORMAT(20X,7HSTATIONI3,11H STREAMLINEI3,5H PASSI3,16H CONTINUITY LS05*0312
     1OOPI3,9H VELOCITYF9.2)                                            S05*0313
698   FORMAT(10X,48HCONTINUITY AND MOMENTUM NOT SATISFIED AT STATIONI3,5S05*0314
     1H PASSI3,17H RATIO OF FLOWS =F7.4,22H RATIO OF VELOCITIES =F7.4)  S05*0315
700   CONTINUE                                                          S05*0316
      IF(WRAT.LT.1.0)GO TO 704                                          S05*0317
      GF=(GAMMA-1.0)/2.0                                                S05*0318
      WMXRAT=W(NLINES)/(WR*FLOW)*((1.0+GF*WR**2)/(1.0+GF))**((GAMMA+1.0)S05*0319
     1/(2.0*(GAMMA-1.0)))                                               S05*0320
      IF(WMXRAT.GT.1.0)GO TO 704                                        S05*0321
      IF(ISOLUT.EQ.0.AND.K.EQ.1.AND.KSUMRY(ICASE).EQ.1)WRITE(NTAPE,702)IS05*0322
     1CASE,L,I,WMXRAT                                                   S05*0323
702   FORMAT(/,10X,4HCASE,I3,27H ASSUMED CHOKED DURING PASS,I3,11H AT STS05*0324
     1ATION,I3,42H RATIO OF CHOKING FLOW TO SPECIFIED FLOW =,F7.4)      S05*0325
      IF(IISOL.EQ.0)IISOL=2                                             S05*0326
704   IF(ISOLUT.EQ.0.AND.K.EQ.1.AND.KSUMRY(ICASE).EQ.1)WRITE(NTAPE,698)IS05*0327
     1,L,WRAT,VRAT                                                      S05*0328
705   IF(ISOLUT.EQ.0)ISOLUT=IISOL                                       S05*0329
      DELM(1)=0.0                                                       S05*0330
      DO 707 J=2,NLINES                                                 S05*0331
      W(J)=W(J)/W(NLINES)                                               S05*0332
      IF(I.EQ.1)DELM(J)=W(J)                                            S05*0333
707   CONTINUE                                                          S05*0334
      IF(L.EQ.1)GO TO 770                                               S05*0335
      IF(I.EQ.1)GO TO 770                                               S05*0336
      DO 760 J=2,NTUB                                                   S05*0337
      DO 750 JJ=2,NLINES                                                S05*0338
      IF(DELM(J)-W(JJ))710,720,750                                      S05*0339
710   RCAL(J)=R(I,JJ-1)+(R(I,JJ)-R(I,JJ-1))*(DELM(J)-W(JJ-1))/(W(JJ)-W(JS05*0340
     1J-1))                                                             S05*0341
      GO TO 760                                                         S05*0342
720   RCAL(J)=R(I,JJ)                                                   S05*0343
      GO TO 760                                                         S05*0344
750   CONTINUE                                                          S05*0345
760   CONTINUE                                                          S05*0346
      DO 780 J=2,NTUB                                                   S05*0347
780   R(I,J)=(DAMP*R(I,J)+RCAL(J))/(1.0+DAMP)                           S05*0348
770   DO 800 J=1,NLINES                                                 S05*0349
800   CXM(I,J)=AFA(J)                                                   S05*0350
      NDIV=1                                                            S05*0351
      DO 900 I=1,NX                                                     S05*0352
      DO 900 J=1,NLINES,NTUB                                            S05*0353
      IF( ABS(CMT(I,J)/WMT(I,J)-1.0).GT.0.005)NDIV=2                    S05*0354
900   CONTINUE                                                          S05*0355
      RETURN                                                            S05*0356
      END                                                               S05*0357
```

```
      SUBROUTINE GRAPH3(XIN,YOUT,X,Y,NPOINT,I)
      DIMENSION X(30,15),Y(30,15)
      IF(NPOINT.EQ.1)GO TO 30
      K=NPOINT
      IF(XIN.GE.X(I,K-1))GO TO 20
      K=2
10    IF(XIN.LE.X(I,K))GO TO 20
      K=K+1
      GO TO 10
20    YOUT=Y(I,K-1)+(XIN-X(I,K-1))/(X(I,K)-X(I,K-1))*(Y(I,K)-Y(I,K-1))
      RETURN
30    YOUT=Y(I,1)
      RETURN
      END
```

```
      SUBROUTINE LSQLNE(EY,Y,JAY,NPTS,NEX,DYDX,D2YDX2,ANGLN,RCB,LL)
      DIMENSION Y(30,15),EX(30),DYDX(30,15),D2YDX2(30,15),U(15),V(15),AC
     1O(3),EY(1),ANGLN(1),RCB(30,15)
      J=JAY
      NX=NEX
      DO 10 I=1,NX
10    EX(I)=TAN(ANGLN(I)/57.296)*Y(I,J)+EY(I)
      NX1=NX-1
      I1=(NPTS+1)/2
      I2=NX-(NPTS-1)
      MID=I1
      MID1=MID-1
C     ***** DETERMINE HORIZONTAL PORTIONS OF Y VS. EX CURVE. *****
      DO 100 I=1,NX1
      YAV=(Y(I+1,J)+Y(I,J))/2.0
      DELTY=ABS((Y(I+1,J)-Y(I,J))/(YAV*(EX(I+1)-EX(I))))
      IF (DELTY-0.00001)50,50,120
   50 DYDX(I,J)=0.0
      D2YDX2(I,J)=0.0
  100 CONTINUE
      DYDX(NX,J)=0.0
      D2YDX2(NX,J)=0.0
      GO TO 810
  120 K1=I
      KEND=NX-K1
      DO 160 K=1,KEND
      KNX=NX-K+1
      YAV=(Y(KNX,J)+Y(KNX-1,J))/2.0
      DELTY=ABS((Y(KNX,J)-Y(KNX-1,J))/(YAV*(EX(KNX)-EX(KNX-1))))
      IF (DELTY-0.00001)150,150,200
  150 DYDX(KNX,J)=0.0
      D2YDX2(KNX,J)=0.0
  160 CONTINUE
C     *****LAST STATION FROM INLET WHICH IS IN A HORIZONTAL SECTION OF
C     THE CURVE IS K1.  LAST STATION FROM OUTLET WHICH IS IN A
C     HORIZONTAL SECTION IS KNX. *****
C     ***** COMPARE VALUES OF K1 AND KNX WITH VALUES OF I1 AND I2. *****
  200 IF (K1-I2)210,300,300
  210 IF (KNX-I1)330,220,220
  220 IF (K1-I1)230,240,240
 230  IK1=1
      GO TO 250
  240 IK1=2
  250 IF (KNX-I2)260,260,270
  260 GO TO (360,390),IK1
  270 GO TO (420,460),IK1
C     ***** USE FITTING PROCEDURE 2 ONLY.  Y VS. EX CURVE HORIZONTAL
C     FOR ALL STATIONS BUT THOSE WHOSE INDICES ARE GREATER THAN I2. ****
  300 LE1=K1
      IF (KNX-NX)305,310,310
  305 LE2=KNX
      GO TO 315
  310 LE2=NX1
  315 INORM=2
      GO TO 600
C     ***** USE FITTING PROCEDURE 1 ONLY.  Y VS. EX CURVE HORIZONTAL
C     FOR ALL STATIONS BUT THOSE WHOSE INDICES ARE LESS THAN I2. *****
  330 IF (K1-1)335,335,340
  335 LB1=2
      GO TO 345
```

```
  340 LB1=K1
  345 LB2=KNX
      IND=2
      INORM=2
      GO TO 500
C     ***** USE FITTING PROCEDURES 1 AND 3.  STATIONS WHOSE INDICES
C     ARE GREATER THAN I2 ARE IN A HORIZONTAL SECTION OF THE Y VS. EX
C     CURVE. *****
  360 IF (K1-1)365,365,370
  365 LB1=2
      GO TO 375
  370 LB1=K1
  375 LB2=I1-1
      IND=2
      INORM=1
      IIN=I1
      IFIN=KNX
      GO TO 500
C     ***** USE FITTING PROCEDURE 3 ONLY.  STATIONS WHOSE INDICES ARE
C     GREATER THAN I2 OR LESS THAN I1 ARE IN HORIZONTAL SECTIONS OF THE
C     Y VS. EX CURVE. *****
  390 IIN=K1
      IFIN=KNX
      GO TO 700
C     *****USE FITTING PROCEDURES 1, 2, AND 3. *****
  420 IF (K1-1)425,425,430
  425 LB1=2
      GO TO 435
  430 LB1=K1
  435 LB2=I1-1
      IND=1
      LE1=I2+1
      IF (KNX-NX)440,445,445
  440 LE2=KNX
      GO TO 450
  445 LE2=NX1
  450 INORM=1
      IIN=I1
      IFIN=I2
      GO TO 500
C     ***** USE FITTING PROCEDURES 2 AND 3.  STATIONS WHOSE INDICES ARE
C     LESS THAN I1 ARE IN A HORIZONTAL SECTION OF THE Y VS. EX CURVE. **
  460 LE1=I2+1
      IF (KNX-NX)465,470,470
  465 LE2=KNX
      GO TO 475
  470 LE2=NX1
  475 INORM=1
      IIN=I1
      IFIN=I2
      GO TO 600
C     *****FITTING PROCEDURE 1 - USED FOR STATIONS WHOSE INDICES ARE
C     LESS THAN I1. *****
  500 DO 530 LM=LB1,LB2
      NDP=I1-LM
      NDP1=NDP+1
      K=NDP1
      DO 510 L=1,NDP
      U(L)=Y(1,J)
      V(L)=EX(1)-(EX(K)-EX(1))
```

```
      K=K-1
  510 CONTINUE
      K=1
      DO 520 L=NDP1,NPTS
      U(L)=Y(K,J)
      V(L)=EX(K)
      K=K+1
  520 CONTINUE
      CALL LSQFIT (U,V,NPTS,2,ACO)
      B=2.0*ACO(3)
      DYDX(LM,J)=ACO(2)+B*EX(LM)
      D2YDX2(LM,J)=B
  530 CONTINUE
      GO TO (600,540),IND
  540 GO TO (700,750),INORM
C     *****FITTING PROCEDURE 2 - USED FOR STATIONS WHOSE INDICES ARE
C     GREATER THAN I2. *****
  600 DO 630 LM=LE1,LE2
      NDP=LM-I2
      NNP=NPTS-NDP
      NNP1=NNP+1
      K=LM-MID1
      DO 610 L=1,NNP
      U(L)=Y(K,J)
      V(L)=EX(K)
      K=K+1
  610 CONTINUE
      NXN=NX
      DO 620 L=NNP1,NPTS
      NXN=NXN-1
      U(L)=Y(NX,J)
      V(L)=EX(NX)+(EX(NX)-EX(NXN))
  620 CONTINUE
      CALL LSQFIT (U,V,NPTS,2,ACO)
      B=2.0*ACO(3)
      DYDX(LM,J)=ACO(2)+B*EX(LM)
      D2YDX2(LM,J)=B
  630 CONTINUE
      GO TO (700,750),INORM
C     *****FITTING PROCEDURE 3 - USED FOR STATIONS WHOSE INDICES ARE
C     BETWEEN I1 AND I2. *****
  700 DO 730 LM=IIN,IFIN
      LMMID=LM-MID
      DO 720 L=1,NPTS
      K=LMMID+L
      U(L)=Y(K,J)
  720 V(L)=EX(K)
      CALL LSQFIT (U,V,NPTS,2,ACO)
      B=2.0*ACO(3)
      DYDX(LM,J)=ACO(2)+B*EX(LM)
      D2YDX2(LM,J)=B
  730 CONTINUE
C     CHECK POSITION OF K1 AND KNX
  750 IF (K1-1)760,760,770
  760 DYDX(1,J)=(Y(2,J)-Y(1,J))/(EX(2)-EX(1))
      D2YDX2(1,J)=0.0
      GO TO 780
  770 DYDX(K1,J)=0.0
  780 IF (KNX-NX)800,790,790
  790 DYDX(NX,J)=(Y(NX,J)-Y(NX1,J))/(EX(NX)-EX(NX1))
```

```
      D2YDX2(NX,J)=0.0                                              S07*0180
      GO TO 810                                                     S07*0181
  800 DYDX(KNX,J)=0.0                                               S07*0182
810   DO 850 I=1,NX                                                 S07*0183
      D2YDX2(I,J)=D2YDX2(I,J)/(1.0+DYDX(I,J)**2)**1.5               S07*0184
850   DYDX(I,J)=ATAN(DYDX(I,J))                                     S07*0185
      IF(LL.EQ.1)GO TO 900                                          S07*0186
      DO 870 I=2,NX1                                                S07*0187
870   RCB(I,J)=(RCB(I,J)+D2YDX2(I,J))/2.0                           S07*0188
      RETURN                                                        S07*0189
900   DO 950 I=2,NX1                                                S07*0190
950   RCB(I,J)=D2YDX2(I,J)                                          S07*0191
      RETURN                                                        S07*0192
      END                                                           S07*0193
```

```
      SUBROUTINE LSQFIT (Y,X,NPTS,IFR,A)
C           EVALUATES THE COEFFICIENTS OF A LEAST-SQUARES PARABOLA
C     FOR THE Y VERSUS X CURVE
      DIMENSION X(11),Y(11),SUM(10),V(10),A(3),B(10,10)
      N=NPTS
      M=2
      DO 90 I=1,3
   90 A(I)=0.0
      GO TO (50,2),IFR
C     DETERMINE WHETHER OR NOT Y IS CONSTANT
   50 N1=N-1
      DO 60 K=1,N1
      YAV=(Y(K+1)+Y(K))/2.0
      IF(ABS(YAV)-0.00001)60,60,51
 51   DELTY=ABS((Y(K+1)-Y(K))/YAV)
      IF (DELTY-0.00001)60,60,2
   60 CONTINUE
      A(1)=Y(1)
      GO TO 41
    2 LS=2*M+1
      LB=M+2
      LV=M+1
      DO 5 J=2,LS
    5 SUM(J)=0.0
      SUM(1)=N
      DO 6 J=1,LV
    6 V(J)=0.0
      DO 16 I=1,N
      P=1.0
      V(1)=V(1)+Y(I)
      DO 13 J=2,LV
      P=X(I)*P
      SUM(J)=SUM(J)+P
   13 V(J)=V(J)+Y(I)*P
      DO 16 J=LB,LS
      P=X(I)*P
   16 SUM(J)=SUM(J)+P
   17 DO 20 I=1,LV
      DO 20 K=1,LV
      J=K+I
   20 B(K,I)=SUM(J-1)
      DO 22 K=1,LV
   22 B(K,LB)=V(K)
   23 DO 31 L=1,LV
      DIVB=B(L,L)
      DO 26 J=L,LB
   26 B(L,J)=B(L,J)/DIVB
      I1=L+1
      IF (I1-LB) 28,33,33
   28 DO 31 I=I1,LV
      FMULTB=B(I,L)
      DO 31 J=L,LB
   31 B(I,J)=B(I,J)-B(L,J)*FMULTB
   33 A(LV)=B(LV,LB)
      I=LV
   35 SIGMA =0.0
      DO 37 J=I,LV
   37 SIGMA=SIGMA+B(I-1,J)*A(J)
      I=I-1
      A(I)=B(I,LB)-SIGMA
```

```
40 IF (I-1)41,41,35                                   S08*0060
41 RETURN                                             S08*0061
   END                                                S08*0062
```

```
      SUBROUTINE SEARCH(ISOLUT,ISLOPE,LL,WPREV,PCALC)
      COMMON/DATA/CP1,CP2,GASK,HUBLOC,IFBL,IFCYL,IFRAD,IFRPM,IFSIMP,IFSTS
     1AT,IFTYPE,IGV,IMASS,INCPO,INCWRI,IPUNCH,IRAD,IROT,ITAPE,ITER,MTAPES
     2,NBETA,NCURVE,NLINES,NPASS,NPLOT,NPTS,NSTAGE,NTAPE,NTOPO,NWRIT,NX,
     3TOLCX,VIS1,VIS2,ANGLN(30),BBP(30,15),BETA(30,15),BLAK(30,15),BLAM(
     430),CLOSS(30,15),COMENT(12),CO(30,15),DELM(15),ESTFLO(1),EXPLOS(30
     5),FLOWI(1,1),IBETA2(30),IFCAX(30),IFLVSI(30),IFMACH(30),IFMLOS(30)
     6,IFPROF(30),IFREYL(30),IFREYN(30),IFTHIC(30),ILOSS(30),IMACHI(30),
     7KSUMRY(1),RPM2(30),RHP(30),RSP(30),ROT(1),SIGMA(30,15),STAG(30,15)
     8,WIDTH(1),X(30),XMACH(30),ZBAR(30,15),DELR(15)
      COMMON/CALC/FLOW,ICASE,KRPM,L,NCASE,NDIV,NDUM,NMASS,NPLET,NROT,NWRS
     1ET,ALPHA(30,15),CMT(30,15),PO(30,15),RCB(30,15),STATT(30,15),TO(30
     2,15),WMT(30,15)
      COMMON/TOGOSS/BET1(30,15),CLOS(30,15),CR(30,15),CUPONS(30,15),RM1(
     130,15),STAGG(30,15),STATP(30,15),TANBET(30,15)
      IF(L.EQ.3)PCALC=0.0
      ISOLUT=ISOLUT+1
      STEP=WIDTH(NROT)*0.45*FLOW
      GO TO(20,220,200),ISOLUT
   20 CONTINUE
      IF(L.GT.3)PPREV=PCALC
      PCALC=0.0
      NTUB=NLINES-1
      DO 26 J=1,NTUB
   26 PCALC=PCALC+(DELM(J+1)-DELM(J))*(STATP(NX,J+1)+STATP(NX,J))/2.0
      IF(LL.EQ.0.AND.ABS(PCALC/FLOWI(1,1)-1.0).LE.0.002)GO TO 210
      IF(PCALC.LT.FLOWI(NMASS,NROT))GO TO 100
      IF(LL.NE.0)GO TO 60
      LL=1
   40 CONTINUE
      WPREV=BIG(WPREV,FLOW)
      FLOW=WPREV+STEP
      GO TO 400
   60 CONTINUE
      LL=LL+1
      IF(FLOW.EQ.WPREV)GO TO 80
      SLOPE=(PCALC-PPREV)/(FLOW-WPREV)
      IF(SLOPE)70,80,40
   70 CONTINUE
      WTEMP=FLOW
      FLOW=FLOW+(FLOW-WPREV)*(FLOWI(NMASS,NROT)-PCALC)/(PCALC-PPREV)
      WX=BIG(WTEMP,WPREV)
      IF(FLOW.GT.WX*(1.0+WIDTH(NROT)))FLOW=WX*(1.0+WIDTH(NROT))
      IF(FLOW.LT.WTEMP*0.7)FLOW=WTEMP*0.7
      WPREV=WTEMP
      GO TO 400
   80 CONTINUE
      WTEMP=FLOW
      FLOW=(WPREV+FLOW)/2.0
      WPREV=BIG(WTEMP,WPREV)
      GO TO 400
  100 CONTINUE
      IF(LL.NE.0)GO TO 120
      LL=1
      WPREV=FLOW
  110 CONTINUE
      FLOW=FLOW-STEP
      GO TO 400
  120 CONTINUE
      LL=LL+1
```

```
      IF(FLOW.EQ.WPREV)GO TO 80
      SLOPE=(PCALC-PPREV)/(FLOW-WPREV)
      IF(SLOPE)140,80,170
140   CONTINUE
      ISLOPE=1
      GO TO 70
170   IF(ISLOPE.EQ.0)GO TO 40
      WTEMP=FLOW
      FLOW=(WPREV+FLOW)/2.0
      WPREV=WTEMP
      GO TO 400
200   CONTINUE
      IF(LL.EQ.0)GO TO 110
      FLOW=WPREV+(FLOW-WPREV)/2.0
      GO TO 400
210   LL=LL+1
      WPREV=FLOW
      GO TO 400
220   CONTINUE
      IF(LL.EQ.0)GO TO 40
      FLOW=WPREV-(WPREV-FLOW)/2.0
400   CONTINUE
      ESTFLO(NROT)=FLOW
      WRITE(NTAPE,420)L,FLOW,PCALC
420   FORMAT(/,10X,4HPASS,I3,7H FLOW =,F7.2,8H PCALC =,F9.1)
      RETURN
      END
```

```
      FUNCTION BIG(X,Y)                                                 S10*0000
      BIG=X                                                             S10*0001
      IF(X.LT.Y)BIG=Y                                                   S10*0002
      RETURN                                                            S10*0003
      END                                                               S10*0004
```

```
      FUNCTION SGIN(X)                                                  S11*0000
      SGIN=1.0                                                          S11*0001
      IF(X.LT.0.0)SGIN=-1.0                                             S11*0002
      RETURN                                                            S11*0003
      END                                                               S11*0004
```

```
      SUBROUTINE RATE(X,Y,IS,IE,DYDX)
      DIMENSION X(1),Y(1),DYDX(1)
      IS=IS
      IE=IE
      IF(IS-IE)8,5,8
    5 DYDX(IS)=0.0
      GO TO 60
    8 DO 50 I=IS,IE
      IF(I-IS) 10,10,20
   10 DX1=X(IS+1)-X(IS)
      DY1=Y(IS+1)-Y(IS)
      DYDX(IS)=DY1/DX1
      GO TO 50
   20 IF(I-IE)30,40,60
   30 DYDX(I)=((Y(I+1)-Y(I))/(X(I+1)-X(I))+(Y(I)-Y(I-1))/(X(I)-X(I-1)))/
     12.0
      GO TO 50
   40 DXIE=X(IE)-X(IE-1)
      DYIE=Y(IE)-Y(IE-1)
      DYDX(IE)=DYIE/DXIE
   50 CONTINUE
   60 RETURN
      END
```

```
      SUBROUTINE INTERP(I,J,VF)
      COMMON/DATA/CP1,CP2,GASK,HUBLOC,IFBL,IFCYL,IFRAD,IFRPM,IFSIMP,IFSTS
     1AT,IFTYPE,IGV,IMASS,INCPO,INCWRI,IPUNCH,IRAD,IROT,ITAPE,ITER,MTAPES
     2,NBETA,NCURVE,NLINES,NPASS,NPLOT,NPTS,NSTAGE,NTAPE,NTOPO,NWRIT,NX,
     3TOLCX,VIS1,VIS2,ANGLN(30),BBP(30,15),BETA(30,15),BLAK(30,15),BLAM(
     430),CLOSS(30,15),COMENT(12),CO(30,15),DELM(15),ESTFLO(1),EXPLOS(30
     5),FLOWI(1,1),IBETA2(30),IFCAX(30),IFLVSI(30),IFMACH(30),IFMLOS(30)
     6,IFPROF(30),IFREYL(30),IFREYN(30),IFTHIC(30),ILOSS(30),IMACHI(30),
     7KSUMRY(1),RPM2(30),RHP(30),RSP(30),ROT(1),SIGMA(30,15),STAG(30,15)
     8,WIDTH(1),X(30),XMACH(30),ZBAR(30,15),DELR(15)
      COMMON/WENT/GJ,GJCP,GR2,CXM(30,15),Y(9)
      COMMON/CALC/FLOW,ICASE,KRPM,L,NCASE,NDIV,NDUM,NMASS,NPLET,NROT,NWRS
     1ET,ALPHA(30,15),CMT(30,15),PO(30,15),RCB(30,15),STATT(30,15),TO(30
     2,15),WMT(30,15)
      COMMON/RESULT/DAMP,R(30,15)
      COMMON/TOGOSS/BET1(30,15),CLOS(30,15),CR(30,15),CUPONS(30,15),RM1(
     130,15),STAGG(30,15),STATP(30,15),TANBET(30,15)
      SLOC1(X,Y)=ATAN(TAN(X*.01745)*Y)
      TANB2(X,Y)=(-(1.-X)+SQRT((1.-X)**2-4.*X*TAN(Y)**2))/(2.*X*TAN(Y))
      IF(IFRAD-1)120,120,122
120   BHP1=(R(I,J)-RHP(I))/(RSP(I)-RHP(I))
      BHP2=(R(I-1,J)-RHP(I))/(RSP(I)-RHP(I))
      GO TO 125
122   BHP1=(R(I,J))
      BHP2=R(I-1,J)
125   BHP3=(BHP1+BHP2)/2.0
      IF(IBETA2(I).NE.1)GO TO 127
      CALL GRAPH3(BHP1,Y(1),BBP,STAG,NBETA,I)
      CALL GRAPH3(BHP1,Y(6),BBP,CLOSS,NBETA,I)
      GO TO 165
127   COSA1=COS(ALPHA(I,J))
      COSA2=COS(ALPHA(I-1,J))
C*****INTERPOLATE FOR CAMBER AND STAGGER AT BLADE INLET AND EXIT, AND
C         FOR ZBAR
      IBE=IBETA2(I)
      GO TO(165,130,133,133,130),IBE
130   CALL GRAPH3(BHP2,TEMP1,BBP,STAG,NBETA,I)
      CALL GRAPH3(BHP1,Y(1),BBP,STAG,NBETA,I)
      CALL GRAPH3(BHP2,TEMP2,BBP,BETA,NBETA,I)
      CALL GRAPH3(BHP1,Y(2),BBP,BETA,NBETA,I)
      CALL GRAPH3(BHP2,TEMP3,BBP,ZBAR,NBETA,I)
      CALL GRAPH3(BHP1,Y(5),BBP,ZBAR,NBETA,I)
C*****CORRECT STAGGER AND CAMBER FOR STREAMLINE SLOPE
      TEMP1=SLOC1(TEMP1,COSA2)
      Y(1)=SLOC1(Y(1),COSA1)
      TEMP2=SLOC1(TEMP2,COSA2)
      Y(2)=SLOC1(Y(2),COSA1)
C*****FIND EQUIVALENT BLADE INLET AND LEAVING ANGLES
      ABARU=TEMP3/(3.*TEMP3-1.)*(2.-3.*TEMP3)/(TEMP3-1.)
      ABARD=Y(5)/(3.*Y(5)-1.)*(2.-3.*Y(5))/(Y(5)-1.)
      TANB1U=TANB2(ABARU,TEMP2)*ABARU
      B1U=ATAN(TANB1U)+TEMP1
      TANB2D=TANB2(ABARD,Y(2))
      B2D=ATAN(TANB2D)+Y(1)
      GO TO 135
C*****INTERPOLATE FOR BLADE INLET ANGLES FOR IBETA2=4
133   CALL GRAPH3(BHP2,TEMP2,BBP,STAG,NBETA,I)
      CALL GRAPH3(BHP1,TEMP1,BBP,BETA,NBETA,I)
C*****CORRECT FOR STREAMLINE SLOPE
      B1U=SLOC1(TEMP2,COSA2)
```

```
      B2D=SLOC1(TEMP1,COSA1)                                            S13*0060
135   Y(2)=(B1U-B2D)/.01745                                             S13*0061
C*****FIND MAXIMUM CAMBER POINT AND STAGGER                             S13*0062
      CALL GRAPH3(BHP3,Y(5),BBP,ZBAR,NBETA,I)                           S13*0063
      ABAR=Y(5)/(3.*Y(5)-1.)*(2.-3.*Y(5))/(Y(5)-1.)                     S13*0064
      TBIN=Y(2)*.01745                                                  S13*0065
      TBIN=TANB2(ABAR,TBIN)*ABAR                                        S13*0066
      BBIN=ATAN(TBIN)                                                   S13*0067
      Y(1)=(B1U-BBIN)/.01745                                            S13*0068
      CALL GRAPH3(BHP3,Y(3),BBP,SIGMA,NBETA,I)                          S13*0069
      CALL GRAPH3(BHP3,Y(4),BBP,BLAK,NBETA,I)                           S13*0070
C*****CORRECT THICKNESS/CHORD AND SOLIDITY FOR STREAMLINE SLOPE          S13*0071
      COSA=(COSA1+COSA2)/2.                                             S13*0072
      Y(3)=Y(3)/COSA                                                    S13*0073
      CUPONS(I,J)=Y(3)                                                  S13*0074
      Y(4)=Y(4)*COSA                                                    S13*0075
C*****INTERPOLATE FOR LOSS COEFFICIENT AND ADDITIONAL LOSS FACTOR        S13*0076
      CALL GRAPH3(BHP1,Y(6),BBP,CLOSS,NBETA,I)                          S13*0077
      CALL GRAPH3(BHP1,Y(7),BBP,CO,NBETA,I)                             S13*0078
      CALL CASCDE(Y(1),Y(2),Y(4),Y(5),I,J,Y(6),VF)                      S13*0079
      SIGN=SGIN(BET1(I-1,J))                                            S13*0080
      STAGG(I-1,J)=STAGG(I-1,J)*SIGN                                    S13*0081
      TANBET(I,J)=TAN(SIGN*TANBET(I,J)/57.296)                          S13*0082
      CLOS(I,J)=CLOS(I,J)*(1.0+Y(7))                                    S13*0083
      GO TO 166                                                         S13*0084
165   TANBET(I,J)=TAN(Y(1)/57.296)*COS(ALPHA(I,J))                      S13*0085
      CLOS(I,J)=Y(6)                                                    S13*0086
166   CONTINUE                                                          S13*0087
      RETURN                                                            S13*0088
      END                                                               S13*0089
```

```
      SUBROUTINE CASCDE(STAR,THETA,THICK,ZEE,I,J,WSPEC,VF)
      COMMON/DATA/CP1,CP2,GASK,HUBLOC,IFBL,IFCYL,IFRAD,IFRPM,IFSIMP,IFSTS
     1AT,IFTYPE,IGV,IMASS,INCPO,INCWRI,IPUNCH,IRAD,IROT,ITAPE,ITER,MTAPES
     2,NBETA,NCURVE,NLINES,NPASS,NPLOT,NPTS,NSTAGE,NTAPE,NTOPO,NWRIT,NX,
     3TOLCX,VIS1,VIS2,ANGLN(30),BBP(30,15),BETA(30,15),BLAK(30,15),BLAM(
     430),CLOSS(30,15),COMENT(12),CO(30,15),DELM(15),ESTFLO(1),EXPLOS(30
     5),FLOWI(1,1),IBETA2(30),IFCAX(30),IFLVSI(30),IFMACH(30),IFMLOS(30)
     6,IFPROF(30),IFREYL(30),IFREYN(30),IFTHIC(30),ILOSS(30),IMACHI(30),
     7KSUMRY(1),RPM2(30),RHP(30),RSP(30),ROT(1),SIGMA(30,15),STAG(30,15)
     8,WIDTH(1),X(30),XMACH(30),ZBAR(30,15),DELR(15)
      COMMON/CALC/FLOW,ICASE,KRPM,L,NCASE,NDIV,NDUM,NMASS,NPLET,NROT,NWRS
     1ET,ALPHA(30,15),CMT(30,15),PO(30,15),RCB(30,15),STATT(30,15),TO(30
     2,15),WMT(30,15)
      COMMON/RESULT/DAMP,R(30,15)
      COMMON/TOGOSS/BET1(30,15),CLOS(30,15),CR(30,15),CUPONS(30,15),RM1(
     130,15),STAGG(30,15),STATP(30,15),TANBET(30,15)
      DIMENSION WORD1(3),WORD3(4),WORD4(6)
      DATA (WORD1(K),K= 1,3)/3HYES,3H NO,3HMOD/
      DATA (WORD3(K),K=1,4)/6HBY DEQ,6HIN DAT,3HDES,3HA  /
      DATA(WORD4(K),K=1,6)/6HNACA   ,6HD.C.AR,6HNOT AP,1H5,1HC,1H./
      DATA BCD1/6H STALL/,BCD2/6HED     /,BCD3/6H CHOKE/,BCD4/6HD      /
      KBETA2=IBETA2(I)
      KTHIC=IFTHIC(I)
      KCAX=IFCAX(I)
      KMACHD=IFMACH(I)
      KREYD=IFREYN(I)
      KMLOS=IFMLOS(I)
      KLVSI=IFLVSI(I)
      KPROF=IFPROF(I)
      KREYL=IFREYL(I)
      KLOS=ILOSS(I)
      SOLID=CUPONS(I,J)
      RMN=RM1(I-1,J)
      NRATIO=3
      GAMMA=1.0/(1.0-GASK/(778.0*CP1))
      IF(KBETA2.NE.5)GO TO 90
      B2=THETA/2.0+STAR-0.17*THETA/SOLID
      B1=0.0
      FRSTB2=B2
      WDES=WSPEC
      DL=WSPEC
      IF(L.EQ.1)GO TO 928
      GO TO825
   90 NCHOKE=1
      FCOEFL=THETA/2.49
      CALL OPTANG(SOLID,FCOEFL,ATTACK)
      B1DES=ATTACK+STAR
      X2=STAR+THETA/2.0
      IF(ZEE.EQ.0.5)GO TO 96
      X3=X2
      X1=ZEE*(2.0-3.0*ZEE)/((3.0*ZEE-1.0)*(ZEE-1.0))
      X2=ATAN(X1*(X1-1.0+SQRT((1.0-X1)**2-4.0*X1*(TAN(THETA/57.296))**2)
     1)/(2.0*X1*TAN(THETA/57.296)))*57.296+STAR
      B1DES=B1DES-X3+X2
96    CONTINUE
      B1=ABS(BET1(I-1,J))
C*****FIND DESIGN EXIT FLOW ANGLE, CORRECTING FOR EFFECT OF AXIAL
C             VELOCITY VARIATION IF REQUIRED
      VF=0.0
      IF(THETA.LT.20.0)VF=(20.0-THETA)/40.0
```

```
      IF(L.EQ.1)GO TO 102                                               S14*0060
  101 GO TO(103,102),KCAX                                               S14*0061
  102 VRATIO=1.0                                                        S14*0062
      GO TO 121                                                         S14*0063
  103 VRATIO=CMT(I,J)/CMT(I-1,J)                                        S14*0064
      VRATIO=VRATIO*(1.0+VF*(1.0-VRATIO)/VRATIO)                        S14*0065
      IF(VRATIO-0.8)104,104,110                                         S14*0066
  104 NRATIO=1                                                          S14*0067
      VRATIO=0.8                                                        S14*0068
      GO TO 121                                                         S14*0069
  110 IF(VRATIO-1.25)121,121,115                                        S14*0070
  115 NRATIO=2                                                          S14*0071
      VRATIO=1.25                                                       S14*0072
  121 EX=B1DES*.01745                                                   S14*0073
      BE1=ATAN(2.0*TAN(EX)/(1.0+VRATIO))/.01745                         S14*0074
      CALLSTLANG(BE1,STAR,SOLID,FCOEFL,ZEE,NCHOKE,BE2,BCHOKE,BSTALL,RE1,S14*0075
     1J,RMN)                                                            S14*0076
      IF(ZEE.EQ.0.5)GO TO 1211                                          S14*0077
      BSTALL=BSTALL-X3+X2                                               S14*0078
      BCHOKE=BCHOKE-X3+X2                                               S14*0079
 1211 IF(SOLID.LE.1.6)GO TO 1212                                        S14*0080
      SHIFT=6.07*(SOLID-1.6)                                            S14*0081
      BSTALL=BSTALL+SHIFT                                               S14*0082
      B1DES=B1DES+SHIFT                                                 S14*0083
      BCHOKE=BCHOKE+SHIFT                                               S14*0084
 1212 IF(BCHOKE.GT.B1DES-2.0)BCHOKE=B1DES-2.0                           S14*0085
      IF(BSTALL.LT.B1DES+2.0)BSTALL=B1DES+2.0                           S14*0086
      FACINC=0.6+4.0*THICK                                              S14*0087
      IF(THICK.LT.0.1)FACINC=17.125*THICK-71.25*THICK**2                S14*0088
      X3=(B1DES-X2)*(FACINC-1.0)                                        S14*0089
      XEWSTL=BSTALL+X3                                                  S14*0090
      X4=B1DES+X3                                                       S14*0091
      XEWCHK=BCHOKE+X3                                                  S14*0092
      B1DESE=BE1                                                        S14*0093
      B2DESE=BE2                                                        S14*0094
      EX=BE2*.01745                                                     S14*0095
      B2DIN=ATAN(0.5*(1.0/VRATIO+1.0)*TAN(EX))/.01745                   S14*0096
C*****CORRECT DESIGN DEFLECTION FOR EFFECTS OF THICKNESS IF REQUIRED    S14*0097
      GO TO(122,123),KTHIC                                              S14*0098
  122 EMM=0.17+B1DES*(-3.33E-4+3.33E-5*B1DES)                           S14*0099
      BEB=0.965+B1DES*(4.66E-5-8.40E-5*B1DES)                           S14*0100
      BEE=SOLID**BEB                                                    S14*0101
      D10=THETA*EMM/BEE                                                 S14*0102
      CAY=THICK*(6.25+37.5*THICK)                                       S14*0103
      DOT=D10*(CAY-1.0)                                                 S14*0104
      B2DES=B2DIN+DOT                                                   S14*0105
      GO TO 129                                                         S14*0106
  123 B2DES=B2DIN                                                       S14*0107
  129 CONTINUE                                                          S14*0108
  140 B1RAD=B1DES/57.296                                                S14*0109
      B2RAD=B2DES/57.296                                                S14*0110
C*****FIND CRITICAL MACH NUMBER, INCLUDING EFFECTS OF BLADE THICKNESS   S14*0111
      VMAX=1.03+THICK*0.7+(0.4+THICK)*COS(B1RAD)/SOLID*(TAN(B1RAD)- TAN(S14*0112
     1B2RAD))                                                           S14*0113
      CP=VMAX**2-1.0                                                    S14*0114
      CRITM=1.0+CP*(-.67+.3*CP)                                         S14*0115
      GO TO(501,145,147),KLOS                                           S14*0116
C*****FIND DIFFUSION FACTOR IF REQUIRED                                 S14*0117
  145 DEQDES=-COS(B1RAD)/COS(B2RAD)+VMAX                                S14*0118
C*****WDES IS DESIGN LOSS COEFFICIENT                                   S14*0119
```

```
      WSTARD=.003+.02375*DEQDES-.05*DEQDES**2+.125*DEQDES**3
      WDES=2.0*WSTARD*SOLID/COS(B2RAD)
      GO TO 148
  147 WDES=WSPEC
C*****CORRECT BCHOKE, BSTALL AND B1DES FOR MACH NUMBER IF REQUIRED
C*****CORRECT WDES IF REQUIRED.SET XPLOS
  148 GO TO(149,1821),KMLOS
  149 GO TO(150,180),KPROF
C*****NACA 65 TYPE MACH CORRECTIONS
150   B1DCOR=X4
      RANGE=(XEWSTL-X4)*EXP(1.386*(0.5-RMN))
      B1SCOR=B1DCOR+RANGE
      RANGE=(X4-XEWCHK)*EXP(1.386*(0.5-RMN))
      B1CCOR=B1DCOR-RANGE
      XPLOS=4.5
      IF(RMN-CRITM)151,151,152
  151 WOPT=WDES
      GO TO 194
  152 WOPT =WDES*(2.0*(RMN-CRITM)+1.0)
      GO TO 194
C*****DOUBLE C.A. TYPE MACH CORRECTIONS
180   B1DCOR=X4
      RANGE=(XEWSTL-X4)*(0.658+0.342*EXP(1.386*(0.5-RMN)))
      B1SCOR=B1DCOR+RANGE
      RANGE=(X4-XEWCHK)*EXP(1.386*(0.5-RMN))
      B1CCOR=B1DCOR-RANGE
      XPLOS=2.0
      IF(RMN-CRITM)181,181,182
  181 WOPT=WDES
      GO TO 194
182   WOPT=WDES+(RMN-CRITM)*(0.025+0.1945*(RMN-CRITM))
      GO TO 194
1821  B1DCOR=X4
      B1SCOR=XEWSTL
      B1CCOR=XEWCHK
      WOPT=WDES
      GO TO(183,184),KPROF
  183 XPLOS=4.5
      GO TO 194
  184 XPLOS=2.0
  194 GO TO(197,260),KLVSI
  197 XPLOS=EXPLOS(I)
C*****CALCULATE ACTUAL LOSS COEFF WITHOUT REYNOLDS NO CORRECTION
C*****NOTE NEXT STATEMENT REQUIRES  B1DES = B1DCOR
260   IF(B1-X4)300,500,400
  300 S=(B1-B1DCOR)/(B1CCOR-B1DCOR)
      GO TO 450
  400 S=(B1-B1DCOR)/(B1SCOR-B1DCOR)
450   DL=WOPT*(1.0+S**XPLOS)
      CLMR=DL
      GO TO 620
  500 DL=WOPT
      CLMR=DL
      GO TO 620
  501 DL=WSPEC
      WDES=0.0
      WOPT=0.0
      DEQDES=0.0
      CLMR=0.0
620   IF(DL.GT.0.25)DL=0.25
```

```
C*****CALCULATE GAS EXIT ANGLE                                          S14*0180
 630  EX=B1*.01745                                                      S14*0181
      NCHOKE=2                                                          S14*0182
      BE1DES=ATAN(2.0*TAN(B1RAD)/(1.0+VRATIO))/0.01745                  S14*0183
      BE1=ATAN(2.0*TAN(EX)/(1.0+VRATIO))/.01745                         S14*0184
C*****CORRECT DEFLECTION FOR EFFECT OF AXIAL VELOCITY VARIATION         S14*0185
 801  IF(BE1-BSTALL)805,803,803                                         S14*0186
 803  BSTLLL=BSTALL                                                     S14*0187
      CALL STLANG(BSTLLL,STAR,SOLID,FCOEFL,ZEE,NCHOKE,B2STAL,BCHOKE,BSTAS14*0188
     1LL,BE1DES,J,RMN)                                                  S14*0189
      BE2=B2STAL+BE1-BSTALL                                             S14*0190
      GO TO 806                                                         S14*0191
805   CALL STLANG(BE1,STAR,SOLID,FCOEFL,ZEE,NCHOKE,BE2,BCHOKE,BSTALL,BE1S14*0192
     1DES,J,RMN)                                                        S14*0193
 806  EX=BE2*.01745                                                     S14*0194
      B2=ATAN(0.5*(1.0/VRATIO+1.0)*TAN(EX))/.01745                      S14*0195
 807  BUNCOR=B2                                                         S14*0196
C*****CORRECT DEFLECTION FOR EFFECTS OF THICKNESS IF REQUIRED           S14*0197
      GO TO(808,809),KTHIC                                              S14*0198
 808  EMM=0.17+B1*(-3.33E-4+3.33E-5*B1)                                 S14*0199
      BEB=0.965+B1*(4.66E-5-8.40E-5*B1)                                 S14*0200
      BEE=SOLID**BEB                                                    S14*0201
      D10=THETA*EMM/BEE                                                 S14*0202
      DOT=D10*(CAY-1.0)                                                 S14*0203
      FRSTB2=B2+DOT                                                     S14*0204
      B2=FRSTB2                                                         S14*0205
      GO TO8091                                                         S14*0206
 809  FRSTB2=B2                                                         S14*0207
C*****CORRECT DEFLECTION FOR EFFECTS OF INLET MACH NUMBER               S14*0208
 8091 GO TO(8092,825,810),KMACHD                                        S14*0209
 8092 XMACH(I)=1.0                                                      S14*0210
 810  IF(RMN-1.0) 812,812,811                                           S14*0211
811   RMP=SQRT((RMN**2+2.0/(GAMMA-1.0))/(2.0*GAMMA/(GAMMA-1.0)*RMN**2-1.S14*0212
     10))                                                               S14*0213
      GO TO 813                                                         S14*0214
 812  RMP=RMN                                                           S14*0215
 813  IF(RMP-CRITM)825,825,820                                          S14*0216
820   CALL MCDEVN(CRITM,B1RAD,B2RAD,I,J,RMP,B2NEW)                      S14*0217
 822  B2=B2+ABS(B2DES-B2NEW)*XMACH(I)                                   S14*0218
C*****CALCULATE REYNOLDS NO AND CORRECTIONS IF REQUIRED                 S14*0219
825   IF(L.EQ.1)GO TO 928                                               S14*0220
      RENN=CMT(I-1,J)*(X(I)-X(I-1)+R(I,J)*TAN(ANGLN(I)/57.296)-R(I-1,J)*S14*0221
     1TAN(ANGLN(I-1)/57.296))*STATP(I-1,J)/(COS(B1/57.296)*STATT(I-1,J)*S14*0222
     2(VIS1+STATT(I-1,J)*VIS2)*GASK)                                    S14*0223
      RR=ALOG(RENN)-10.82                                               S14*0224
      RRR=RR-1.48                                                       S14*0225
      B2MR=B2                                                           S14*0226
      GO TO(915,9201),KREYL                                             S14*0227
 915  IF(RRR)917,9201,9201                                              S14*0228
 917  IF(RR)918,918,919                                                 S14*0229
 918  WREN=2.0*WDES                                                     S14*0230
      GO TO 920                                                         S14*0231
 919  WREN=WDES*(2.0-2.7386*RR**2+1.2336*RR**3)                         S14*0232
 920  DL=DL+WREN                                                        S14*0233
 9201 GO TO(921,928),KREYD                                              S14*0234
 921  IF(RRR)923,928,928                                                S14*0235
 923  IF(RR)924,924,925                                                 S14*0236
 924  DVR=.1*ABS(B1-FRSTB2)                                             S14*0237
      GO TO 926                                                         S14*0238
 925  DVR=ABS(B1-FRSTB2)*(.1-.137*RR**2+.0617*RR**3)                    S14*0239
```

```
 926  CHECK=(B2-FRSTB2)-DVR
      IF(CHECK)927,928,928
 927  B2=FRSTB2+DVR
928   TANBET(I,J)=B2
      CLOS(I,J)=DL
      STAGG(I-1,J)=X4
      IF(L.NE.NPLOT.OR.IFRPM.EQ.1.OR.IFRPM.EQ.4.OR.IFRPM.GE.7)GO TO 999
      IF(KBETA2.EQ.5)GO TO 9981
      IF(J.NE.1)GO TO 991
      WRITE(NTAPE,13)I,L,WORD1(KCAX),WORD1(KTHIC),WORD1(KMACHD),XMACH(I)
     1,WORD1(KREYD)
      GO TO(980,990,990),KLOS
 980  WRITE(NTAPE,14)
      GO TO 991
 990 0WRITE(NTAPE,15)WORD3(KLOS -1),WORD3(KLOS +1),WORD1(KMLOS),WORD4(KP
     1ROF),WORD4(KPROF+3),XPLOS,WORD1(KREYL)
 991  WRITE(NTAPE,5)J
      GO TO(992,993,994),NRATIO
 992  WRITE(NTAPE,1)
      GO TO 994
 993  WRITE(NTAPE,2)
 994  IF(BE1-BSTALL)995,996,996
 995  IF(BE1-BCHOKE)997,997,998
 996  WRITE(NTAPE,7)BCD1,BCD2
      GO TO 998
 997  WRITE(NTAPE,7)BCD3,BCD4
998   WRITE(NTAPE,11,STAR,SOLID,THETA,THICK,ZEE,B1,X4,XEWCHK,XEWSTL,B1DE
     1SE,B2DESE,B2DIN
     0WRITE(NTAPE,12)B2DES,BE1,BE2,BUNCOR,FRSTB2,RMN,VMAX,CRITM,B2MR,REN
     1N,B2,DEQDES,WDES,WOPT,B1SCOR,B1CCOR,CLMR,DL
      GO TO999
 9981  IF(J-1)999,9982,9983
 9982 WRITE(NTAPE,3)I,L,WORD1(KREYD),WORD1(KREYL)
 9983 WRITE(NTAPE,4)J,FRSTB2,RENN,B2,WSPEC,DL
 999  CONTINUE
  1   FORMAT(10X,62HCMT(EXIT)/CMT(INLET) IS LESS THAN 0.8. IT IS SET EQU
     1AL TO 0.8.)
  2  0FORMAT(10X,67HCMT(EXIT)/CMT(INLET) IS GREATER THAN 1.25. IT IS SET
     2 EQUAL TO 1.25.)
  3  0FORMAT(///,10X,81HI.G.V. CALCULATION APPLIED TO ALL SECTIONS OF BL
     1ADE WITH TRAILING EDGE AT STATION,I3,4X,9HPASS NO =,I3//1X,74HDEFL
     2ECTION GIVEN BY MODIFIED CARTERS RULE      CORRECTED FOR REYNOLDS
     3NO.-,A3/1X74HLOSS COEFFICIENT GIVEN IN DATA                  CORREC
     4TED FOR REYNOLDS NO.-A3//)
  4  0FORMAT(1X,16HSTREAMLINE NO = ,I3/10X,29HB2 FROM MODIFIED CARTERS R
     1ULE,7X1H=F7.3/10X,23HSECTION REYNOLDS NUMBER,13X1H=,F8.0/10X,33HB2
     2 CORRECTED FOR REY.NO.(IF REQD),3X,1H=,F7.3/10X,12HCL FROM DATA,24
     3X,1H=,F7.4/10X,33HCL CORRECTED FOR REY.NO.(IF REQD),3X,1H=,F7.4//)
  5   FORMAT(1X,16HSTREAMLINE NO = ,I3)
  7   FORMAT(10X,21HTHIS BLADE SECTION IS,2A6)
 11  0FORMAT(                 10X,8HSTAGGER= ,F8.4,2X,                 8HSOLID
     1. = ,F8.4,2X,8HCAMBER = ,F8.4,2X,8HTHICN. = ,F8.4,2X,8HZBAR    = ,F
     28.4,2X,8HIN.ANG.= ,F8.4,/10X,8HB1DES   = ,F8.4,2X8HBCHOKE =F8.4, 10
     3H  BSTALL = F8.4,2X,8HB1DESE = ,F8.4,2X,8HB2DESE = ,F8.4,2X,8HB2DA
     4X   = ,F8.4)
 12  0FORMAT(10X,8HB2DES   = ,F8.4,2X,8HBE1     = ,F8.4,2X,8HBE2     = ,F8.
     14,2X,8HB2AXVL = ,F8.4,2X,8HB2THIC = ,F8.4,2X,8HRM1      = ,F8.4,/10X
     2,8HVMAX    = ,F8.4,2X,8HCRITM   = ,F8.4,2X,8HB2M      = ,F8.4,2X,8HREY
     3N.   = ,F8.0,2X,8HB2       = ,F8.4,2X,8HDEQDES = ,F8.4,/10X,8HWDES
     4= ,F8.4,2X,8HWOPT    = ,F8.4,2X,8HB1SCOR = ,F8.4,2X,8HB1CCOR = ,F8.
```

```
     54,2X,8HCLI     = ,F8.4,2X,8HCL      = ,F8.4,//)
 13 0FORMAT(///,10X,78HCALCULATION METHOD FOR ALL SECTIONS OF BLADE ROWS
    1 WITH TRAILING EDGE AT STATION I3,4X,10HPASS NO =  ,I3//1X,35HDEFLES
    2CTION GIVEN BY CARTER/LIEBLEIN,2X,42HCORRECTIONS APPLIED-  AXIAL VS
    3ELOCITY RATIO,
    3         13X,2H- ,A3/60X,27HBLADE THICKNESS/CHORD RATIO,6X,2H- ,A3,
    4/60X,26HSUPER-CRITICAL MACH NUMBER,7X,2H- ,A3,/60X,29HVALUE OF XMAS
    5CH USED FOR ABOVE,4X,1H-,F5.2/60X,28HSUB-CRITICAL REYNOLDS NUMBER,
    65X,2H- ,A3/)
 14  FORMAT(1X,44HLOSS COEFF GIVEN IN DATA WITH NO CORRECTIONS//)
 15 0FORMAT(1X,23HLOSS COEFFICIENT GIVEN ,A6,A3,5X,48HCORRECTIONS APPLIS
    1ED-  SUPER-CRITICAL MACH NUMBER,7X,2H- ,A3/60X,25HINCID.RANGE/MACHS
    2 NO RULES8X,2H- ,A6,A1/60X,29HVALUE OF XPLOS USED FOR ABOVE,4X,1H-
    3F5.2/60X,28HSUB-CRITICAL REYNOLDS NUMBER,5X,2H- ,A3//)
1000 RETURN
     END
```

```
      SUBROUTINE GRAPH2(D1,D2,D3,S,Y)
      QS=SQRT(1.0/S)
      IF(S-1.0)1,2,2
    1 CONST=(D1-D2)/.4142
      Y=D2+CONST*(QS-1.0)
      GO TO 3
    2 CONST=(D2-D3)/.1835
      Y=D3+CONST*(QS-.8165)
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STLANG(BIN,STAR,SOLID,COEFFL,ZEE,NCHOKE,BEXIT,BCHOKE,BS
     1TALL,B1OPT,J,RM1)
      SOL(X)=(SQRT(1.0/X)-1.0)/.184
      THETA=2.49*COEFFL
      NW=1
      S=STAR
      GO TO(2,600),NCHOKE
C*****DO ALL CALCULATIONS INITIALLY FOR SOLIDITY=1.5
 2    IF(COEFFL)3,14,3
 3    DELTAC=4.0
      CLIFT=0.0
      NUM=0
      DO 5 I=1,3
      CLIFT=CLIFT+4.0
      NUM=NUM+1
      IF(COEFFL-CLIFT)11,10,5
 5    CONTINUE
      DELTAC=3.0
      DO 6 I=1,4
      CLIFT=CLIFT+3.0
      NUM=NUM+1
      IF(COEFFL-CLIFT)11,10,6
 6    CONTINUE
      GO TO 11
 10   INDIC=1
      GO TO 12
 11   INDIC=2
 12   GO TO (20,30,40,50,60,70,80),NUM
C*****LIFT COEFF.=0  SOLIDITY=1.5
 14   INDIC=1
      CLIFT=COEFFL
 15   D1=-6.1999716
      D2=1.0566653
      D3=-.66665219E-03
      E1=7.5000285
      E2=1.4666653
      E3=-.66666521E-02
 17   GO TO (105,100),INDIC
C*****LIFT COEFF.=4   SOLIDITY=1.5
 20   D1=-14.90006
      D2=1.3466693
      D3=-.13333622E-02
      E1=29.19981
      E2=.50334245
      E3=.42221209E-02
 22   GO TO (105,100),INDIC
C*****LIFT COEFF.=8   SOLIDITY=1.5
 30   D1=4.6664719
      D2=.66334388
      D3=.59998698E-02
      E1=27.0
      E2=0.8
      E3=0.0
 32   GO TO (105,100),INDIC
C*****LIFT COEFF.=12  SOLIDITY=1.5
 40   D1=10.111044
      D2=.70889307
      D3=.57777198E-02
      E1=31.0
      E2=0.825
```

```
      E3=0.0
 42   GO TO (105,100),INDIC
C*****LIFT COEFF.=15  SOLIDITY=1.5
 50   D1=13.500001
      D2=.77476182
      D3=.39047629E-02
      E1=36.39999
      E2=.65904836
      E3=-.38096377E-03
 52   GO TO (105,100),INDIC
C*****LIFT COEFF.=18  SOLIDITY=1.5
 60   D1=-1.6249918
      D2=1.7899992
      D3=-.89999852E-02
      E1=30.487291
      E2=1.4900181
      E3=-.017500357
 62   GO TO (105,100),INDIC
C*****LIFT COEFF.=21  SOLIDITY=1.5
70    D1=18.0
      D2=1.0
      D3=.0
      E1=36.9464
      E2=.71024293
      E3=.0
 72   GO TO (105,100),INDIC
C*****LIFT COEFF.=24  SOLIDITY=1.5
 80   D1=15.0
      D2=1.2
      D3=0.0
      E1=39.5
      E2=.61
      E3=0.0
 82   GO TO (105,100),INDIC
C*****FIND B2 FOR SOLIDITY=1.5
 100  AC15=D1+D2*S+D3*S**2
      AS15=E1+E2*S+E3*S**2
 102  INDIC=INDIC-1
      IF(COEFFL-CLIFT)104,104,70
 104  NUM=NUM-1
      IF(NUM)15,15,12
 105  BC15=D1+D2*S+D3*S**2
      BS15=E1+E2*S+E3*S**2
 1053 IF(COEFFL-CLIFT)107,106,107
 106  COUT15=BC15
      SOUT15=BS15
      GO TO 202
 107  COUT15=AC15-(CLIFT-COEFFL)*(AC15-BC15)/DELTAC
      SOUT15=AS15-(CLIFT-COEFFL)*(AS15-BS15)/DELTAC
C*****FIND NEGATIVE STALLING AND STALLING INLET ANGLES FOR SOLIDITY=1.0
 202  IF(COEFFL)203,214,203
 203  DELTAC=4.0
      CLIFT=0.0
      NUM=0
      DO 205 I=1,3
      CLIFT=CLIFT+4.0
      NUM=NUM+1
      IF(COEFFL-CLIFT)211,210,205
 205  CONTINUE
      DELTAC=3.0
```

```
      DO 206 I=1,4
      CLIFT=CLIFT+3.0
      NUM=NUM+1
      IF(COEFFL-CLIFT)211,210,206
  206 CONTINUE
      GO TO 211
  210 INDIC=1
      GO TO 212
  211 INDIC=2
  212 GO TO (220,230,240,250,260,270,280),NUM
C*****LIFT COEFF.=0    SOLIDITY=1.0
  214 INDIC=1
      CLIFT=COEFFL
  215 D1=-5.9999723
      D2=.87999869
      D3=.17777922E-02
      E1=9.1999397
      E2=1.353336
      E3=-.53333622E-02
  217 GO TO (305,300),INDIC
C*****LIFT COEFF.=4    SOLIDITY=1.0
  220 D1=-5.0999717
      D2=1.083332
      D3=-.66665219E-03
      E1=18.200332
      E2=.9699842
      E3=-.11109374E-02
  222 GO TO (305,300),INDIC
C*****LIFT COEFF.=8    SOLIDITY=1.0
  230 D1=-.39972782
      D2=1.0066536
      D3=.44458912E-03
      E1=20.7
      E2=0.867
      E3=0.0
  232 GO TO (305,300),INDIC
C*****LIFT COEFF.=12   SOLIDITY=1.0
  240 D1=.72216034
      D2=1.1755589
      D3=-.17778211E-02
      E1=23.5
      E2=0.87
      E3=0.0
  242 GO TO (305,300),INDIC
C*****LIFT COEFF.=15   SOLIDITY=1.0
  250 D1=10.680085
      D2=.69132743
      D3=.57334245E-02
      E1=28.66017
      E2=.83265435
      E3=-.25331425E-02
  252 GO TO (305,300),INDIC
C*****LIFT COEFF.=18   SOLIDITY=1.0
  260 D1=13.300028
      D2=.77999797
      D3=.30000335E-02
      E1=30.199638
      E2=.66002568
      E3=.19995768E-02
  262 GO TO (305,300),INDIC
```

```
C*****LIFT COEFF.=21   SOLIDITY=1.0
270    D1=24.0
       D2=0.7
       D3=0.0
       E1=32.000088
       E2=0.7
       E3=.0
 272   GO TO (305,300),INDIC
C*****LIFT COEFF.=24   SOLIDITY=1.0
 280   D1=13.250065
       D2=1.0499956
       D3=0.0
       E1=36.200032
       E2=.5799978
       E3=0.0
 282   GO TO (305,300),INDIC
 300   AC10=D1+D2*S+D3*S**2
       AS10=E1+E2*S+E3*S**2
 302   INDIC=INDIC-1
       IF(COEFFL-CLIFT)304,304,270
 304   NUM=NUM-1
       IF(NUM)215,215,212
 305   BC10=D1+D2*S+D3*S**2
       BS10=E1+E2*S+E3*S**2
 3053  IF(COEFFL-CLIFT)307,306,307
 306   COUT1=BC10
       SOUT1=BS10
       GO TO 402
 307   COUT1=AC10-(CLIFT-COEFFL)*(AC10-BC10)/DELTAC
       SOUT1=AS10-(CLIFT-COEFFL)*(AS10-BS10)/DELTAC
C*****FIND NEGATIVE STALLING AND STALLING INLET ANGLE FOR SOLIDITY=0.5
 402   IF(COEFFL)403,414,403
 403   DELTAC=4.0
       CLIFT=0.0
       NUM=0
       DO 405 I=1,3
       CLIFT=CLIFT+4.0
       NUM=NUM+1
       IF(COEFFL-CLIFT)411,410,405
 405   CONTINUE
       DELTAC=3.0
       DO 406 I=1,4
       CLIFT=CLIFT+3.0
       NUM=NUM+1
       IF(COEFFL-CLIFT)411,410,406
 406   CONTINUE
       GO TO 411
 410   INDIC=1
       GO TO 412
 411   INDIC=2
  412  GO TO (420,430,440,430,460,430,430),NUM
C*****LIFT COEFF.=0     SOLIDITY=0.5
 414   INDIC=1
       CLIFT=COEFFL
 415   D1=2.0*BC10-BC15
       D2=0.0
       D3=0.0
       E1=2.0*BS10-BS15
       E2=0.0
       E3=0.0
```

```
 417  GO TO (505,500),INDIC
C*****LIFT COEFF.=4    SOLIDITY=0.5
 420  D1=-.31698847
      D2=.850683
      D3=.19931806E-02
      E1=-19.502003
      E2=2.0900799
      E3=-.010000791
 422  GO TO (505,500),INDIC
C*****LIFT COEFF.=8,15  SOLIDITY=0.5
 430  GO TO (432,431),INDIC
 431  D1=AC10+(AC10-AC15)*2.25
      E1=AS10+(AS10-AS15)*2.25
      GO TO 433
 432  D1=BC10+(BC10-BC15)*2.25
      E1=BS10+(BS10-BS15)*2.25
 433  D2=0.0
      D3=0.0
      E2=0.0
      E3=0.0
 435  GO TO (505,500),INDIC
C*****LIFT COEFF.=12    SOLIDITY=0.5
 440  D1=.066037178
      D2=1.0215195
      D3=-.16856238E-04
      E1=15.0
      E2=1.0
      E3=0.0
 442  GO TO (505,500),INDIC
C*****LIFT COEFF.=18    SOLIDITY=0.5
 460  D1=-14.011114
      D2=1.8505599
      D3=-.010006979
      E1=61.196022
      E2=-1.0497981
      E3=.021997466
 462  GO TO (505,500),INDIC
 500  AC05=D1+D2*S+D3*S**2
      AS05=E1+E2*S+E3*S**2
 502  INDIC=INDIC-1
      IF(COEFFL-CLIFT)504,504,431
 504  NUM=NUM-1
      IF(NUM)415,415,412
 505  BC05=D1+D2*S+D3*S**2
      BS05=E1+E2*S+E3*S**2
 5056 IF(COEFFL-CLIFT)507,506,507
 506  COUT05=BC05
      SOUT05=BS05
      GO TO 602
 507  COUT05=AC05-(CLIFT-COEFFL)*(AC05-BC05)/DELTAC
      SOUT05=AS05-(CLIFT-COEFFL)*(AS05-BS05)/DELTAC
C*****INTERPOLATE FOR SOLIDITY
602   CALL GRAPH2(COUT05,COUT1,COUT15,SOLID,COUT)
      BCHOKE=COUT
      CALL GRAPH2(SOUT05,SOUT1,SOUT15,SOLID,SOUT)
      BSTALL=SOUT
600   CALL OUTANG(STAR,THETA,SOLID,BIN,ZEE,BLEAV,B1OPT,J,RM1)
      BEXIT=BLEAV
 604  RETURN
      END
```

```
      SUBROUTINE OUTANG(STAR,THETA,SOLID,B1STAR,ZEE,BLEAV,B1OPT,J,RM1)
      DIMENSION DBDA(4)
      DIMENSION AA(20),BB(20),CC(20),XP(1),YP(1),ZP(1)
C*****FIND DESIGN EXIT ANGLE USING CARTER#S RULE
      RG=STAR*.01745
 102  EM1=.2158+RG*(.0538+RG*(.05+RG*(.06258-RG*.0272)))
      IF(STAR-30.0)110,110,111
 110  W=-1.547*RG+2.07
      GO TO 112
111   W=-0.8766*RG+1.72
 112  EM=EM1*(1.0-W*(1.0-2.0*ZEE))
      DELTA=EM*THETA*SQRT(1.0/SOLID)
      RTH=THETA*.01745
      ABAR=ZEE/(3.*ZEE-1.)*(2.-3.*ZEE)/(ZEE-1.)
      TANB2D=(-(1.-ABAR)+SQRT((1.-ABAR)**2-4.*ABAR*TAN(RTH)**2))/(2.*
     1ABAR*TAN(RTH))
      FUNCY=4.0-3.08*RM1
      IF(RM1.LT.1.0)FUNCY=1.0
      IF(RM1.GT.1.3)FUNCY=0.0
      GUTCY=0.0
      IF(RM1.GT.1.0)GUTCY=0.4
      B2DIN=STAR+ATAN(TANB2D)/0.01745+DELTA+3.*EXP(-0.22*THETA)*FUNCY-GUS
     1TCY
C     FIND D DELTAB/DALPHA
      AB1=ABS(B1STAR)
      IF(AB1-B1OPT)205,205,207
205   DB=1.0
      GO TO 500
207   IF(AB1)215,215,211
 211  IF(AB1-30.0)220,220,212
 212  IF(AB1-50.0)230,230,240
C     *****B1STAR=0.0*****
215   DBDA(1)=1.0-1.28*EXP(-3.46*SOLID)
      N=1
      IF(AB1-0.0)250,250,260
C     *****B1STAR=30.*****
220   DBDA(2)=1.0-1.08*EXP(-2.98*SOLID)
      N=2
      IF(AB1-30.0)215,250,270
C     *****B1STAR=50.*****
230   DBDA(3)=1.0-0.925*EXP(-0.90*SOLID)
      N=3
      IF(AB1-50.0)220,250,270
C      *****B1STAR=70.*****
240   DBDA(4)=1.0-0.925*EXP(-0.90*SOLID)
      N=4
      IF(AB1-70.0)230,250,230
 250  DB=DBDA(N)
      GO TO 280
 260  DB=DBDA(N)+AB1/30.0*(DBDA(N+1)-DBDA(N))
      GO TO 280
 270  ON=N
      DB=DBDA(N)+(AB1-10.0*(2.0*ON-1.0))/20.0*(DBDA(N+1)-DBDA(N))
 280  IF(B1STAR)281,281,500
 281  D=8.404E-06+2.756908*SOLID-2.7690602*SOLID**2+.81905201*
     1SOLID**3
      DB=2.0*D-DB
 500  DBSTAR=B1OPT-B2DIN
      BLEAV=B1STAR-DBSTAR-DB*(B1STAR-B1OPT)
      RETURN
```

END S17*0060

```
      SUBROUTINE OPTANG(SOLID,COEFFL,ATTACK)
      DIMENSION A123(3,8)
      DATA((A123(I,J),I=1,3),J=1,8)/-0.3,5.3,-1.4,0.8,7.1,-1.4,1.8,9.4,-
     11.6,3.4,10.2,-1.2,3.2,14.6,-2.8,3.0,18.0,-4.0,6.0,15.0,-2.0,5.3,22
     2.1,-5.4/
      M=1
      DELTAC=4.0
      CLIFT=0.0
10    CLIFT=CLIFT+DELTAC
      M=M+1
      IF(COEFFL.LE.CLIFT)GO TO 60
      IF(M.EQ.4)DELTAC=3.0
      IF(M.EQ.8)GO TO 60
      GO TO 10
60    A=(A123(3,M)*SOLID+A123(2,M))*SOLID+A123(1,M)
      B=(A123(3,M-1)*SOLID+A123(2,M-1))*SOLID+A123(1,M-1)
      ATTACK=A-(A-B)*(CLIFT-COEFFL)/DELTAC
      RETURN
      END
```

```
      SUBROUTINE MCDEVN(CRITM,B1RAD,B2RAD,I,J,RMP,B2NEW)
      COMMON/DATA/CP1,CP2,GASK,HUBLOC,IFBL,IFCYL,IFRAD,IFRPM,IFSIMP,IFSTS
     1AT,IFTYPE,IGV,IMASS,INCPO,INCWRI,IPUNCH,IRAD,IROT,ITAPE,ITER,MTAPES
     2,NBETA,NCURVE,NLINES,NPASS,NPLOT,NPTS,NSTAGE,NTAPE,NTOPO,NWRIT,NX,
     3TOLCX,VIS1,VIS2,ANGLN(30),BBP(30,15),BETA(30,15),BLAK(30,15),BLAM(
     430),CLOSS(30,15),COMENT(12),CO(30,15),DELM(15),ESTFLO(1),EXPLOS(30
     5),FLOWI(1,1),IBETA2(30),IFCAX(30),IFLVSI(30),IFMACH(30),IFMLOS(30)
     6,IFPROF(30),IFREYL(30),IFREYN(30),IFTHIC(30),ILOSS(30),IMACHI(30),
     7KSUMRY(1),RPM2(30),RHP(30),RSP(30),ROT(1),SIGMA(30,15),STAG(30,15)
     8,WIDTH(1),X(30),XMACH(30),ZBAR(30,15),DELR(15)
      COMMON/WENT/GJ,GJCP,GR2,CXM(30,15),Y(9)
      COMMON/CALC/FLOW,ICASE,KRPM,L,NCASE,NDIV,NDUM,NMASS,NPLET,NROT,NWRS
     1ET,ALPHA(30,15),CMT(30,15),PO(30,15),RCB(30,15),STATT(30,15),TO(30
     2,15),WMT(30,15)
      COMMON/RESULT/DAMP,R(30,15)
      EMM(X)=(1.0+(GAMMA-1.0)/2.0*X**2)**(1.0/(GAMMA-1.0))
      GAMMA=CP1/(CP1-GASK/778.0)
      C1=COS(B1RAD)
      C2=COS(B2RAD)
      EXP=(GAMMA+1.0)/(2.0*(GAMMA-1.0))
      EXPO=(GAMMA-1.0)/2.0
      M=I-1
      V1CRIT=SQRT(GR2*GAMMA/(1.0/CRITM**2+EXPO))
      IF(J-2)20,22,22
   20 A2=R(I,2)**2-R(I,1)**2
      A1=R(M,2)**2-R(M,1)**2
      GO TO 25
   22 IF(J-NLINES)23,24,24
   23 A2=(R(I,J+1)**2-R(I,J-1)**2)/2.0
      A1=(R(M,J+1)**2-R(M,J-1)**2)/2.0
      GO TO 25
   24 A2=R(I,NLINES)**2-R(I,NLINES-1)**2
      A1=R(M,NLINES)**2-R(M,NLINES-1)**2
C     SOLVE CONTINUITY EQUATION FOR CRITICAL MACH NUMBER AT BLADE EXIT
   25 G=A1/A2*C1/C2*CRITM/((1.0+EXPO*CRITM**2)**EXP)
      GUESS=G
      NCOUNT=0
   27 EMNEW=G*(1.0+EXPO*GUESS**2)**EXP
      IF(EMNEW-1.0)279,279,271
  271 EMNEW=1.0
      IF(L-NPLOT)32,272,272
  272 WRITE (NTAPE,6)
      GO TO 32
  279 DIFF=EMNEW-GUESS
      NCOUNT=NCOUNT+1
      IF(ABS(DIFF)-.001)32,32,28
   28 GUESS=GUESS+DIFF
      IF(NCOUNT-40)27,27,30
   30 IF(L-NPLOT)32,31,31
   31 WRITE (NTAPE,5)
C     FIND RATIO OF EXIT TO INLET VELOCITY AT CRITICAL MACH NUMBER
   32 RATIO=SQRT(GR2*GAMMA/(1.0/EMNEW**2+EXPO))/V1CRIT
      V1=SQRT(GR2*GAMMA/(1.0/RMP**2+EXPO))
      V2=RATIO*V1
      EM2=SQRT(1.0/(GR2*GAMMA/V2**2-EXPO))
      COSB2=C1*A1/A2*EMM(EM2)/EMM(RMP)/RATIO
      IF(ABS(COSB2)-1.0)40,33,33
   33 COSB2=0.99*COSB2/ABS(COSB2)
   40 B2NEW=ATAN(SQRT(1.0-COSB2**2)/COSB2)/.01745*B2RAD/ABS(B2RAD)
    5 FORMAT(1H0,5X,40HNON-CONVERGENCE ON CRITM2 AFTER 40 LOOPS)
```

```
6     FORMAT(1H0,5X,60HCRITM2 HAS EXCEEDED UNITY-IT IS TEMPORARILY SET ES19*0060
     1QUAL TO 1.0)                                                      S19*0061
      RETURN                                                            S19*0062
      END                                                               S19*0063
```

```
      SUBROUTINE BLTHIC(I,BLOCK)
      COMMON/DATA/CP1,CP2,GASK,HUBLOC,IFBL,IFCYL,IFRAD,IFRPM,IFSIMP,IFSTS
     1AT,IFTYPE,IGV,IMASS,INCPO,INCWRI,IPUNCH,IRAD,IROT,ITAPE,ITER,MTAPES
     2,NBETA,NCURVE,NLINES,NPASS,NPLOT,NPTS,NSTAGE,NTAPE,NTOPO,NWRIT,NX,
     3TOLCX,VIS1,VIS2,ANGLN(30),BBP(30,15),BETA(30,15),BLAK(30,15),BLAM(
     430),CLOSS(30,15),COMENT(12),CO(30,15),DELM(15),ESTFLO(1),EXPLOS(30
     55),FLOWI(1,1),IBETA2(30),IFCAX(30),IFLVSI(30),IFMACH(30),IFMLOS(30)
     6,IFPROF(30),IFREYL(30),IFREYN(30),IFTHIC(30),ILOSS(30),IMACHI(30),
     7KSUMRY(1),RPM2(30),RHP(30),RSP(30),ROT(1),SIGMA(30,15),STAG(30,15)
     8,WIDTH(1),X(30),XMACH(30),ZBAR(30,15),DELR(15)
      COMMON/CALC/FLOW,ICASE,KRPM,L,NCASE,NDIV,NDUM,NMASS,NPLET,NROT,NWRS
     1ET,ALPHA(30,15),CMT(30,15),PO(30,15),RCB(30,15),STATT(30,15),TO(30
     2,15),WMT(30,15)
      COMMON/RESULT/DAMP,R(30,15)
      DIMENSION BLOCK(30),HUBI(30),CASEI(30),THETHB(30),THETCS(30)
      IF(I.GT.1)GO TO 100
      THETHB(1)=(R(1,1)-RHP(1))/1.4
      IF(THETHB(1).LE.0.0)GO TO 50
      HUBI(1)=(THETHB(1)*CMT(1,1)**3.4/0.006)**1.25
      GO TO 60
50    HUBI(1)=0.0
60    THETCS(1)=(RSP(1)-R(1,NLINES))/1.4
      IF(THETCS(1).LE.0.0)GO TO 70
      CASEI(1)=(THETCS(1)*CMT(1,NLINES)**3.4/0.006)**1.25
      GO TO 400
70    CASEI(1)=0.0
      GO TO 400
100   DXHUB=X(I)-X(I-1)+R(I,1)*TAN(ANGLN(I)/57.296)-R(I-1,1)*TAN(ANGLN(I
     1-1)/57.296)
      DXCASE=X(I)-X(I-1)+R(I,NLINES)*TAN(ANGLN(I)/57.296)-R(I-1,NLINES)*
     1TAN(ANGLN(I-1)/57.296)
      HUBI(I)=HUBI(I-1)+DXHUB*CMT(I,1)**4
      CASEI(I)=CASEI(I-1)+DXCASE*CMT(I,NLINES)**4
      THETA=0.006/CMT(I,1)**3.4*HUBI(I)**0.8
      H=(THETA-THETHB(I-1))/DXHUB*30.0+1.5
      IF(H.LT.1.1)H=1.1
      IF(H.GT.2.2)H=2.2
      DELTA=H*THETA
      ANNUL=RSP(I)-RHP(I)
      IF(DELTA.GT.ANNUL*0.25)DELTA=ANNUL*0.25
      RHUB=RHP(I)+DELTA
      THETHB(I)=THETA
      THETA=0.006/CMT(I,NLINES)**3.4*CASEI(I)**0.8
      H=(THETA-THETCS(I-1))/DXCASE*30.0+1.5
      IF(H.LT.1.1)H=1.1
      IF(H.GT.2.2)H=2.2
      DELTA=H*THETA
      IF(DELTA.GT.0.20*ANNUL)DELTA=ANNUL*0.2
      RCASE=RSP(I)-DELTA
      THETCS(I)=THETA
      GAP=(RCASE-RHUB)/ANNUL
      IF(GAP.GT.0.7)GO TO 200
      RHUB=RHP(I)+(RHUB-RHP(I))*0.3/(1.0-GAP)
      RCASE=RSP(I)-(RSP(I)-RCASE)*0.3/(1.0-GAP)
200   RHUB=0.2*RHUB+0.8*R(I,1)
      RCASE=0.2*RCASE+0.8*R(I,NLINES)
      RH2=RHUB**2
      RI12=R(I,1)**2
      RRR=(RCASE**2-RH2)/(R(I,NLINES)**2-RI12)
      DO 220 J=1,NLINES
```

```
220    R(I,J)=SQRT(RRR*(R(I,J)**2-RI12)+RH2)
400    BLOCK(I)=1.0-(R(I,NLINES)**2-R(I,1)**2)/(RSP(I)**2-RHP(I)**2)
       IF(ITER.EQ.2)RETURN
       DELHUB=RHUB-RHP(I)
       DELCSE=RSP(I)-RCASE
       WRITE(NTAPE,300)I,BLOCK(I),DELHUB,DELCSE
300    FORMAT(20X,7HSTATION,I3,11H BLOCKAGE =,F9.6,9H DELHUB =,F9.6,10H D
      1ELCASE =,F9.6)
       RETURN
       END
```

```
      SUBROUTINE PRINT
      COMMON/CALC/FLOW,ICASE,KRPM,L,NCASE,NDIV,NDUM,NMASS,NPLET,NROT,NWRS
     1ET,ALPHA(30,15),CMT(30,15),PO(30,15),RCB(30,15),STATT(30,15),TO(30
     2,15),WMT(30,15)
      COMMON/DATA/CP1,CP2,GASK,HUBLOC,IFBL,IFCYL,IFRAD,IFRPM,IFSIMP,IFSTS
     1AT,IFTYPE,IGV,IMASS,INCPO,INCWRI,IPUNCH,IRAD,IROT,ITAPE,ITER,MTAPES
     2,NBETA,NCURVE,NLINES,NPASS,NPLOT,NPTS,NSTAGE,NTAPE,NTOPO,NWRIT,NX,
     3TOLCX,VIS1,VIS2,ANGLN(30),BBP(30,15),BETA(30,15),BLAK(30,15),BLAM(
     430),CLOSS(30,15),COMENT(12),CO(30,15),DELM(15),ESTFLO(1),EXPLOS(30
     5),FLOWI(1,1),IBETA2(30),IFCAX(30),IFLVSI(30),IFMACH(30),IFMLOS(30)
     6,IFPLOF(30),IFREYL(30),IFREYN(30),IFTHIC(30),ILOSS(30),IMACHI(30),
     7KSUMRY(1),RPM2(30),RHP(30),RSP(30),ROT(1),SIGMA(30,15),STAG(30,15)
     8,WIDTH(1),X(30),XMACH(30),ZBAR(30,15),DELR(15)
      COMMON/WENT/GJ,GJCP,GR2,CXM(30,15),Y(9)
      COMMON/RESULT/DAMP,R(30,15)
      COMMON/TOGOSS/BET1(30,15),CLOS(30,15),CR(30,15),CUPONS(30,15),RM1(
     130,15),STAGG(30,15),STATP(30,15),TANBET(30,15)
      DIMENSION VABS(15),GAMMA(15),XABS(15),RCF(15),CCF(15),CONFAC(15),D
     1ENS(15),RN(15),TN(15),RO(15),TO(15),GAMIN(15),GAMUP(15),GOJ(15),GN
     2J(15)
      G=32.175
      CJ=778.0
      NWRIT=NWRIT+INCWRI
      WRITE(NTAPE,1)
      WRITE(NTAPE,10)L
1     FORMAT(1H1)
10    FORMAT(55X,16HOUTPUT FROM PASS,I3,/55X,19H*******************)
      LNCT=2
      DO 900 I=1,NX
      IF(NLINES+9.LE.60-LNCT)GO TO 50
      WRITE(NTAPE,1)
      LNCT=0
50    WRITE(NTAPE,55)I
55    FORMAT(/60X,7HSTATION I3,/60X,10H**********
     6                          ,//,53X,23HGENERAL FLOW PARAMETERS,//,1X,4
     1HLOCA,3X,6HRADIUS,8X,19HV E L O C I T I E S,8X,12HTEMPERATURES,8X,
     29HPRESSURES,6X,4HMACH,3X,5HWHIRL,3X,6HSLOPE ,2X,6HRAD.OF,2X,6HSTATS
     3IC,3X,4HLOCA,/,1X,4HTION,13X,35HABSOLUTE  MERIDNL. TANGENTL.  TOTA
     4L,3X,6HSTATIC,4X,5HTOTAL,4X,6HSTATIC,2X,6HNUMBER,2X,5HANGLE,3X5HAN
     5GLE,2X,8HCURVTRE.,8H DENSITY,2X,4HTION,/)
      LNCT=LNCT+9+NLINES
      DO 300 J=1,NLINES
      VABS(J)=CMT(I,J)*(1.0+CXM(I,J)**2)
      VTAN=CMT(I,J)*CXM(I,J)
      GAMMA(J)=1.0/(1.0-GASK/(CJ*(CP1+CP2*STATT(I,J))))
      IF(I.GT.1)GAMUP(J)=1.0/(1.0-GASK/(CJ*(CP1+CP2*STATT(I-1,J))))
      XABS(J)=VABS(J)/SQRT(GASK*GAMMA(J)*G*STATT(I,J))
      ANG=ATAN(CXM(I,J))*57.296
      RC=0.0
      IF(ABS(RCB(I,J)).GT.1.0E-05)RC=1.0/RCB(I,J)
      SLOPE=ALPHA(I,J)*57.296
      DENS(J)=STATP(I,J)/(GASK*STATT(I,J))
300   WRITE(NTAPE,350)J,R(I,J),VABS(J),CMT(I,J),VTAN,TO(I,J),STATT(I,J),
     1PO(I,J),STATP(I,J),XABS(J),ANG,SLOPE,RC,DENS(J),J
350   FORMAT(1X,I3,2X,F10.4,1X,F9.3,1X,F9.3,1XF9.3,1X,F7.1,1X,F7.1,1X,F9
     1.2,1X,F9.2,F7.4,1X,F7.3,1X,F7.3,1X,F8.2,F8.4,2X,I3)
      IF(I.EQ.1)GO TO 900
      IF(IBETA2(I).EQ.6)GO TO 700
      IF(NLINES+6.LE.60-LNCT)GO TO 400
      WRITE(NTAPE,1)
```

```
      LNCT=0                                                            S21*0060
400   LNCT=LNCT+NLINES+6                                                S21*0061
      WRITE(NTAPE,420)I,RPM2(I)                                         S21*0062
420   FORMAT(/,32X,7HSTATION,I3,42H IS AT THE EXIT OF A BLADE ROW ROTATIS21*0063
     1NG AT,F8.1,5H RPM.,//,1X,126HSTREAM   RELATIVE GAS ANGLES   RELATIS21*0064
     2VE VELOCITIES  RELATIVE MACH NO.S  LOSS   DE HALL  DIFFUS  DELTA PS21*0065
     3 BLADE SPEEDS   STREAM,/1X,29H-LINE  OPT.IN. INLET   OUTLET,4X,5HS21*0066
     4INLET,4X,6HOUTLET,6X,5HINLET,3X,6HOUTLET,4X,5HCOEFF 2X,6HNUMBER,3XS21*0067
     5,6HFACTOR,2X,6HUPON Q,3X,5HINLET,2X,6HOUTLET,2X,5H-LINE,/)        S21*0068
      Y=1.0                                                             S21*0069
      IF(RPM2(I).NE.0.0)Y=-1.0                                          S21*0070
      DO 500 J=1,NLINES                                                 S21*0071
      A2R=ATAN(TANBET(I,J))*57.296                                      S21*0072
      V1R=CMT(I-1,J)/COS(BET1(I-1,J)/57.296)                            S21*0073
      V2R=CMT(I,J)*SQRT(1.0+TANBET(I,J)**2)                             S21*0074
      X2R=XABS(J)*V2R/VABS(J)                                           S21*0075
      DHN=V2R/V1R                                                       S21*0076
      DFF=0.0                                                           S21*0077
      IF(IBETA2(I).EQ.1)GO TO 430                                       S21*0078
      DFF=1.0-DHN+Y*(CMT(I-1,J)*TAN(BET1(I-1,J)/57.296)-CMT(I,J)*TANBET(S21*0079
     1I,J))/(2.0*CUPONS(I,J)*V1R)                                       S21*0080
430   DPQ=(STATP(I,J)/STATP(I-1,J)-1.0)/((1.0+(GAMUP(J)-1.0)/2.0*RM1(I-1S21*0081
     1,J)**2)**(GAMUP(J)/(GAMUP(J)-1.0))-1.0)                           S21*0082
      U1=RPM2(I)*0.1047197*R(I-1,J)                                     S21*0083
      U2=U1*R(I,J)/R(I-1,J)                                             S21*0084
500   WRITE(NTAPE,510)J,STAGG(I-1,J),BET1(I-1,J),A2R,V1R,V2R,RM1(I-1,J),S21*0085
     1X2R,CLOS(I,J),DHN,DFF,DPQ,U1,U2,J                                 S21*0086
510   FORMAT(2X,I3,2X,F7.3,1X,F7.3,1X,F7.3,1X,F9.3,1X,F9.3,3X,F7.4,3X,F7S21*0087
     1.4,2X,F7.4,1X,F6.3,3X,F6.4,1X,F7.4,1X,F7.1,1X,F7.1,3X,I3)         S21*0088
700   IF(NLINES+7.LE.60-LNCT)GO TO 710                                  S21*0089
      WRITE(NTAPE,1)                                                    S21*0090
      LNCT=0                                                            S21*0091
710   LNCT=LNCT+NLINES+7                                                S21*0092
      DO 740 J=1,NLINES                                                 S21*0093
      GAMIN(J)=1.0/(1.0-GASK/(CJ*(CP1+CP2*STATT(1,J))))                 S21*0094
      GOJ(J)=(GAMIN(J)+GAMMA(J))/2.0                                    S21*0095
      GNJ(J)=(GAMUP(J)+GAMMA(J))/2.0                                    S21*0096
      RN(J)=PO(I,J)/PO(I-1,J)                                           S21*0097
      TN(J)=(TO(I,J)-TO(I-1,J))/TO(I-1,J)                               S21*0098
      RO(J)=PO(I,J)/PO(1,J)                                             S21*0099
740   TO(J)=(TO(I,J)-TO(1,J))/TO(1,J)                                   S21*0100
      CUMRN=0.0                                                         S21*0101
      CUMRO=0.0                                                         S21*0102
      CUMTO=0.0                                                         S21*0103
      CUMTN=0.0                                                         S21*0104
      GAMMN=0.0                                                         S21*0105
      GAMMO=0.0                                                         S21*0106
      DO 750 J=2,NLINES                                                 S21*0107
      CUMRN=CUMRN+(DELM(J)-DELM(J-1))*(RN(J)+RN(J-1))/2.0               S21*0108
      CUMRO=CUMRO+(DELM(J)-DELM(J-1))*(RO(J)+RO(J-1))/2.0               S21*0109
      CUMTO=CUMTO+(DELM(J)-DELM(J-1))*(TO(J)+TO(J-1))/2.0               S21*0110
      GAMMN=GAMMN+(DELM(J)-DELM(J-1))*(GNJ(J)+GNJ(J-1))/2.0             S21*0111
      GAMMO=GAMMO+(DELM(J)-DELM(J-1))*(GOJ(J)+GOJ(J-1))/2.0             S21*0112
750   CUMTN=CUMTN+(DELM(J)-DELM(J-1))*(TN(J)+TN(J-1))/2.0               S21*0113
      CUMEN=0.0                                                         S21*0114
      IF(CUMTN.NE.0.0)CUMEN=(CUMRN**((GAMMN-1.0)/GAMMN)-1.0)/CUMTN      S21*0115
      CUMEO=0.0                                                         S21*0116
      IF(CUMTO.NE.0.0)CUMEO=(CUMRO**((GAMMO-1.0)/GAMMO)-1.0)/CUMTO      S21*0117
      WRITE(NTAPE,760)CUMRN,CUMRO,CUMTN,CUMTO,CUMEN,CUMEO               S21*0118
760   FORMAT(/,47X,30HOVERALL PERFORMANCE PARAMETERS,//,1X,125HSTREAM  SS21*0119
```

```
      1TATION-TO-STATION-PARAMETERS  INLET-TO-STATION-PARAMETERS      MEAS21*0120
      2N PARAMETERS  STATION-TO-STATION  INLET-TO-STATION,/1X,86H-LINE  S21*0121
      3PRESSURE  DELTA T  ISENTROPIC  PRESSURE  DELTA T ISENTROPIC     PRS21*0122
      4ESSURE RATIO,7X,F8.4,12X,F8.4,/,11X,74HRATIO    ON T    EFFICIENCYS21*0123
      5    RATIO    ON T    EFFICIENCY      DELTA T ON T,9X,F8.4,12X,F8.4,/S21*0124
      6,73X,13HISEN. EFFICY.,9X,F7.4,13X,F7.4)                            S21*0125
       DO 800 J=1,NLINES                                                  S21*0126
       EN=0.0                                                             S21*0127
       IF(TN(J).NE.0.0)EN=(RN(J)**((GNJ(J)-1.0)/GNJ(J))-1.0)/TN(J)        S21*0128
       E0=0.0                                                             S21*0129
       IF(T0(J).NE.0.0)E0=(R0(J)**((G0J(J)-1.0)/G0J(J))-1.0)/T0(J)        S21*0130
       WRITE(NTAPE,910)J,RN(J),TN(J),EN,R0(J),T0(J),E0                    S21*0131
  800  CONTINUE                                                           S21*0132
  900  CONTINUE                                                           S21*0133
  910  FORMAT(1X,I3,5X,F8.4,1X,F8.4,3X,F7.4,4X,F8.4,1X,F8.4,2X,F7.4)     S21*0134
       RETURN                                                             S21*0135
       END                                                                S21*0136
```

```
      SUBROUTINE PUNCH(J1,J21,J22,J31,J32,J41,J42,J5,LTAPE1,LTAPE2,LTAPES22*0000
     13,LTAPE4,LTAPE5,NTHET)                                            S22*0001
      COMMON/CALC/FLOW,ICASE,KRPM,L,NCASE,NDIV,NDUM,NMASS,NPLET,NROT,NWRS22*0002
     1ET,ALPHA(30,15),CMT(30,15),PO(30,15),RCB(30,15),STATT(30,15),TO(30S22*0003
     2,15),WMT(30,15)                                                   S22*0004
      COMMON/DATA/CP1,CP2,GASK,HUBLOC,IFBL,IFCYL,IFRAD,IFRPM,IFSIMP,IFSTS22*0005
     1AT,IFTYPE,IGV,IMASS,INCPO,INCWRI,IPUNCH,IRAD,IROT,ITAPE,ITER,MTAPES22*0006
     2,NBETA,NCURVE,NLINES,NPASS,NPLOT,NPTS,NSTAGE,NTAPE,NTOPO,NWRIT,NX,S22*0007
     3TOLCX,VIS1,VIS2,ANGLN(30),BBP(30,15),BETA(30,15),BLAK(30,15),BLAM(S22*0008
     430),CLOSS(30,15),COMENT(12),CO(30,15),DELM(15),ESTFLO(1),EXPLOS(30S22*0009
     5),FLOWI(1,1),IBETA2(30),IFCAX(30),IFLVSI(30),IFMACH(30),IFMLOS(30)S22*0010
     6,IFPROF(30),IFREYL(30),IFREYN(30),IFTHIC(30),ILOSS(30),IMACHI(30),S22*0011
     7KSUMRY(1),RPM2(30),RHP(30),RSP(30),ROT(1),SIGMA(30,15),STAG(30,15)S22*0012
     8,WIDTH(1),X(30),XMACH(30),ZBAR(30,15),DELR(15)                    S22*0013
      COMMON/WENT/GJ,GJCP,GR2,CXM(30,15),Y(9)                           S22*0014
      COMMON/RESULT/DAMP,R(30,15)                                       S22*0015
      COMMON/TOGOSS/BET1(30,15),CLOS(30,15),CR(30,15),CUPONS(30,15),RM1(S22*0016
     130,15),STAGG(30,15),STATP(30,15),TANBET(30,15)                    S22*0017
      COMMON/NEW/RDIS(30,15),XMU(30,15),DPDTH(30,15),DFDTH(30,15),ARATIOS22*0018
     1(30,15)                                                           S22*0019
      IFSUM=1                                                           S22*0020
      NUNIT=LTAPE1                                                      S22*0021
      II1=1                                                             S22*0022
      II2=J1                                                            S22*0023
   10 WRITE(NUNIT,11)NPTS,ESTFLO(1)                                     S22*0024
   11 FORMAT(6X,I6,E13.6)                                               S22*0025
      WRITE(NUNIT,20)COMENT                                             S22*0026
   20 FORMAT(6X,12A6)                                                   S22*0027
      DO 65 J=1,NLINES                                                  S22*0028
      STATT(1,J)=TO(1,J)                                                S22*0029
   65 STATP(1,J)=PO(1,J)                                                S22*0030
      IF(NPTS.EQ.1)GO TO 70                                             S22*0031
      WRITE(NUNIT,25)NX,NLINES,NBETA,NTOPO,IMASS,IFSIMP,NPASS,IFBL,NPTS,S22*0032
     1IGV,IFRAD,IFRPM,ITER,IFSUM,NPLET,INCPO,NWRIT,INCWRI,IRAD,IPUNCH,IFS22*0033
     2TYPE                                                              S22*0034
   25 FORMAT(6X,5I6,/,6X,5I6,/,6X,11I6)                                 S22*0035
      IF(IFSUM.EQ.2)GO TO 35                                            S22*0036
      WRITE(NUNIT,30)(KSUMRY(I),I=1,IMASS)                              S22*0037
   30 FORMAT(6X,12I6)                                                   S22*0038
   35 WRITE(NUNIT,40)TOLCX,HUBLOC,GASK,VIS1,VIS2,CP1                    S22*0039
   40 FORMAT(2X,6E13.6)                                                 S22*0040
      DO 60 I=2,NX                                                      S22*0041
      WRITE(NUNIT,30)IBETA2(I),IFTHIC(I),IFCAX(I),IFMACH(I),IFREYN(I),ILS22*0042
     1OSS(I),IFMLOS(I),IFLVSI(I),IFPROF(I),IFREYL(I)                    S22*0043
      IF(IBETA2(I).EQ.6)GO TO 60                                        S22*0044
      IF((IBETA2(I).EQ.2.OR.IBETA2(I).EQ.4).AND.IFLVSI(I).EQ.1)WRITE(NUNS22*0045
     1IT,40)EXPLOS(I)                                                   S22*0046
      IF((IBETA2(I).EQ.2.OR.IBETA2(I).EQ.4).AND.IFMACH(I).EQ.3)WRITE(NUNS22*0047
     1IT,40)XMACH(I)                                                    S22*0048
      WRITE(NUNIT,40)(BBP(I,J),STAG(I,J),BETA(I,J),SIGMA(I,J),BLAK(I,J),S22*0049
     1ZBAR(I,J),CLOSS(I,J),CO(I,J),J=1,NBETA)                           S22*0050
   60 CONTINUE                                                          S22*0051
      WRITE(NUNIT,40)(DELR(I),I=1,NLINES),(BLAM(I),I=1,NX),(FLOWI(I,1),IS22*0052
     1=1,IMASS),ESTFLO(1),WIDTH(1)                                      S22*0053
      WRITE(NUNIT,40)(X(I),RHP(I),RSP(I),I=1,NX)                        S22*0054
      WRITE(NUNIT,40)(RPM2(I),I=2,NX)                                   S22*0055
      IF(IFTYPE.EQ.0)GO TO 70                                           S22*0056
      WRITE(NUNIT,30)(IMACHI(I),I=1,NX)                                 S22*0057
      WRITE(NUNIT,40)(ANGLN(I),I=1,NX)                                  S22*0058
   70 WRITE(NUNIT,40)((R(I,J),STATP(I,J),STATT(I,J),CMT(I,J),CXM(I,J),ALS22*0059
```

```
     1PHA(I,J),J=1,NLINES),I=II1,II2)                                   S22*0060
      IF(NUNIT.EQ.LTAPE5)RETURN                                         S22*0061
      IF(NUNIT.EQ.LTAPE1)GO TO 80                                       S22*0062
      IF(NUNIT.EQ.LTAPE2)GO TO 85                                       S22*0063
      IF(NUNIT.EQ.LTAPE3)GO TO 90                                       S22*0064
75    NUNIT=LTAPE5                                                      S22*0065
      IF(NTHET.EQ.1)REWIND NUNIT                                        S22*0066
      II1=J5                                                            S22*0067
      II2=NX                                                            S22*0068
      GO TO 70                                                          S22*0069
80    IF(J21.EQ.0)GO TO 75                                              S22*0070
      NUNIT=LTAPE2                                                      S22*0071
      IF(NTHET.EQ.1)REWIND NUNIT                                        S22*0072
      II1=J21                                                           S22*0073
      II2=J22                                                           S22*0074
      GO TO 70                                                          S22*0075
85    IF(J31.EQ.0)GO TO 75                                              S22*0076
      NUNIT=LTAPE3                                                      S22*0077
      IF(NTHET.EQ.1)REWIND NUNIT                                        S22*0078
      II1=J31                                                           S22*0079
      II2=J32                                                           S22*0080
      GO TO 70                                                          S22*0081
90    IF(J41.EQ.0)GO TO 75                                              S22*0082
      NUNIT=LTAPE4                                                      S22*0083
      IF(NTHET.EQ.1)REWIND NUNIT                                        S22*0084
      II1=J41                                                           S22*0085
      II2=J42                                                           S22*0086
      GO TO 70                                                          S22*0087
      END                                                               S22*0088
```

```
      SUBROUTINE THETA(J1,J21,J22,J31,J32,J41,J42,J5,LTAPE1,LTAPE2,
     1LTAPE3,LTAPE4,LTAPE5,JTAPE1,JTAPE2,JTAPE3,JTAPE4,JTAPE5,NTAPE,
     2KTAPE1,KTAPE2,IFSTAR)
C***** VERSION OF JULY 68
C***** INPUT FROM DISTORT - RADIN,PIN,TIN,VMIN,TANAIN,PHI
C***** OUTPUT TO DISTORT - RADOUT,RREF,AMU,DPDTH,DFIDTH,DTHETA,DRCTH
C***** THIS VERSION MODIFIES THE FLOWFIELD UPSTREAM AND DOWNSTREAM OF
C***** THE COMPRESSOR (KATZ METHOD EXTENDED BY GES)
C
      COMMON /GEN/TITLE(12),NTHETA,LP,NLOOPS,IFPRIN,R32(18)
C
      COMMON/SECTOR/IFFULL,EF(18),CARD(12,18),I1,I2,I3,I4,I5,I6,I7,I8,
     1I9,I10,I11,I12,I13,I14,I15,I16,I17,I18,I19,I20,I21,I22,R1,R2,R3,
     2R4,R5,R6,I23(30),I24(30),I25(30),I26(30),I27(30),I28(30),I29(30),
     3I30(30),I31(30),I32(30),R13(30,15),R14(30,15),R15(30,15),
     4R16(30,15),R17(30,15),R18(30,15),R19(30,15),R20(30,15),R21(15),
     5R22(30),R23,R24,R25,R26(30),R27(30),R28(30),R29(30),I33(30),
     6R30(30),EXPLOS(30),XMACH(30)
C
      COMMON/ARRAY/RADIN(15,13,7),PIN(15,13,7),TIN(15,13,7),VMIN(15,13,7
     1),TANAIN(15,13,7),PHIN(15,13,7)
C
      COMMON/AUXIL/RREF(15,7),ZREF(15,7),THETAS(13,7),THETAP(13,7),
     1SEARAT(13,7),PREV(13,7),DPDTHS(13,7),DRCTHS(13,7),PTOTAL(13),
     2PSTAT(13),RADUM(15),PDUM(15),TDUM(15),VMDUM(15),PHIDUM(15),
     3TANDUM(15),VS(13),WS(13),DPREV(13,7)
      COMMON/ONE/NF,CNZ(36),DNZ(36),XL1,CNF(36),DNF(36),XL2,IND2,ENF(36)
     1,FNF(36),XL3,IND3,WZ,VZ,WCAP1,VCAP1,XK1,WCAP2,VCAP2,XK2,SS,EP,
     2VMQ(2),AMQ(2,36),BMQ(2,36),CMQ(2,36),DMQ(2,36,2),EMQ(2,36,2),
     3FMQ(2,36,2)
C
      DIMENSION THETA1(15,13),DPDTH1(15,13),THETAR(15,13,7),AMU(15,13,7)
     1,DPDTH(15,13,7),DTHETA(15,13,7),DRCTH(15,13,7),DFIDTH(15,13,7),
     2RADOUT(15,13,7),RINLET(15,13),PINLET(15,13),TINLET(15,13),
     3AINLET(15,13),FINLET(15,13),THETMP(15,13),THETI(13),THETI1(13)
C
      EQUIVALENCE (RADIN(1),RADOUT(1),RINLET(1),DPDTH1(1)),(PIN(1),
     1DPDTH(1),PINLET(1)),(TIN(1),TINLET(1),DTHETA(1)),(VMIN(1),DRCTH(1)
     2,AINLET(1)),(TANAIN(1),AMU(1),THETAR(1)),(PHIN(1),DFIDTH(1),
     3THETA1(1),FINLET(1)),(THETMP(1),AMU(1171)),(I1,IZAMAX),(I2,NLINES)
     4,(RADUM(1),THETI1(1)),(TDUM(1),THETI(1))
C
      COMMON/OV/IZR1,IZR2,IZA1,GUMBO,LOG1
C
      TOTAL(T,V,A)= 1.0 + (1.0 + A**2)/CP/T*V**2
C***** SECTION ONE - INPUT SECTION
C
      WRITE(NTAPE,1001)
1001  FORMAT(1H1,60X,13HPROGRAM THETA,/,61X,13H*************)
C
C***** DETERMINE PRINCIPAL STATIONS,INITAL TAPES,DEFINE CONSTANTS
C
      LTAPE= LTAPE1
      JTAPE= JTAPE1
      IZR1=2
      IZALE= J1
      IZATE= J5
      IZA1= 1
      IZA2= J1
      IZR2= IZA2-IZA1+IZR1
```

```
      RADIAN= 180.0/3.1415926
      GZERO= 32.174
      WORKJ= 778.17
C
C***** READ GENERAL INPUT
C
      READ(LTAPE,1003)TITLE,NTHETA,LP,NLOOPS,IFPRIN,RFAC1,RFAC2,RFAC3,
     1RFAC4,IFAC1,IFAC2,(R32(N),N=1,NTHETA)
1003  FORMAT(12A6,/,4I6,/,4F8.5,/,2I6,/,(9F8.4))
      RFAC4= IFAC2
      RF1= 1.0 - RFAC1**(1.0/RFAC4)
      WRITE(NTAPE,1004)TITLE,NTHETA,(R32(I),I= 1,NTHETA)
1004  FORMAT(35X,12A6/20X,19HNUMBER OF SECTORS =,I3,/,20X,25HSECTOR CENT
     1ER-LINE THETAS,5X,F8.4,/(50X,F8.4))
      WRITE(NTAPE,9901)
9901  FORMAT(1H1)
      WRITE(NTAPE,9902) (N, N= 1,NTHETA)
9902  FORMAT(53X,24HLOCATION OF CENTER-LINES,/,53X,24H******************
     1******,//,62X,6HSECTOR,/,62X,6H******,/,27X,18I6)
      WRITE(NTAPE,9903) (R32(N), N= 1,NTHETA)
9903  FORMAT(/,1X,10HSTATION  1,6X,42H(THE CENTER-LINES ARE RADIAL IN TH
     1E INLET),/,1X,10H**********,//,28X,17F6.1)
C
C***** READ COMPRESSOR DATA AND STREAMLINE DATA
C
1005  DO 1022 N= 1,NTHETA
      IF(LTAPE .NE. LTAPE1) GO TO 1021
C
      READ(LTAPE,1006)IFFULL,EF(N),(CARD(J,N),J= 1,12)
1006  FORMAT(6X,I6,E13.6,/,6X,12A6)
      IF(IFFULL.EQ.1)GO TO 1021
C
      READ(LTAPE,1008)I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11,I12,I13,I14,I15
     1,I16,I17,I18,I19,I20,I21,I22
1008  FORMAT(6X,5I6,/,6X,5I6,/,6X,11I6,/,6X,I6)
      READ(LTAPE,1009)R1,R2,R3,R4,R5,R6
1009  FORMAT(2X,6E13.6)
      CP= WORKJ*R6
      GUMBO= CP/R3
      CP= 2.0*CP*GZERO
      R4= R4*1.0E6
      R5= R5*1.0E6
C
      DO 1014 I=2,I1
      READ(LTAPE,1011)I23(I),I24(I),I25(I),I26(I),I27(I),I28(I),I29(I),
     1I30(I),I31(I),I32(I)
1011  FORMAT(6X,12I6)
      IF(I23(I).EQ.6)GO TO 1014
      IF((I23(I) .EQ. 2 .OR. I23(I) .EQ. 4) .AND. I30(I) .EQ. 1)
     1READ(LTAPE,1009)EXPLOS(I)
      IF((I23(I) .EQ. 2 .OR. I23(I) .EQ. 4) .AND. I26(I) .EQ. 3)
     1READ(LTAPE,1009) XMACH(I)
      READ(LTAPE,1009)(R13(I,J),R14(I,J),R15(I,J),R16(I,J),R17(I,J),
     1R18(I,J),R19(I,J),R20(I,J),J= 1,I3)
1014  CONTINUE
      READ(LTAPE,1009)(R21(I),I=1,I2),(R22(I),I=1,I1),R23,R24,R25
      READ(LTAPE,1009)(R26(I),R27(I),R28(I),I=1,I1)
      READ(LTAPE,1009)(R29(I),I=2,I1)
      IF(I21.EQ.0)GO TO 1021
C
```

```
      READ(LTAPE,1011)(I33(I),I=1,I1)
      READ(LTAPE,1009)(R30(I),I=1,I1)
1021  READ(LTAPE,1009)((RADIN(J,N,I),PIN(J,N,I),TIN(J,N,I),VMIN(J,N,I),
     1TANAIN(J,N,I),PHIN(J,N,I),J= 1,I2),I= IZR1,IZR2)
1022  CONTINUE
C
C***** SECTION TWO - PRELIMINARY CALCULATIONS
C
      IF(LTAPE.NE.LTAPE1)GO TO 2027
      XNTHET= NTHETA
      DO 2025 N=1,NTHETA
      R32(N)= R32(N)/RADIAN
      THETAS(N,IZR1)= R32(N)
      DO 2025 J=1,NLINES
      PINLET(J,N)= PIN(J,N,IZR1)
      TINLET(J,N)= TIN(J,N,IZR1)
      TIN(J,N,IZR1)= TOTAL(-TINLET(J,N),VMIN(J,N,IZR1),TANAIN(J,N,IZR1))
      PIN(J,N,IZR1)= PINLET(J,N)*TIN(J,N,IZR1)**GUMBO
      TIN(J,N,IZR1)= TINLET(J,N)*TIN(J,N,IZR1)
      AINLET(J,N)= ATAN(TANAIN(J,N,IZR1))*RADIAN
      THETA1(J,N)= R32(N)
2025  CONTINUE
      DO 2026 J= 1,NLINES
      DO 2026 N= 1,NTHETA
      ROMEAN= PIN(J,N,IZR1)/R3/TIN(J,N,IZR1)/GZERO
      DPDTH1(J,N)= DERIV(PIN,THETAS,NTHETA,J,N,IZR1)/ROMEAN
2026  CONTINUE
      LOG1=NTAPE
      CALL OVERAL
C
C***** COMPUTE REFERENCE RADII AND AXIAL COORDINATES
C
2027  DO 2033 I= IZR1,IZR2
      IZA= I-IZR1+IZA1
      TAN30= TAN(R30(IZA)/RADIAN)
      DO 2033 J=1,NLINES
      RREF(J,I)= 0.0
      DO 2030 N=1,NTHETA
      RREF(J,I)= RREF(J,I) + RADIN(J,N,I)
2030  CONTINUE
      RREF(J,I)=RREF(J,I)/XNTHET
      ZREF(J,I)=R26(IZA)+TAN30*RREF(J,I)
2033  CONTINUE
      CALL OVLTWO
C
C***** INTERPOLATE DATA ARRAYS AT REFERENCE RADII
C
      DO 2039 I=IZR1,IZR2
      DO 2039 N=1,NTHETA
      THETAP(N,I)= R32(N)
      DPREV(N,I)= 0.0
      DO 2037 J=1,NLINES
      RADUM(J)= RADIN(J,N,I)
      PDUM(J)= PIN(J,N,I)
      TDUM(J)= TIN(J,N,I)
      VMDUM(J)= VMIN(J,N,I)
      TANDUM(J)= TANAIN(J,N,I)
      PHIDUM(J)=PHIN(J,N,I)
2037  CONTINUE
      DO 2039 J=1,NLINES
```

```
      PIN(J,N,I)= FINDY(RREF(J,I),RADUM,PDUM,NLINES)
      TIN(J,N,I)= FINDY(RREF(J,I),RADUM,TDUM,NLINES)
      VMIN(J,N,I)= FINDY(RREF(J,I),RADUM,VMDUM,NLINES)
      PHIN(J,N,I)= FINDY(RREF(J,I),RADUM,PHIDUM,NLINES)
      TANAIN(J,N,I)= FINDY(RREF(J,I),RADUM,TANDUM,NLINES)
2039  CONTINUE
C
C***** SET UP J-LOOP
C
      J=1
2041  ICOUNT= 0
2042  I= IZR1
      IF(LTAPE.EQ.LTAPE1)I=IZR1+1
      IF(ICOUNT.GT.0)GO TO 2046
      IF(LP.GT.1.OR.IFSTAR.NE.0)READ(KTAPE1)((THETAP(N,L),N=1,NTHETA),L=
     1IZR1,IZR2),((DPREV(N,L),N=1,NTHETA),L=IZR1,IZR2)
C
2046  DO 2047 N= 1,NTHETA
      THETI1(N)= THETA1(J,N)
      DPDTHS(N,I-1)= DPDTH1(J,N)
2047  CONTINUE
2048  KATZ= 0
      IZA=I-IZR1+IZA1
C
C***** SECTION THREE - THETA COMPUTATION
C
3050  DO 3056 N= 1,NTHETA
      DR= RREF(J,I) - RREF(J,I-1)
      DM= SQRT(DR**2 + (ZREF(J,I)-ZREF(J,I-1))**2)
      RM= RREF(J,I) + RREF(J,I-1)
      IF(ABS(DR/RM)-0.0001)3052,3052,3054
3052  TANRDM= (TANAIN(J,N,I) + TANAIN(J,N,I-1))*DM/RM
      GO TO 3055
3054  TANRDM= (TANAIN(J,N,I) - TANAIN(J,N,I-1) + (RREF(J,I)*TANAIN(J,N,I
     1-1) - RREF(J,I-1)*TANAIN(J,N,I))/DR*ALOG(RREF(J,I)/RREF(J,I-1)))
     2*DM/DR
3055  THETI(N)= THETI1(N) + TANRDM
3056  CONTINUE
C
C***** DAMPING OF THETA
C
      DO 3057 N= 1,NTHETA
      THETAS(N,I)= (1.0-RFAC2)*THETI(N) + RFAC2*THETAP(N,I)
3057  CONTINUE
C
C***** SECTOR WIDTH RATIO
C
      DO 3059 N=1,NTHETA
      SEARAT(N,I)= SERVUS(THETAS,R32,NTHETA,N,I)
      IF(SEARAT(N,I) .GT. 2.0 .OR. SEARAT(N,I) .LT. 0.5) GO TO 3060
3059  CONTINUE
      GO TO 3068
C
C***** IF STREAMLINES CONVERGE OR DIVERGE EXCESSIVELY CORRECT TANAIN
C
3060  CONTINUE
      NOTE= N
      WRITE(NTAPE,3061) NOTE,J,IZA,(N,N= 1,NTHETA),(TANAIN(J,N,I),N= 1,
     1NTHETA)
3061  FORMAT(1H1/16X,26HTHE WIDTH RATIO OF SECTOR ,I2,15H AT STREAMLINE
```

```
     1,I2, 9H STATION ,I2,44H IS EITHER GREATER THAN 2.0 OR LESS THAN 0.
     25/16X,44HTAN(ALPHA) BEFORE AND AFTER MODIFICATION ARE//16X,6HSECTO
     3R,18I6/4X,17HORIGINAL TANALPHA,3X,18F6.4)
      AVTAN= 0.0
      DO 3063 N=1,NTHETA
      AVTAN= AVTAN + TANAIN(J,N,I)/RREF(J,I)+TANAIN(J,N,I-1)/RREF(J,I-1)
3063  CONTINUE
      AVTAN= AVTAN/XNTHET
      DO 3065 N=1,NTHETA
      TANAIN(J,N,I)= RREF(J,I)*(AVTAN-TANAIN(J,N,I-1)/RREF(J,I-1))
3065  CONTINUE
      WRITE(NTAPE,3066) (TANAIN(J,N,I),N= 1,NTHETA)
3066  FORMAT(4X,17HMODIFIED TANALPHA,3X,18F6.4)
      GO TO 3050
C
C***** TRANSFER STORAGE OF UNDAMPED THETA(I) TO THETA(I-1)
C
3068  CONTINUE
      DO 3070 N=1,NTHETA
      THETI1(N)= THETI(N)
3070  CONTINUE
C
C***** CHECK ON REVISION OF THETAS
C
      IF(IZA .EQ. IZALE) KATZ= 1
      IF(IZA .EQ. I1) KATZ= 2
      IF(KATZ .EQ. 0 .OR. RFAC5 .EQ. 1.0) GO TO 5121
C
C***** SECTION FOUR - KATZ LOOP
C
      IF(KATZ.EQ.2.AND.ICOUNT.GT.0)GO TO 4097
C
C***** DEFINE AVERAGE VALUES OF RHO,CM,CTHETA AT BOUNDARY STATIONS
C
      ROME1= XMEAN(PIN,THETAS,NTHETA,J,IZR1)/R3/XMEAN(TIN,THETAS,NTHETA,
     1J,IZR1)/GZERO
      IF(KATZ .EQ. 1) ROME2= XMEAN(PIN,THETAS,NTHETA,J,IZR2)/R3/
     1XMEAN(TIN,THETAS,NTHETA,J,IZR2)/GZERO
      CMEAN1= XMEAN(VMIN,THETAS,NTHETA,J,IZR1)
      CMEAN2= XMEAN(VMIN,THETAS,NTHETA,J,IZR2)
      AMEAN1= XMEAN(TANAIN,THETAS,NTHETA,J,IZR1)
      CTHET1= AMEAN1*CMEAN1
      CTHET2= XMEAN(TANAIN,THETAS,NTHETA,J,IZR2)*CMEAN2
      IF(ICOUNT.GT.0)GO TO 4077
C
C***** STORE UNMODIFIED PRESSURE,TANAIN
C
      DO 4076 I=IZR1,IZR2
      DO 4076 N=1,NTHETA
      IF(KATZ .EQ. 1) PREV(N,I)= PIN(J,N,I)
      IF(KATZ .EQ. 2) PREV(N,I)= TANAIN(J,N,I)
4076  CONTINUE
      NF= 2*NTHETA
      ISTART= IZR1 + 1
C
C***** DEFINE BOUNDARY FUNCTIONS UPSTREAM AND DOWNSTREAM
C
4077  DO 4081 N= 1,NTHETA
      IF(KATZ-1) 4078,4078,4080
4078  PTOTAL(N)= PIN(J,N,1)/ROME1
```

```
      PSTAT(N)= PREV(N,IZR2)/ROME2
      GO TO 4081
4080  PTOTAL(N)= PIN(J,N,IZR1)*TOTAL(TIN(J,N,IZR1),VMIN(J,N,IZR1),
     1TANAIN(J,N,IZR1))**GUMBO/ROME1
      PSTAT(N)= ATAN((PREV(N,IZR1)-AMEAN1)/(1.0+PREV(N,IZR1)*AMEAN1))*
     1(1.0 + AMEAN1**2)*CMEAN1**2
4081  CONTINUE
C
C***** DEFINE FOURIER COEFFICIENTS CNZ,DNZ,CNF,DNF
C
      CALL FOUR(PTOTAL,THETAS,NTHETA,AZERO,DNZ,CNZ,NF,IZR1)
      IF(KATZ .EQ. 1) I= IZR2
      IF(KATZ .EQ. 2) I= IZR1
      CALL FOUR(PSTAT,THETAS,NTHETA,AZERO,DNF,CNF,NF,I)
C
C***** COMPUTE XL1,XL2,IND2,ENF,XL3,IND3,WZ,VZ,WCAP1,VCAP1,XK1,WCAP2,
C***** VCAP2,XK2
C
      XL1=0.0
      DO 4087 I=ISTART,IZR2
      DR= RREF(J,I)-RREF(J,I-1)
      RM= 0.5*(RREF(J,I) + RREF(J,I-1))
      DZ= ZREF(J,I) - ZREF(J,I-1)
      XL1= XL1 - SQRT(DR**2 + DZ**2)/RM
4087  CONTINUE
      IF(KATZ-1) 4088,4088,4092
4088  XL2= 0.0
      IND2= 0
      XL3= XL1
      IND3= 1
      CALL NEWRAP(CMEAN1,0.0,0.0,CMEAN2,XL1,0.0,0.0,0.0,0)
      CALL NEWRAP(CTHET1,0.0,0.0,CTHET2,XL1,0.0,0.0,0.0,1)
      GO TO 4093
4092  XL3= -XL1
      XL1= 0.0
      XL2= 0.0
      IND2= 1
      IND3= 0
      CALL NEWRAP(CMEAN1,0.0,0.0,CMEAN2,0.0,0.0,0.0,XL3,0)
      CALL NEWRAP(CTHET1,0.0,0.0,CTHET2,0.0,0.0,0.0,XL3,1)
4093  DO 4094 N= 1,NF
      ENF(N)= 0.0
      FNF(N)= 0.0
4094  CONTINUE
C
C***** SUBROUTINE SMITH COMPUTES THE REST OF COMMON/ONE/
C
      CALL SMITH
C
      IF(KATZ-1)4106,4106,4099
 4097 IF(ICOUNT .GE. IFAC2) GO TO 4108
C
C***** DETERMINE NEW ALPHA IN DOWNSTREAM REGION DIRECTLY
C
4099  DO 4103 I= IZR1,IZR2
      ZZ= 2.0*(ZREF(J,I)-ZREF(J,IZR1))/(RREF(J,I)+RREF(J,IZR1))
      CALL GEORGE(THETAS(1,I),ZZ,WS,VS,NTHETA)
      DO 4103 N=1,NTHETA
      V= VBARF(ZZ) + VS(N)
      W= WBARF(ZZ) + WS(N)
```

```
      TANAIN(J,N,I)=(1.0-RFAC5)*V/W+RFAC5*PREV(N,I)
4103  CONTINUE
      IF(ICOUNT.NE.0)GO TO 4105
      DO 102 I=IZR1,IZR2
      IP=I-IZR1+IZA1
      DO 101 N=1,NTHETA
      ENF(N)=ATAN(PREV(N,I))*RADIAN
101   FNF(N)=ATAN(TANAIN(J,N,I))*RADIAN
      WRITE(NTAPE,4104)J,IP,(ENF(N),N=1,NTHETA)
4104  FORMAT(1H0,35X,47HORIGINAL AND MODIFIED FLOW ANGLES,STREAMSURFACE,
     1I3,8H STATION,I3,//,5X,13F9.2)
102   WRITE(NTAPE,4114)(FNF(N),N=1,NTHETA)
4105  CONTINUE
C
      ICOUNT=ICOUNT+1
      GO TO 2042
C
C***** COMPUTE PRESSURE RELAXATION FACTOR, TANAIN RELAXING FACTOR
C
4106  IF(ICOUNT .GT. 0) GO TO 4108
      RFAC5=0.0
      L1=IFAC1
      YOUTEL= LP-L1
      IF(YOUTEL .GT. 0.0) RFAC5= YOUTEL*RFAC3
      IF(RFAC5 .GT. 1.0) RFAC5= 1.0
      IF(RFAC5 .EQ. 1.0) GO TO 5121
4108  CONTINUE
C
C***** COMPUTE STATIC PRESSURES AND RELAX
C
      DO 4118 I=IZR1,IZR2
      IF(KATZ .EQ. 1) ZZ= 2.0*(ZREF(J,I)-ZREF(J,IZR2))/(RREF(J,I)+
     1RREF(J,IZR2))
      IF(KATZ .EQ. 2) ZZ= 2.0*(ZREF(J,I)-ZREF(J,IZR1))/(RREF(J,I)+
     1RREF(J,IZR1))
      CALL EDWIN(THETAS(1,I),ZZ,PSTAT,PTOTAL,NTHETA)
      DM= SQRT((RREF(J,I)-RREF(J,I-1))**2+(ZREF(J,I)-ZREF(J,I-1))**2)
      PMEAN= XMEAN(PIN,THETAS,NTHETA,J,I)
      ROMEAN= PMEAN/R3/XMEAN(TIN,THETAS,NTHETA,J,I)/GZERO
      DO 4112 N=1,NTHETA
      IF(KATZ .EQ. 1)
     1PIN(J,N,I)=(PMEAN+PSTAT(N)*ROMEAN)*(1.0-RFAC5) + PREV(N,I)*RFAC5
      IF(KATZ .EQ. 2)
     1PIN(J,N,I)=(PMEAN+PSTAT(N)*ROMEAN)*(1.0-RFAC5) + PIN(J,N,I)*RFAC5
4112  CONTINUE
      IF(ICOUNT.NE.0.OR.KATZ.NE.1)GO TO 4115
      IP=I-1
      WRITE(NTAPE,4113)J,IP,(PREV(N,I),N=1,NTHETA)
4113  FORMAT(1H0,35X,52HORIGINAL AND MODIFIED STATIC PRESSURES,STREAMSUR
     1FACE,I3,8H STATION,I3,//,5X,13F9.2)
      WRITE(NTAPE,4114)(PIN(J,N,I),N=1,NTHETA)
4114  FORMAT(5X,13F9.2)
4115  CONTINUE
C
C***** COMPUTE DPDTHS,DRCTHS,TANAIN FOR THE KATZ LOOP
C
      DO 4117 N=1,NTHETA
      ROMEAN= PIN(J,N,I)/R3/TIN(J,N,I)/GZERO
      DPDTHS(N,I)=DERIV(PIN,THETAS,NTHETA,J,N,I)/ROMEAN*(1.0-RFAC1)
     1+DPREV(N,I)*RFAC1
```

```
      IF(I.EQ.IZR1)GO TO 4117
      DRCTHS(N,I)=0.5*(DPDTHS(N,I)/VMIN(J,N,I)+DPDTHS(N,I-1)/
     1VMIN(J,N,I-1))*DM
      IF(ICOUNT .GE. IFAC2) GO TO 4117
      TANAIN(J,N,I)=(-DRCTHS(N,I)+RREF(J,I-1)*VMIN(J,N,I-1)*TANAIN(
     1J,N,I-1))/RREF(J,I)/VMIN(J,N,I)
4117  CONTINUE
4118  CONTINUE
C***** BRANCHPOINT FOR KATZ-LOOP
      ICOUNT=ICOUNT+1
      IF(ICOUNT - IFAC2) 2042,2042,6129
C
C***** SECTION FIVE - REGULAR COMPUTATION OF DPDTHS AND DRCTHS
C
5121  CONTINUE
      IF(I.NE.IZR2)GO TO 6128
      DO 5127 I=IZR1,IZR2
      IZA= I-IZR1+IZA1
      DM= SQRT((RREF(J,I)-RREF(J,I-1))**2+(ZREF(J,I)-ZREF(J,I-1))**2)
      DO 5127 N=1,NTHETA
      ROMEAN= PIN(J,N,I)/R3/TIN(J,N,I)/GZERO
      DPDTHS(N,I)=DERIV(PIN,THETAS,NTHETA,J,N,I)/ROMEAN*(1.0-RFAC1)
     1 + DPREV(N,I)*RFAC1
      DPREV(N,I)= (1.0 - RF1)*DPREV(N,I) + DERIV(PIN,THETAS,NTHETA,J,N,I
     1)/ROMEAN*RF1
      IF(I23(IZA) .NE. 6) GO TO 5126
      DRCTHS(N,I)= 0.5*(DPDTHS(N,I)/VMIN(J,N,I) + DPDTHS(N,I-1)/VMIN(J,N
     1,I-1))*DM
      IF(ICOUNT .GE. IFAC2) GO TO 5126
      TANAIN(J,N,I)=(-DRCTHS(N,I) + RREF(J,I-1)*VMIN(J,N,I-1)*TANAIN(J,N
     1,I-1))/RREF(J,I)/VMIN(J,N,I)
 5126 DPDTHS(N,I)= DPREV(N,I)
5127  CONTINUE
      ICOUNT= ICOUNT + 1
      IF(ICOUNT - IFAC2) 2042,2042,6129
C
C***** SECTION SIX - INCREMENT I AND TRANSFER DATA
C
6128  I= I+1
      IF(I .LE. IZR2) GO TO 2048
C
C***** COMPLETE DFIDTH, TRANSFER TANA1,THETA1,DPDTH1,DPDTH,DRCTH,DTHET
C
6129  CONTINUE
      DO 6140 I=IZR1,IZR2
      RREF(J,I-1)=RREF(J,I)
      ZREF(J,I-1)=ZREF(J,I)
      DO 6140 N=1,NTHETA
      RADOUT(J,N,I-1)=RADIN(J,N,I)
      DFIDTH(J,N,I-1)=DERIV(PHIN,THETAS,NTHETA,J,N,I)
      THETAR(J,N,I-1)=THETAS(N,I)
      IF(LTAPE .EQ. LTAPE1 .AND. I .EQ. IZR1) GO TO 6140
      DPDTH(J,N,I-1)=DPDTHS(N,I)
      DTHETA(J,N,I-1)=SEARAT(N,I)
      DRCTH(J,N,I-1)=DRCTHS(N,I)
      IF(I .LT. IZR2) GO TO 6140
      RADIN(J,N,I)= DPDTHS(N,I)
      PHIN(J,N,I)= TANAIN(J,N,I)
      PIN(J,N,I)= THETI(N)
6140  CONTINUE
```

```
      WRITE(KTAPE2)((THETAS(N,I),N=1,NTHETA),I=IZR1,IZR2),
     1((DPDTHS(N,I),N= 1,NTHETA), I= IZR1,IZR2)
C
C***** INCREMENT J-LOOP
C
      J= J+1
      IF(J.LE.NLINES)GO TO 2041
      DO 9910 I= IZA1,IZA2
      IZR= I-IZA1+1
      IF(I .EQ. 1) GO TO 9910
      WRITE(NTAPE,9904) I
 9904 FORMAT(/,1X,7HSTATION,I3,/,1X,10H**********)
      DO 9909 J=1,NLINES
      DO 9907 N=1,NTHETA
      PDUM(N)= THETAR(J,N,IZR)*RADIAN
 9907 CONTINUE
      WRITE(NTAPE,9908) J,RREF(J,IZR),(PDUM(N), N= 1,NTHETA)
 9908 FORMAT(/,2X,4HLOCN,I3,3X,8HRADIUS =,F7.3,1X,18F6.1)
 9909 CONTINUE
 9910 CONTINUE
C
C***** SECTION SEVEN - COMPUTE MU, DETERMINE NEXT TAPES AND STATIONS
C
      DO 7154 I=IZR1,IZR2
      DO 7146 N=1,NTHETA
      DO 7146 J=1,NLINES
      THETMP(J,N)= THETAR(J,N,I-1)
 7146 CONTINUE
      DO 7154 N=1,NTHETA
      COS1= COS(THETMP(1,N))
      SIN1= SIN(THETMP(1,N))
      X1=RREF(1,I-1)*COS1
      Y1=RREF(1,I-1)*SIN1
      AM1= 0.0
      DO 7154 J=2,NLINES
      COS2= COS(THETMP(J,N))
      SIN2= SIN(THETMP(J,N))
      X2= RREF(J,I-1)*COS2
      Y2= RREF(J,I-1)*SIN2
      DUM= SQRT((X2-X1)**2 + (Y2-Y1)**2)
      X12= (X2-X1)/DUM
      Y12= (Y2-Y1)/DUM
      AM1=ATAN((COS1*Y12-SIN1*X12)/(COS1*X12+SIN1*Y12))+AM1
      AM2=ATAN((COS2*Y12-SIN2*X12)/(COS2*X12+SIN2*Y12))
      AMU(J-1,N,I-1)=AM1/2.0
      IF(J.EQ.2)AMU(1,N,I-1)=AM1
      IF(J.EQ.NLINES)AMU(J,N,I-1)=AM2
      AM1= AM2
      COS1= COS2
      SIN1= SIN2
      X1= X2
      Y1= Y2
 7154 CONTINUE
C
      I4=I2
      I12= IFPRIN
      I19= 1
C
C***** DETERMINE NEXT TAPE UNITS
C
```

```
      IF(IZA+1.NE.J21)GO TO 7156
      NIZA1=J21
      NIZA2=J22
      NEWJ=JTAPE2
      NEWL=LTAPE2
      GO TO 7159
7156  IF(IZA2+1 .NE. J31) GO TO 7157
      NIZA1=J31
      NIZA2=J32
      NEWJ=JTAPE3
      NEWL=LTAPE3
      GO TO 7159
7157  IF(IZA2+1 .NE. J41) GO TO 7158
      NIZA1=J41
      NIZA2=J42
      NEWJ=JTAPE4
      NEWL=LTAPE4
      GO TO 7159
7158  CONTINUE
      NIZA1=J5
      NIZA2=I1
      NEWL= LTAPE5
      NEWJ= JTAPE5
7159  CONTINUE
C
C***** SECTION EIGHT - OUTPUT
C
      DO 8177 N=1,NTHETA
C
      IF(JTAPE .NE. JTAPE1) GO TO 8165
      WRITE(JTAPE,8161) (CARD(J,N), J= 1,12)
8161  FORMAT(1X,12A6)
      IZA1= 2
      IZR1= 3
      I9= 0
      IF(N .GT. 1) I9= 1
      WRITE(JTAPE,8162)I1,I2,I3,I4,I5,I6,I7,I8,I9,I10,I11,I12,I13,I14,
     1I15,I16,I17,I18,I19,I20,I21,I22
8162  FORMAT(1X,I5,4I6,/,1X,I5,4I6,/,1X,I5,10I6,/,1X,I5)
      WRITE(JTAPE,8163)R1,R2,R3,R6,R4,R5
8163  FORMAT(1X,F7.4,2F8.4,F8.5,8X,2F8.4)
      WRITE(JTAPE,8164) (RINLET(J,N),TINLET(J,N),PINLET(J,N),AINLET(J,N),
     1FINLET(J,N), J= 1,NLINES)
8164  FORMAT(1X,F7.3,2F8.2,F8.4,8X,F8.4)
C
8165  WRITE(JTAPE,8166) IZA1,IZA2
8166  FORMAT(1X,I5,9I6)
C
      DO 8174 I=IZA1,IZA2
      IZR= I-IZA1+IZR1-1
      WRITE(JTAPE,8166) I23(I),I24(I),I25(I),I26(I),I27(I),I28(I),I29(I)
     1,I30(I),I31(I),I32(I)
      IF(I23(I) .EQ. 6) GO TO 8172
      IF((I23(I) .EQ. 2 .OR. I23(I) .EQ. 4) .AND. I30(I) .EQ. 1)
     1WRITE(JTAPE,8168) EXPLOS(I)
      IF((I23(I) .EQ. 2 .OR. I23(I) .EQ. 4) .AND. I26(I) .EQ. 3)
     1WRITE(JTAPE,8168) XMACH(I)
8168  FORMAT(1X,F7.4)
      WRITE(JTAPE,8169) (R13(I,J),R14(I,J),R15(I,J),R16(I,J),R17(I,J),R18
     1(I,J),R19(I,J),R20(I,J),J=1,I3)
```

```
8169  FORMAT(1X,F7.4,7F8.4)
      WRITE(JTAPE,8170)(RADOUT(J,N,IZR),RREF(J,IZR),AMU(J,N,IZR),
     1DPDTH(J,N,IZR),DFIDTH(J,N,IZR),DTHETA(J,N,IZR),J= 1,NLINES)
8170  FORMAT(1X,F7.4,F8.4,F8.5,F8.0,F8.4,F8.5)
      GO TO 8174
C
8172  WRITE(JTAPE,8173)(RADOUT(J,N,IZR),RREF(J,IZR),AMU(J,N,IZR),
     1DPDTH(J,N,IZR),DFIDTH(J,N,IZR),DTHETA(J,N,IZR),DRCTH(J,N,IZR),
     2J= 1,NLINES)
8173  FORMAT(1X,F7.4,F8.4,F8.5,F8.0,F8.4,F8.5,F8.1)
8174  CONTINUE
8175  IF(JTAPE .NE. JTAPE5) WRITE(JTAPE,8176) NEWJ
8176  FORMAT(1X,I5)
C
      IF(JTAPE .NE. JTAPE5) GO TO 8185
      WRITE(JTAPE,8178) (R21(J), J= 1,NLINES)
8178  FORMAT(1X,F7.4,8F8.4)
      WRITE(JTAPE,8179)(R22(J), J= 1,I1)
8179  FORMAT(1X,F7.4,8F8.4)
      WRITE(JTAPE,8180)R23,EF(N),R25,(R26(I),R27(I),R28(I), I= 1,I1)
8180  FORMAT(1X,F7.1,/1X,F7.2,/,1X,F7.4,/(1X,F7.4,8F8.4))
      WRITE(JTAPE,8181) (R29(J), J= 2,I1)
8181  FORMAT(1X,F7.1,8F8.1)
      IF(I21.NE.1)GO TO 8185
      WRITE(JTAPE,8183)(I33(I), I= 1,I1)
8183  FORMAT(1X,I5,11I6)
      WRITE(JTAPE,8184)(R30(I), I= 1,I1)
8184  FORMAT(1X,F7.4,8F8.4)
8185  CONTINUE
8177  CONTINUE
      IF(JTAPE.EQ.JTAPE5)GO TO 9190
C
C***** SECTION NINE - TRANSFERS
C
      DO 9189 J=1,NLINES
      RREF(J,1)= RREF(J,IZR2)
      ZREF(J,1)= ZREF(J,IZR2)
      DO 9188 N=1,NTHETA
      DPDTH1(J,N)= RADIN(J,N,IZR2)
      TANAIN(J,N,1)= PHIN(J,N,IZR2)
      THETA1(J,N)= PIN(J,N,IZR2)
      VMIN(J,N,1)= VMIN(J,N,IZR2)
9188  CONTINUE
9189  CONTINUE
      IZA1=NIZA1
      IZA2=NIZA2
      LTAPE= NEWL
      JTAPE= NEWJ
      IZR1= 2
      IZR2= IZA2-IZA1+IZR1
      GO TO 1005
9190  CONTINUE
      CALL OTHREE
      RETURN
      END
```

```
      SUBROUTINE OVERAL
      COMMON/ARRAY/RADIN(15,13,7),PIN(15,13,7),TIN(15,13,7),VMIN(15,13,7
     1),TANAIN(15,13,7),PHIN(15,13,7)
      COMMON/SECTOR/IFFULL,EF(18),CARD(12,18),I1,I2,I3,I4,I5,I6,I7,I8,
     1I9,I10,I11,I12,I13,I14,I15,I16,I17,I18,I19,I20,I21,I22,R1,R2,R3,
     2R4,R5,R6,I23(30),I24(30),I25(30),I26(30),I27(30),I28(30),I29(30),
     3I30(30),I31(30),I32(30),R13(30,15),R14(30,15),R15(30,15),
     4R16(30,15),R17(30,15),R18(30,15),R19(30,15),R20(30,15),R21(15),
     5R22(30),R23,R24,R25,R26(30),R27(30),R28(30),R29(30),I33(30),
     6R30(30),EXPLOS(30),XMACH(30)
      COMMON /GEN/TITLE(12),NTHETA,LP,NLOOPS,IFPRIN,R32(18)
      COMMON/OV/IZR1,IZR2,IZA1,GUMBO,LOG1
      DIMENSION DELW(13,15),TINLET(15,13),PINLET(15,13),TZ(30),R(30),EI(
     130),EQ(30)
      EQUIVALENCE (I2,NLINES),(TINLET(1),TIN(1))
      SWITCH(W,X,Y,Z)=W+X*Y**2*(1.0+Z**2)/XKP
      XKP=2.0*32.175*778.0*R6
      NTUB=NLINES-1
      TOTALW=0.0
      TZ(1)=0.0
      DO 20 N=1,NTHETA
      DO 10 J=1,NLINES
   10 PINLET(J,N)=PINLET(J,N,1)
      NP1=N+1
      NM1=N-1
      IF(N.EQ.NTHETA)NP1=1
      IF(N.EQ.1)NM1=NTHETA
      WIDTH=R32(NP1)-R32(NM1)
      IF(N.EQ.NTHETA.OR.N.EQ.1)WIDTH=WIDTH+2.0*3.1415926535
      DO 20 J=1,NTUB
      TSWCHJ=SWITCH(TINLET(J,N),-1.0,VMIN(J,N,2),TANAIN(J,N,2))
      TSWCH1=SWITCH(TINLET(J+1,N),-1.0,VMIN(J+1,N,2),TANAIN(J+1,N,2))
      PSWCHJ=PINLET(J,N)*(TSWCHJ/TINLET(J,N))**GUMBO
      PSWCH1=PINLET(J+1,N)*(TSWCH1/TINLET(J+1,N))**GUMBO
      DELW(N,J)=WIDTH/(16.0*R3)*(VMIN(J,N,2)*COS(PHIN(J,N,2))+VMIN(J+1,N
     1,2)*COS(PHIN(J+1,N,2)))*(PSWCHJ/TSWCHJ+PSWCH1/TSWCH1)*(RADIN(J+1,N
     2,2)**2-RADIN(J,N,2)**2)
      TOTALW=TOTALW+DELW(N,J)
      TZ(1)=TZ(1)+(TINLET(J,N)+TINLET(J+1,N))/2.0*DELW(N,J)
   20 CONTINUE
      RETURN
      ENTRY OVLTWO
      DO 40 I=IZR1,IZR2
      IZA=I-IZR1+IZA1
      IF(IZA.EQ.1)GO TO 40
      TZ(IZA)=0.0
      R(IZA)=0.0
      DO 30 N=1,NTHETA
      DO 30 J=1,NTUB
      TSWCHJ=SWITCH(TIN(J,N,I),1.0,VMIN(J,N,I),TANAIN(J,N,I))
      TSWCH1 =SWITCH(TIN(J+1,N,I),1.0,VMIN(J+1,N,I),TANAIN(J+1,N,I))
      PSWCHJ=PIN(J,N,I)*(TSWCHJ/TIN(J,N,I))**GUMBO
      PSWCH1 =PIN(J+1,N,I)*(TSWCH1 /TIN(J+1,N,I))**GUMBO
      R(IZA)=DELW(N,J)*(((PSWCHJ+PSWCH1 )/(PINLET(J,N)+PINLET(J+1,N)))**
     1(1.0/GUMBO)-1.0)+R(IZA)
   30 TZ(IZA)=TZ(IZA)+DELW(N,J)*(TSWCHJ+TSWCH1)/2.0
      R(IZA)=(R(IZA)/TOTALW+1.0)**GUMBO
      TZ(IZA)=TZ(IZA)/TZ(1)-1.0
      EI(IZA)=0.0
      EQ(IZA)=0.0
```

```
      IF(TZ(IZA).NE.0.0.AND.R(IZA).GT.0.0)EI(IZA)=(R(IZA)**(1.0/GUMBO)-1S24*0060
     1.0)/TZ(IZA)                                                       S24*0061
      IF(TZ(IZA).GT.-1.0.AND.R(IZA).GT.0.0)EQ(IZA)=ALOG(R(IZA))/(ALOG(TZS24*0062
     1(IZA)+1.0)*GUMBO)                                                 S24*0063
40    CONTINUE                                                          S24*0064
      RETURN                                                            S24*0065
      ENTRY OTHREE                                                      S24*0066
      DO 50 I=2,I1                                                      S24*0067
      SPEED=R29(I)                                                      S24*0068
      IF(SPEED.NE.0.0)GO TO 60                                          S24*0069
50    CONTINUE                                                          S24*0070
60    WRITE(LOG1,70)LP,TOTALW,SPEED,(I,R(I),TZ(I),EI(I),EQ(I),I=2,I1)   S24*0071
70    FORMAT(1H1,36X,52HOVERALL PERFORMANCE DETERMINED FROM RESULTS OF CS24*0072
     1YCLE,I3,/,37X,55H*************************************************S24*0073
     2******//48X30HWEIGHT FLOW (LBS PER SECOND) =,F7.2,//,45X34HFIRST NS24*0074
     3ON-ZERO BLADE SPEED (RPM) =,F9.1,///,35X,60HSTATION   TOTAL PRESSUS24*0075
     4RE   DELTA T   ISENTROPIC   POLYTROPIC,/,35X,60HNUMBER        RATIS24*0076
     5O          ON T      EFFICIENCY   EFFICIENCY,//,(35X,I3,F17.4,F14.4,S24*0077
     6F11.4,F13.4))                                                     S24*0078
      RETURN                                                            S24*0079
      END                                                               S24*0080
```

```
      FUNCTION SERVUS(THETA,THETA1,NTHETA,N,I)
C THIS FUNCTION COMPUTES THE RELATIVE SECTOR WIDTH RATIO
      DIMENSION THETA(13,7),THETA1(13)
      NP1= N+1
      NM1= N-1
      IF(N .EQ. 1) NM1= NTHETA
      IF(N .EQ. NTHETA) NP1= 1
      SERVUS= DELTA(THETA(NP1,I),THETA(NM1,I))/DELTA(THETA1(NP1),
     1THETA1(NM1))
      RETURN
      END
```

```
      FUNCTION FINDY(X,ARGX,ARGY,IMX)
C THIS FUNCTION INTERPOLATES AND EXTRAPOLATES LINEARLY
      DIMENSION ARGX(IMX),ARGY(IMX)
      ARG= (ARGX(IMX)-ARGX(1))/ABS(ARGX(IMX)-ARGX(1))
      IF(IMX-2) 10,20,20
10    FINDY= ARGY(1)
      RETURN
20    DO 30 I= 2,IMX
      IF((ARGX(I)-X)*ARG .GE. 0.0) GO TO 40
30    CONTINUE
40    FINDY= ARGY(I)+(X-ARGX(I))*(ARGY(I)-ARGY(I-1))/(ARGX(I)-ARGX(I-1))
      RETURN
      END
```

```
FUNCTION DELTA(THETA2,THETA1)
DELTA= THETA2-THETA1
IF(DELTA .LT. 0.0) DELTA= DELTA + 2.0*3.1415926
RETURN
END
```

```
      FUNCTION DERIV(P,THETA,NTHETA,J,N,I)
C THIS FUNCTION COMPUTES PARTIAL DPDTHETA AT J,N,I
      DIMENSION P(15,13,7),THETA(13,7)
      NP1= N+1
      NM1= N-1
      IF(N .EQ. 1) NM1= NTHETA
      IF(N .EQ. NTHETA) NP1= 1
      DERIV= 0.5*((P(J,NP1,I)-P(J,N,I))/DELTA(THETA(NP1,I),THETA(N,I))
     1+ (P(J,N,I)-P(J,NM1,I))/DELTA(THETA(N,I),THETA(NM1,I)))
      RETURN
      END
```

```
      SUBROUTINE FOUR(P,THETA,NTHETA,AZERO,AN,BN,NMAX,I)
C THIS SUBROUTINE FINDS THE 2*NMAX + 1 FOURIER COEFFICIENTS OF P(THETA),
C WHICH IS DEFINED BY NTHETA POINTS AS A PIECEWISE LINEAR FUNCTION
      DIMENSION P(13),THETA(13,7),AN(26),BN(26)
      PI= 3.1415926
      AZERO= 0.0
      DO 10 N= 1,NMAX
      AN(N)= 0.0
10    BN(N)= 0.0
      DO 100 N= 1,NTHETA
      NP1= N+1
      NM1= N-1
      IF(N .EQ. NTHETA) NP1= 1
      IF(N .EQ. 1) NM1= NTHETA
      AZERO= AZERO + P(N)*DELTA(THETA(NP1,I),THETA(NM1,I))
      BANG= (P(NP1) - P(N))/DELTA(THETA(NP1,I),THETA(N,I))/PI
      DO 100 NN= 1,NMAX
      XN= NN
      AN(NN)= AN(NN) + (COS(XN*THETA(NP1,I)) - COS(XN*THETA(N,I)))*BANG/
     1XN/XN
      BN(NN)= BN(NN) + (SIN(XN*THETA(NP1,I)) - SIN(XN*THETA(N,I)))*BANG/
     1XN/XN
100   CONTINUE
      AZERO= 0.25*AZERO/PI
      RETURN
      END
```

```
      FUNCTION XMEAN(P,THETA,NTHETA,J,I)                              S30*0000
C THIS FUNCTION COMPUTES THE THETA-MEAN OF P                          S30*0001
      DIMENSION P(15,13,7),THETA(13,7)                                S30*0002
      PI= 3.1415926                                                   S30*0003
      XMEAN=0.0                                                       S30*0004
      DO 10 N= 1,NTHETA                                               S30*0005
      NP1= N+1                                                        S30*0006
      NM1= N-1                                                        S30*0007
      IF(N .EQ. 1) NM1= NTHETA                                        S30*0008
      IF(N .EQ. NTHETA) NP1= 1                                        S30*0009
10    XMEAN= XMEAN + P(J,N,I)*DELTA(THETA(NP1,I),THETA(NM1,I))        S30*0010
      XMEAN= 0.25*XMEAN/PI                                            S30*0011
      RETURN                                                          S30*0012
      END                                                             S30*0013
```

```
      SUBROUTINE EDWIN(Y,Z,P,DP,NT)
        DIMENSION Y(1),P(1),DP(1)
      COMMON /ONE/ NF,CNFZ(36),DNFZ(36),XL1,CNF(36),DNF(36),XL2,
     1   IND2,ENF(36),FNF(36),XL3,IND3,WZ,VZ,WCAP1,VCAP1,XK1,
     2   WCAP2,VCAP2,XK2,SZ,EP,UMQ,AMQ(36),BMQ(36),CMQ(36),DMQ(36,2),
     3   EMQ(36,2),FMQ(36,2)
      COMPLEX AMQ,BMQ,CMQ,DMQ,EMQ,FMQ,UMQ
      COMPLEX DPMQ,PMQ,DETQ
      DO 10 I=1,NT
      P(I)=0.0
      DP(I)=0.0
   10 CONTINUE
      DO 1000 M=1,NF
      XM=M
      A=EXP(XM*Z)
      DETQ=CEXP(-UMQ*Z*SZ*XM)
      WBAR=WBARF(Z)
      VBAR=VBARF(Z)
      VBARD=VBARDF(Z)
      DPMQ=XM*(CMPLX(XM*WBAR,XM*VBAR-VBARD)*AMQ(M)*A
     1     +CMPLX(XM*SZ*(VBAR-SZ*WBAR),-VBARD)*BMQ(M)*DETQ
     2     +CMPLX(XM*WBAR,-XM*VBAR-VBARD)*CMQ(M)/A)
      XK=XK1
      DO 200 I=1,2
      B=EXP(XK*Z)
      DPMQ=DPMQ+CMPLX(WBAR*(XK+XM)**2,XM*((XK+XM)*VBAR-VBARD))*
     1    DMQ(M,I)*B*A + (CMPLX(XK,-XM*SZ)*CMPLX(XK*WBAR,XM*(VBAR-
     2    WBAR*SZ)) - UMQ*XM*VBARD) * EMQ(M,I)*B*DETQ+
     3    CMPLX(WBAR*(XK-XM)**2,XM*((XK-XM)*VBAR-VBARD))*FMQ(M,I)*B/A
      XK=XK2
  200 CONTINUE
      PMQ=UMQ*DPMQ/XM
      DPMQ=-DPMQ
      DO 300 I=1,NT
      A=COS(XM*Y(I))
      B=SIN(XM*Y(I))
      P(I)=P(I)+2.*(REAL(PMQ)*A-AIMAG(PMQ)*B)
      DP(I)=DP(I)+2.0*(REAL(DPMQ)*A-AIMAG(DPMQ)*B)
  300 CONTINUE
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NEWRAP(W1,W2,W3,W4,Z1,Z2,Z3,Z4,IND)
      COMMON/ONE/NF,CNZ(36),DNZ(36),XL1,CNF(36),DNF(36),XL2,IND2,ENF(36)
     1,FNF(36),XL3,IND3,WZ,VZ,WCAP1,VCAP1,XK1,WCAP2,VCAP2,XK2,SS,EP,UMQ(
     22),AMQ(2,36),BMQ(2,36),CMQ(2,36),DMQ(2,36,2),EMQ(2,36,2),FMQ(2,36,
     32)
      FUN(Z,W)=WCAP1*EXP(XK1*Z)+WCAP2*EXP(XK2*Z)-W
      DFUN(Z,XK,WC)=Z*WC*EXP(XK*Z)
      IF (IND .GT. 0)  GO TO 500
   20 ASSIGN 200 TO IEXIT
      XK1 = .35
      XK2 = -.35
      ASSIGN 1000 TO IEXIT
      J=1
      WZ=(W1+W4)/2.0
  100 A=EXP(XK1*Z1)
      B=EXP(XK2*Z1)
      C=EXP(XK1*Z4)
      D=EXP(XK2*Z4)
      IF(WZ .EQ. W4)  GO TO 400
      E=W1-WZ
      F=W4-WZ
      WCAP1=(E*D-B*F)/(A*D-B*C)
      WCAP2=(A*F-C*E)/(A*D-B*C)
      GO TO IEXIT, (200,1000)
  200 XK1P=XK1
      XK2P=XK2
      I=1
  300 A=DFUN(Z2,XK1,WCAP1)
      B=DFUN(Z2,XK2,WCAP2)
      C=DFUN(Z3,XK1,WCAP1)
      D=DFUN(Z3,XK2,WCAP2)
      E=-FUN(Z2,W2-WZ)
      F=-FUN(Z3,W3-WZ)
      DK1=(E*D-B*F)/(A*D-B*C)
      DK2=(A*F-C*E)/(A*D-B*C)
      XK1=XK1+DK1
      XK2=XK2+DK2
      I=I+1
      IF (I.LE. 5 .AND. (ABS(E-WZ) .GT. 1.0E-4 .OR.
     1    ABS(F-WZ) .GT. 1.0E-4)) GO TO 300
      J=J+1
      IF ((ABS(XK2P-XK2) .LT. 1.0E-4 .AND. ABS(XK1P-XK1) .LT.
     1   1.0E-4) .OR. J .GT. 5)   ASSIGN 1000 TO IEXIT
      GO TO 100
  400     WCAP1=0.0
      WCAP2=0.0
      GO TO 1000
  500 VZ=(W1+W4)/2.0
      E=W1-VZ
      F=W4-VZ
      VCAP1=(E*D-B*F)/(A*D-B*C)
      VCAP2=(A*F-C*E)/(A*D-B*C)
 1000  RETURN
      END
```

```
      SUBROUTINE CPOUR(R,L)
      COMPLEX R
C             CONTROLS 0/UFLOW FOR R BY EXTRACTING POWERS OF 10
C                    R - NUMBER
C                    L - RESULTANT POWER OF 10
      ASSIGN 20 TO IB
   10 S=CABS(R)
      GO TO IB, (20,40)
   20 IF (S .LT. 1.0E+9) GO TO 30
      R=R*1.0E-10
      L=L+10
      GO TO 10
   30 ASSIGN 40 TO IB
   40 IF (S .GT. 1.0E-9) GO TO 50
      R=R*1.0E+10
      L=L-10
      GO TO 10
   50 RETURN
      END
```

```
      FUNCTION VBARF(Z)                                                 S34*0000
      COMMON /ONE/ NF,CNFZ(36),DNFZ(36),XL1,CNF(36),DNF(36),XL2,        S34*0001
     1   IND2,ENF(36),FNF(36),XL3,IND3,WZ,VZ,WCAP1,VCAP1,XK1,           S34*0002
     2   WCAP2,VCAP2,XK2,SZ,EP,UMQ,AMQ(36),BMQ(36),CMQ(36),DMQ(36,2),   S34*0003
     3   EMQ(36,2),FMQ(36,2)                                            S34*0004
      COMPLEX AMQ,BMQ,CMQ,DMQ,EMQ,FMQ,UMQ                               S34*0005
      VBARF=VZ+VCAP1*EXP(XK1*Z)+VCAP2*EXP(XK2*Z)                        S34*0006
      RETURN                                                            S34*0007
      END                                                               S34*0008
```

```
      FUNCTION VBARDF(Z)
      COMMON /ONE/ NF,CNFZ(36),DNFZ(36),XL1,CNF(36),DNF(36),XL2,
     1   IND2,ENF(36),FNF(36),XL3,IND3,WZ,VZ,WCAP1,VCAP1,XK1,
     2   WCAP2,VCAP2,XK2,SZ,EP,UMQ,AMQ(36),BMQ(36),CMQ(36),DMQ(36,2),
     3   EMQ(36,2),FMQ(36,2)
      COMPLEX AMQ,BMQ,CMQ,DMQ,EMQ,FMQ,UMQ
      VBARDF=XK1*VCAP1*EXP(XK1* Z)+XK2*VCAP2*EXP(XK2*Z)
      RETURN
      END
```

```
      FUNCTION WBARF(Z)
      COMMON /ONE/ NF,CNFZ(36),DNFZ(36),XL1,CNF(36),DNF(36),XL2,
     1   IND2,ENF(36),FNF(36),XL3,IND3,WZ,VZ,WCAP1,VCAP1,XK1,
     2   WCAP2,VCAP2,XK2,SZ,EP,UMQ,AMQ(36),BMQ(36),CMQ(36),DMQ(36,2),
     3   EMQ(36,2),FMQ(36,2)
      COMPLEX AMQ,BMQ,CMQ,DMQ,EMQ,FMQ,UMQ
      WBARF=WZ+WCAP1*EXP(XK1*Z)+WCAP2*EXP(XK2*Z)
       RETURN
       END
```

```
      SUBROUTINE CSIMEQ(A,X,DET,NR,NS,NDEX,LSGN,LEXP,NRD,NCD,NSD)
      COMPLEX A,X,DET,R
      DIMENSION A(NRD,NCD),X(NRD,NSD)
C                 A - MATRIX AUGMENTED BY CONSTANT VECTORS
C                 X - SOLUTION MATRIX
C                 DET - DETERMINANT, NORMALIZED
C                 NR - NO OF ROWS
C                 NS - NO OF SOLUTIONS
C                 NDEX - -1 -DET ONLY
C                         0  -SOLUTION ONLY
C                        +1 -BOTH DET AND SOLUTION
C                 LSGN - AS OUTPUT = 0 UNLESS SINGULAR
C                        OTHERWISE = ROW OF SINGULARITY
C                 LEXP - EXPONENT INDICATOR FOR DET
C                        DETERMINANT=DET*10.**LEXP
C            1000 SERIES SETS UP SOLUTION TYPE
 1000 IF (NDEX) 1800, 1500, 1200
 1200 ASSIGN 4100 TO MBIPAS
      ASSIGN 3100 TO NBIPAS
      GO TO 1900
 1500 ASSIGN 5000 TO MBIPAS
      ASSIGN 3100 TO NBIPAS
      GO TO 2000
 1800 ASSIGN 4100 TO MBIPAS
      ASSIGN 4000 TO NBIPAS
 1900 LSGN=1
      LEXP=0
      DET=(1.0,0.0)
C            2000 SERIES TRIANGULARIZES MATRIX
 2000 NC=NR+NS
      I=1
C            2100 SERIES MAXIMIZES PIVOTAL ELEMENT
 2100 S=CABS(A(I,I))
      J=I
      IF (I-NR) 2110,2150,3000
 2110 K=I+1
 2120 T=CABS(A(K,I))
      IF (T .LE. S) GO TO 2130
      S=T
      J=K
 2130 K=K+1
      IF (K .LE. NR) GO TO 2120
 2150 IF (S .EQ. 0.0) GO TO 9000
C            2200 SERIES INTERCHANGES ROWS IF NECESSARY
 2200 IF (J .LE. I) GO TO 2500
      LSGN=-LSGN
      K=I
 2250 R=A(I,K)
      A(I,K)=A(J,K)
      A(J,K)=R
      K=K+1
      IF (K .LE. NC) GO TO 2250
C            2500 SERIES THEN REDUCES WITH ZERO CHECK
 2500 J=I+1
 2510 IF (J .LE. NC) GO TO 2520
      I=I+1
      GO TO 2100
 2520 IF (REAL(A(I,J)) .EQ. 0.0 .AND. AIMAG(A(I,J)) .EQ. 0.0 )
     1 GO TO 2590
      A(I,J)=A(I,J)/A(I,I)
```

```
      K=I+1
 2530 IF (K .GT. NR) GO TO 2590
      A(K,J)=A(K,J)-A(I,J)*A(K,I)
      K=K+1
      GO TO 2530
 2590 J=J+1
      GO TO 2510
C         3000 SERIES COMPUTES SOLUTION
 3000 GO TO NBIPAS, (3100,4000)
 3100 K=1
 3200 L=NR+K
      X(NR,K)=A(NR,L)
      I=NR-1
 3300 J=I+1
      R=(0.0,0.0)
 3400 R=R+A(I,J)*X(J,K)
      J=J+1
      IF (J .LE. NR) GO TO 3400
      X(I,K)=A(I,L)-R
      I=I-1
      IF (I .GT. 0) GO TO 3300
      K=K+1
      IF (K .LE. NS) GO TO 3200
C         4000 SERIES COMPUTES DETERMINANT
 4000 GO TO MBIPAS, (4100,5000)
 4100 I=1
 4200 R=A(I,I)
      CALL CPOUR(R,LEXP)
      DET=DET*R
      CALL CPOUR(DET,LEXP)
      I=I+1
      IF (I .LE. NR) GO TO 4200
      S=LSGN
      DET=S*DET
 5000 LSGN=0
      RETURN
 9000 DET=(0.0,0.0)
      LSGN=I
      RETURN
      END
```

```
      SUBROUTINE GEORGE(Y,Z,U,V,NT)
      DIMENSION U(1),V(1),Y(1)
      COMMON /ONE/ NF,CNFZ(36),DNFZ(36),XL1,CNF(36),DNF(36),XL2,
     1   IND2,ENF(36),FNF(36),XL3,IND3,WZ,VZ,WCAP1,VCAP1,XK1,
     2   WCAP2,VCAP2,XK2,SZ,EP,UMQ,AMQ(36),BMQ(36),CMQ(36),DMQ(36,2),
     3   EMQ(36,2),FMQ(36,2)
      COMPLEX AMQ,BMQ,CMQ,DMQ,EMQ,FMQ,UMQ
      COMPLEX ALPHMQ,BETAMQ,DETQ
      DO 10 I=1,NT
       U(I)=0.0
      V(I)=0.0
   10 CONTINUE
      DO 1000 M=1,NF
      XM=M
      A=EXP(XM*Z)
      DETQ=CEXP(-UMQ*Z*SZ*XM)
      ALPHMQ=AMQ(M)*A+BMQ(M)*DETQ+CMQ(M)/A
      BETAMQ=XM*(AMQ(M)*A-UMQ*SZ*BMQ(M)*DETQ-CMQ(M)/A)
      XK=XK1
      DO 200 I=1,2
      B=EXP(XK*Z)
      ALPHMQ=ALPHMQ+B*(A*DMQ(M,I)+DETQ*EMQ(M,I)+FMQ(M,I)/A)
      BETAMQ=BETAMQ+B*((XK+XM)*A*DMQ(M,I)+CMPLX(XK,-XM*SZ)*DETQ*EMQ(M,I)
     1     +(XK-XM)*FMQ(M,I)/A)
      XK=XK2
  200 CONTINUE
      ALPHMQ=ALPHMQ*(-XM*UMQ)
      DO 300 I=1,NT
      A=COS(XM*Y(I))
      B=SIN(XM*Y(I))
      U(I)=U(I)+2.0*(REAL(ALPHMQ)*A-AIMAG(ALPHMQ)*B)
      V(I)=V(I)+2.*(REAL(BETAMQ)*A-AIMAG(BETAMQ)*B)
  300 CONTINUE
 1000 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SMITH
      COMMON /ONE/ NF,CNFZ(36),DNFZ(36),XL1,CNF(36),DNF(36),XL2,
     1   IND2,ENF(36),FNF(36),XL3,IND3,WZ,VZ,WCAP1,VCAP1,XK1,
     2   WCAP2,VCAP2,XK2,SZ,EP,UMQ,AMQ(36),BMQ(36),CMQ(36),DMQ(36,2),
     3   EMQ(36,2),FMQ(36,2)
      COMPLEX AMQ,BMQ,CMQ,DMQ,EMQ,FMQ,UMQ
      COMPLEX HMQ,PMQ,TMQ,COEMQ,BOEMQ,SOLMQ,AQQ,BQQ,DETQ
      DIMENSION COEMQ(3,4),SOLMQ(3),BOEMQ(3,4)
      SZ=VZ/WZ
      EP=1.0/WZ
      UMQ=CMPLX(0.0,1.0)
      DO 1000 M=1,NF
      HMQ=CMPLX(DNFZ(M),-CNFZ(M))/2.0
      PMQ=CMPLX(DNF(M),-CNF(M))/2.0
      TMQ=CMPLX(FNF(M),-ENF(M))/2.0
      XM=M
      AQQ= -XM*UMQ
      XZ=XL1
      COEMQ(1,1)=AQQ*VBARDF(XZ)
      A=EXP(XM*XL1)
      COEMQ(1,3)=COEMQ(1,1)/A
      COEMQ(1,2)=(COEMQ(1,1)-CMPLX(XM**2*(1.0+SZ**2)*WBARF(XZ),
     1   0.0))*CEXP(AQQ*SZ*XL1)
      COEMQ(1,1)=COEMQ(1,1)*A
      COEMQ(1,4)=AQQ*HMQ
      XZ=XL2
      BQQ=PMQ
      IND=IND2
      DO 200 I=2,3
      WBAR=WBARF(XZ)
      VBAR=VBARF(XZ)
      VBARD=VBARDF(XZ)
      A=EXP(XM*XZ)
      IF (IND .GT. 0) GO TO 110
      COEMQ(I,4)=AQQ*BQQ
      COEMQ(I,1)=XM*A*CMPLX(XM*WBAR,XM*VBAR-VBARD)
      COEMQ(I,2)=XM*CEXP(AQQ*SZ*XZ)*CMPLX(XM*SZ*(VBAR-SZ*WBAR),-VBARD)
      COEMQ(I,3)=(XM/A)*CMPLX(XM*WBAR,-XM*VBAR-VBARD)
      GO TO 120
  110 COEMQ(I,4)= BQQ
      COEMQ(I,1)=XM*A*CMPLX(WBAR,VBAR)
      COEMQ(I,2)=AQQ*CEXP(AQQ*SZ*XZ)*(SZ*WBAR-VBAR)
      COEMQ(I,3)=-XM*CMPLX(WBAR,-VBAR)/A
  120 XZ=XL3
      BQQ=TMQ
      IND=IND3
  200 CONTINUE
       DO 250 I=1,3
      DO 250 J=1,4
      BOEMQ(I,J)=COEMQ(I,J)
  250 CONTINUE
      CALL CSIMEQ(BOEMQ,SOLMQ,DETQ,3,1,0,LSGN,LEXP,3,4,1)
      AMQ(M)=SOLMQ(1)
      BMQ(M)=SOLMQ(2)
      CMQ(M)=SOLMQ(3)
      XK=XK1
      WC=WCAP1
      VC=VCAP1
      DO 300 I=1,2
      DMQ(M,I)=-AQQ*XK**2*VC
```

```
      EMQ(M,I)=XM*(UMQ*XK**2*VC-XM*(1.0+SZ**2)*
     1        CMPLX(-XK*WC,  XM*(SZ*WC-VC)))*BMQ(M)
      FMQ(M,I)=DMQ(M,I)*CMQ(M)
      DMQ(M,I)=DMQ(M,I)*AMQ(M)
      DMQ(M,I)=DMQ(M,I)/(XK*(XK+2.0*XM)*CMPLX(XK+XM,  XM*SZ))
      EMQ(M,I)=EMQ(M,I)/(XK*CMPLX(XK**2-XM**2*(1.0+SZ**2),
     1   -2.0*XK*XM*SZ))
      FMQ(M,I)=FMQ(M,I)/(XK*(XK-2.0*XM)*CMPLX(XK-XM,XM*SZ))
      XK=XK2
      WC=WCAP2
      VC=VCAP2
  300 CONTINUE
      DO 320 I=1,3
      COEMQ(I,4)=0.0
  320 CONTINUE
      XK=XK1
      XZ=XL1
      WBAR=WBARF(XZ)
      VBARD=VBARDF(XZ)
      DO 350 I=1,2
      COEMQ(1,4)=COEMQ(1,4)-DMQ(M,I)*CMPLX(XK*WBAR*(XK+2.0*XM),
     1    -XM*VBARD)*EXP((XK+XM)*XZ)-EMQ(M,I)*(CMPLX(XK,-XM*SZ)**2
     2     *WBAR - CMPLX(XM*WBAR,VBARD)*XM)*CEXP(CMPLX(XK,-XM*SZ)*XZ)
     3     -FMQ(M,I)*CMPLX(XK*WBAR*(XK-2.0*XM),-XM*VBARD)*EXP((XK-XM)*XZ)
      XK=XK2
  350 CONTINUE
      XZ=XL2
      IND=IND2
      DO 400 J=2,3
      WBAR=WBARF(XZ)
      VBAR=VBARF(XZ)
      VBARD=VBARDF(XZ)
      XK=XK1
      DO 380 I=1,2
      IF (IND .GT. 0) GO TO 360
      COEMQ(J,4)=COEMQ(J,4)-DMQ(M,I)*CMPLX(WBAR*(XK+XM)**2,XM*
     1       (VBAR*(XK+XM)-VBARD))*EXP((XK+XM)*XZ)-EMQ(M,I)*((CMPLX
     2       (XK,-XM*SZ))*CMPLX(XK*WBAR,XM*(VBAR-WBAR*SZ))-AQQ*VBARD)*
     3   CEXP(XZ*CMPLX(XK,-XM*SZ))-FMQ(M,I)*CMPLX(WBAR*(XK-XM)**2,XM*
     4      (VBAR*(XK-XM)-VBARD))*EXP((XK-XM)*XZ)
      GO TO 370
  360 COEMQ(J,4)=COEMQ(J,4)-DMQ(M,I)*CMPLX((XK+XM)*WBAR,XM*VBAR)*
     1    EXP((XK+XM)*XZ)-EMQ(M,I)*CMPLX(WBAR*XK,XM*(VBAR-WBAR*SZ))*
     2      CEXP(CMPLX(XK,-XM*SZ)*XZ)-FMQ(M,I)*CMPLX((XK-XM)*WBAR,
     3     XM*VBAR)*EXP((XK-XM)*XZ)
  370 XK=XK2
  380 CONTINUE
      XZ=XL3
      IND=IND3
  400 CONTINUE
      CALL CSIMEQ(COEMQ,SOLMQ,DETQ,3,1,0,LSGN,LEXP,3,4,1)
      AMQ(M)=AMQ(M)+EP*SOLMQ(1)
      BMQ(M)=BMQ(M)+EP*SOLMQ(2)
      CMQ(M)=CMQ(M)+EP*SOLMQ(3)
      DO 500 I=1,2
      DMQ(M,I)=DMQ(M,I)*EP
      EMQ(M,I)=EMQ(M,I)*EP
      FMQ(M,I)=FMQ(M,I)*EP
  500 CONTINUE
 1000 CONTINUE
```

```
      RETURN                                                    S39*0120
      END                                                       S39*0121
```

```
      SUBROUTINE PCTURE(NTAPE,JTAPE1,JTAPE2,JTAPE3,JTAPE4,JTAPE5,LTAPE1,
     1LTAPE2,LTAPE3,LTAPE4,LTAPE5,J1,J21,J22,J31,J32,J41,J42,J5,K1,NTHET
     2A,TITLE,IFPRIN,R32)
      COMMON/HUGE/P(15,18,8),T(15,18,8),VM(15,18,8),TANALP(15,18,8),RAD(
     115,8)
      DIMENSION P1(1),T1(1),VM1(1),R1(1),TA1(1),R32(1),TITLE(1)
      EQUIVALENCE (P(1),P1(1)),(T(1),T1(1)),(VM(1),VM1(1)),(TANALP(1),TA
     11(1)),(RAD(1),R1(1))
      CALL PLOT(JTAPE1,JTAPE2,JTAPE3,JTAPE4,JTAPE5,NTAPE,K1,NTHETA,TITLE
     1,J1,J21,J22,J31,J32,J41,J42,J5,CP)
      IF(IFPRIN.NE.3.AND.IFPRIN.NE.4.AND.IFPRIN.NE.6.AND.IFPRIN.NE.7)GO
     1TO 1000
      C1=180.0/3.1415926535
      C2=2.0*32.175*778.0*CP
      JTAPE=LTAPE1
      REWIND JTAPE
      II=J1
      III=1
      READ(JTAPE,50)(DUM,N=1,NTHETA)
50    FORMAT(1X,/,1X,/,1X,/,1X,/,(9F8.4))
210   CONTINUE
      DO 202 N=1,NTHETA
      IF(JTAPE.NE.LTAPE1)GO TO 207
      READ(JTAPE,100)IFFULL
100   FORMAT(6X,I6,/,1X)
      IF(IFFULL.EQ.1)GO TO 200
      READ(JTAPE,110)I1,I2,I3,I21,GASK
110   FORMAT(6X,3I6,/,1X,/,66X,I6,/,1X,/,28X,E13.6)
      DO 150 I=2,I1
      READ(JTAPE,115)I23,IFMACH,IFLVSI
115   FORMAT(6X,I6,12X,I6,18X,I6)
      IF(I23.EQ.6)GO TO 130
      IF((I23.EQ.2.OR.I23.EQ.4).AND.IFLVSI.EQ.1)READ(JTAPE,116)
      IF((I23.EQ.2.OR.I23.EQ.4).AND.IFMACH.EQ.3)READ(JTAPE,116)
116   FORMAT(1X)
      READ(JTAPE,160)((DUM,K=1,8),J=1,I3)
130   CONTINUE
150   CONTINUE
      READ(JTAPE,160)(DUM,I=1,I2),(DUM,I=1,I1),DUM,DUM,DUM
160   FORMAT(2X,6E13.6)
      READ(JTAPE,160)(DUM,DUM,DUM,I=1,I1)
      READ(JTAPE,160)(DUM,I=2,I1)
      IF(I21.EQ.0)GO TO 200
      READ(JTAPE,165)(IDUM,I=1,I1)
165   FORMAT(6X,12I6)
      READ(JTAPE,160)(DUM,I=1,I1)
200   CONTINUE
207   READ(JTAPE,160)((RAD(J,I),P(J,N,I),T(J,N,I),VM(J,N,I),TANALP(J,N,I
     1),DUM,J=1,I2),I=1,II)
202   CONTINUE
      N1=1
      N2=1
      N3=16
      N4=31
      N5=46
      N6=61
      N7=76
      N8=91
      N9=106
      N10=121
```

```
      N11=136
      N12=151
      N13=166
      N14=181
      N15=196
      N16=211
      N17=226
      N18=241
      N19=256
      C3=778.0/GASK*CP
      DO 900 I=1,II
      WRITE(NTAPE,220)III
220   FORMAT(1H1,40X,52HMERIDIONAL VELOCITY VARIATION WITH RADIUS AT STA
     1TION,I3,/)
      CALL            MPLOT(I2,NTHETA,NTAPE,R1(N1),VM1(N2),VM1(N3),VM1(N
     14),VM1(N5),VM1(N6),VM1(N7),VM1(N8),VM1(N9),VM1(N10),VM1(N11),VM1(N
     212),VM1(N13),VM1(N14),VM1(N15),VM1(N16),VM1(N17),VM1(N18),VM1(N19)
     3)
      IF(III.NE.1)WRITE(NTAPE,225)III
      IF(III.EQ.1)WRITE(NTAPE,230)III
225   FORMAT(1H1,40X,48HSTATIC PRESSURE VARIATION WITH RADIUS AT STATION
     1,I3,/)
      CALL            MPLOT(I2,NTHETA,NTAPE,R1(N1),P1(N2),P1(N3),P1(N4),
     1P1(N5),P1(N6),P1(N7),P1(N8),P1(N9),P1(N10),P1(N11),P1(N12),P1(N13)
     2,P1(N14),P1(N15),P1(N16),P1(N17),P1(N18),P1(N19))
      IF(III.EQ.1)WRITE(NTAPE,225)III
      IF(III.NE.1)WRITE(NTAPE,230)III
230   FORMAT(1H1,40X,47HTOTAL PRESSURE VARIATION WITH RADIUS AT STATION,
     1I3,/)
      C4=1.0
      IF(III.EQ.1)C4=-1.0
      DO 240 J=1,I2
      DO 240 N=1,NTHETA
      TT=T(J,N,I)+VM(J,N,I)**2*(1.0+TANALP(J,N,I)**2)/C2*C4
      P(J,N,I)=P(J,N,I)*(TT/T(J,N,I))**C3
      IF(III.NE.1)T(J,N,I)=TT
240   TANALP(J,N,I)=ATAN(TANALP(J,N,I))*C1
      CALL            MPLOT(I2,NTHETA,NTAPE,R1(N1),P1(N2),P1(N3),P1(N4),
     1P1(N5),P1(N6),P1(N7),P1(N8),P1(N9),P1(N10),P1(N11),P1(N12),P1(N13)
     2,P1(N14),P1(N15),P1(N16),P1(N17),P1(N18),P1(N19))
      WRITE(NTAPE,245)III
245   FORMAT(1H1,40X,50HTOTAL TEMPERATURE VARIATION WITH RADIUS AT STATI
     1ON,I3,/)
      CALL            MPLOT(I2,NTHETA,NTAPE,R1(N1),T1(N2),T1(N3),T1(N4),
     1T1(N5),T1(N6),T1(N7),T1(N8),T1(N9),T1(N10),T1(N11),T1(N12),T1(N13)
     2,T1(N14),T1(N15),T1(N16),T1(N17),T1(N18),T1(N19))
      WRITE(NTAPE,250)III
250   FORMAT(1H1,40X,44HWHIRL ANGLE VARIATION WITH RADIUS AT STATION ,I3
     1,/)
      CALL            MPLOT(I2,NTHETA,NTAPE,R1(N1),TA1(N2),TA1(N3),TA1(N
     14),TA1(N5),TA1(N6),TA1(N7),TA1(N8),TA1(N9),TA1(N10),TA1(N11),TA1(N
     212),TA1(N13),TA1(N14),TA1(N15),TA1(N16),TA1(N17),TA1(N18),TA1(N19)
     3)
      N1=N1+15
      N 2=N 2+270
      N 3=N 3+270
      N 4=N 4+270
      N 5=N 5+270
      N 6=N 6+270
      N 7=N 7+270
```

```
      N 8=N 8+270
      N 9=N 9+270
      N10=N10+270
      N11=N11+270
      N12=N12+270
      N13=N13+270
      N14=N14+270
      N15=N15+270
      N16=N16+270
      N17=N17+270
      N18=N18+270
      N19=N19+270
      III=III+1
900   CONTINUE
      IF(JTAPE.EQ.LTAPE5)GO TO 1000
      IF(JTAPE.EQ.LTAPE1)GO TO 920
      IF(JTAPE.EQ.LTAPE2)GO TO 930
      IF(JTAPE.EQ.LTAPE3)GO TO 940
910   JTAPE=LTAPE5
      REWIND JTAPE
      II=I1-J5+1
      GO TO 210
920   IF(J21.EQ.0)GO TO 910
      JTAPE=LTAPE2
      REWIND JTAPE
      II=J22-J21+1
      GO TO 210
930   IF(J31.EQ.0)GO TO 910
      JTAPE=LTAPE3
      REWIND JTAPE
      II=J32-J31+1
      GO TO 210
940   IF(J41.EQ.0)GO TO 910
      JTAPE=LTAPE4
      REWIND JTAPE
      II=J42-J41+1
      GO TO 210
1000  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PLOT(JTAPE1,JTAPE2,JTAPE3,JTAPE4,JTAPE5,LOG2,LOG4,NTHET
     1A,TITLE,J1,J21,J22,J31,J32,J41,J42,J5,CP)
      DIMENSION R(15),Y(18,15),SYMBOL(18),LINE(95),XI(1),DASH(1),CROSS(1
     1),BLANK(1),TITLE(12)
      COMMON/HUGE/P(15,18,8),T(15,18,8),VM(15,18,8),TANALP(15,18,8),RAD(
     115,8)
C P IS THETA HERE
      REAL LINE
      DATA(SYMBOL(I),I=1,18)/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HA,1HB
     1,1HC,1HD,1HE,1HF,1HG,1HH,1HJ/,DASH/1H-/,CROSS/1H+/,BLANK/1H /,XI/1
     2HI/
C
C LOG2       STANDARD OUTPUT
C LOG3       DISTORT INPUT TAPES  (J-SERIES)
C LOG4       SCRATCH TAPE IN THETA FOR ANGLES
C
      REWIND LOG4
      WRITE(LOG2,10)
10    FORMAT(1H1)
      LOG3=JTAPE1
      REWIND LOG3
C
C ASSUME A CARD FOR KSUMRY
C
      READ(LOG3,70)NX,NLINES,NBETA,CP,(R(J),J=1,NLINES)
70    FORMAT(1X,/,3I6,/,1X,/,1X,/,1X,/,24X,F8.4,/,(F8.4))
      WRITE(LOG2,20)TITLE,NX,NLINES,NTHETA
20    FORMAT(60X,12HPROGRAM PLOT,/,60X,12H************,//,20X,7HTITLE =,
     13X,12A6,/,20X,35HNUMBER OF STATIONS (ONE PER PAGE) =,I3,/,20X,23HN
     2UMBER OF STREAMLINES =,I3,/,20X,19HNUMBER OF SECTORS =,I3)
      IEND=J1
      III=1
      I1=1
      READ(LOG3,133)
      DO 1000 I=1,NX
      IF(I.EQ.1)GO TO 190
      IF(I.NE.J21.AND.I.NE.J31.AND.I.NE.J41.AND.I.NE.J5)GO TO 150
      III=1
      IF(LOG3.EQ.JTAPE1)GO TO 105
      IF(LOG3.EQ.JTAPE2)GO TO 110
      IF(LOG3.EQ.JTAPE3)GO TO 115
      IF(LOG3.EQ.JTAPE4)GO TO 120
105   IF(J21.EQ.0)GO TO 120
      LOG3=JTAPE2
      REWIND LOG3
      IEND=J22-J21+1
      GO TO 130
110   IF(J31.EQ.0)GO TO 120
      LOG3=JTAPE3
      REWIND LOG3
      IEND=J32-J31+1
      GO TO 130
115   IF(J41.EQ.0)GO TO 120
      LOG3=JTAPE4
      REWIND LOG3
      IEND=J42-J41+1
      GO TO 130
120   LOG3=JTAPE5
      REWIND LOG3
      IEND=NX-J5+1
```

```
130   READ(LOG3,133)
133   FORMAT(2X)
190   CONTINUE
      DO 142 J=1,NLINES
      READ(LOG4)((P(J,N,II),N=1,NTHETA),II=I1,IEND)
142   CONTINUE
      IF(I.EQ.1)GO TO 200
150   CONTINUE
      READ(LOG3,160)IBETA2,IFMACH,IFLVSI
160   FORMAT(I6,12X,I6,18X,I6)
      IF(IBETA2.EQ.6)GO TO 180
      IF(IFLVSI.EQ.1)READ(LOG3,165)
165   FORMAT(1X)
      IF(IFMACH.EQ.3)READ(LOG3,165)
      DO 170 J=1,NBETA
170   READ(LOG3,165)
180   READ(LOG3,185)(R(J),J=1,NLINES)
185   FORMAT(8X,F8.4)
200   DO 220 N=1,NTHETA
      DO 220 J=1,NLINES
220   Y(N,J)=R(J)*COS(P(J,N,III))
      YINC=R(NLINES)/28.0
      YSTART=R(NLINES)+YINC/2.0
      SCALE=YINC/0.1667
      WRITE(LOG2,250)I,SCALE
250   FORMAT(1H1,42X,7HSTATION,I3,30H SCALE OF PLOT = 1 INCH EQUALS,F6.3
     1,5H FEET,/,43X,51H************************************************
     2***,/)
      DO 500 KLINE=1,57
      IF(KLINE.EQ.29)GO TO 280
      DO 210 L=1,95
210   LINE(L)=BLANK
      LINE(48)=XI
      GO TO 290
280   DO 285 L=1,95
285   LINE(L)=DASH
      LINE(48)=CROSS
290   DO 460 J=1,NLINES
      DO 460 N=1,NTHETA
      IF(Y(N,J).GT.YSTART.OR.Y(N,J).LE.YSTART-YINC)GO TO 460
      L=(R(NLINES)+R(J)*SIN(P(J,N,III)))/(2.0*R(NLINES))*94.0+1.5
      LINE(L)=SYMBOL(N)
460   CONTINUE
      YSTART=YSTART-YINC
      WRITE(LOG2,465)LINE
465   FORMAT(20X,95A1)
500   CONTINUE
      III=III+1
1000  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MPLOT(IX,IY,LOG1,X,Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10,Y11,YS42*0000
     112,Y13,Y14,Y15,Y16,Y17,Y18)                                       S42*0001
      DIMENSION X(1),Y1(1),Y2(1),Y3(1),Y4(1),Y5(1),Y6(1),Y7(1),Y8(1),Y9(S42*0002
     11),Y10(1),Y11(1),Y12(1),Y13(1),Y14(1),Y15(1),Y16(1),Y17(1),Y18(1),S42*0003
     2SYMBOL(18),LINE(121),XNUM(13),DASH(1),CROSS(1),BLANK(1),XI(1)     S42*0004
      DATA(SYMBOL(I),I=1,18)/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HA,1HBS42*0005
     1,1HC,1HD,1HE,1HF,1HG,1HH,1HJ/,DASH/1H-/,CROSS/1H+/,BLANK/1H /,XI/1S42*0006
     2HI/                                                               S42*0007
      REAL LINE                                                         S42*0008
      YMIN=Y1(1)                                                        S42*0009
      XMIN=X(1)                                                         S42*0010
      YMAX=YMIN                                                         S42*0011
      XMAX=XMIN                                                         S42*0012
      DO 200 I=1,IX                                                     S42*0013
      GO TO(185,180,175,170,165,160,155,150,145,140,135,130,125,120,115,S42*0014
     1110,105,100),IY                                                   S42*0015
  100 IF(Y18(I).LT.YMIN)YMIN=Y18(I)                                     S42*0016
      IF(Y18(I).GT.YMAX)YMAX=Y18(I)                                     S42*0017
  105 IF(Y17(I).LT.YMIN)YMIN=Y17(I)                                     S42*0018
      IF(Y17(I).GT.YMAX)YMAX=Y17(I)                                     S42*0019
  110 IF(Y16(I).LT.YMIN)YMIN=Y16(I)                                     S42*0020
      IF(Y16(I).GT.YMAX)YMAX=Y16(I)                                     S42*0021
  115 IF(Y15(I).LT.YMIN)YMIN=Y15(I)                                     S42*0022
      IF(Y15(I).GT.YMAX)YMAX=Y15(I)                                     S42*0023
  120 IF(Y14(I).LT.YMIN)YMIN=Y14(I)                                     S42*0024
      IF(Y14(I).GT.YMAX)YMAX=Y14(I)                                     S42*0025
  125 IF(Y13(I).LT.YMIN)YMIN=Y13(I)                                     S42*0026
      IF(Y13(I).GT.YMAX)YMAX=Y13(I)                                     S42*0027
  130 IF(Y12(I).LT.YMIN)YMIN=Y12(I)                                     S42*0028
      IF(Y12(I).GT.YMAX)YMAX=Y12(I)                                     S42*0029
  135 IF(Y11(I).LT.YMIN)YMIN=Y11(I)                                     S42*0030
      IF(Y11(I).GT.YMAX)YMAX=Y11(I)                                     S42*0031
  140 IF(Y10(I).LT.YMIN)YMIN=Y10(I)                                     S42*0032
      IF(Y10(I).GT.YMAX)YMAX=Y10(I)                                     S42*0033
  145 IF(Y 9(I).LT.YMIN)YMIN=Y 9(I)                                     S42*0034
      IF(Y 9(I).GT.YMAX)YMAX=Y 9(I)                                     S42*0035
  150 IF(Y 8(I).LT.YMIN)YMIN=Y 8(I)                                     S42*0036
      IF(Y 8(I).GT.YMAX)YMAX=Y 8(I)                                     S42*0037
  155 IF(Y 7(I).LT.YMIN)YMIN=Y 7(I)                                     S42*0038
      IF(Y 7(I).GT.YMAX)YMAX=Y 7(I)                                     S42*0039
  160 IF(Y 6(I).LT.YMIN)YMIN=Y 6(I)                                     S42*0040
      IF(Y 6(I).GT.YMAX)YMAX=Y 6(I)                                     S42*0041
  165 IF(Y 5(I).LT.YMIN)YMIN=Y 5(I)                                     S42*0042
      IF(Y 5(I).GT.YMAX)YMAX=Y 5(I)                                     S42*0043
  170 IF(Y 4(I).LT.YMIN)YMIN=Y 4(I)                                     S42*0044
      IF(Y 4(I).GT.YMAX)YMAX=Y 4(I)                                     S42*0045
  175 IF(Y 3(I).LT.YMIN)YMIN=Y 3(I)                                     S42*0046
      IF(Y 3(I).GT.YMAX)YMAX=Y 3(I)                                     S42*0047
  180 IF(Y 2(I).LT.YMIN)YMIN=Y 2(I)                                     S42*0048
      IF(Y 2(I).GT.YMAX)YMAX=Y 2(I)                                     S42*0049
  185 IF(Y 1(I).LT.YMIN)YMIN=Y 1(I)                                     S42*0050
      IF(Y 1(I).GT.YMAX)YMAX=Y 1(I)                                     S42*0051
      IF(X(I).LT.XMIN)XMIN=X(I)                                         S42*0052
      IF(X(I).GT.XMAX)XMAX=X(I)                                         S42*0053
  200 CONTINUE                                                          S42*0054
      IF(XMAX.EQ.XMIN.OR.YMIN.EQ.YMAX)GO TO 900                         S42*0055
      YH=YMAX+(YMAX-YMIN)/18.0                                          S42*0056
      YL=YMIN-(YMAX-YMIN)/18.0                                          S42*0057
      XH=XMAX+(XMAX-XMIN)/24.0                                          S42*0058
      XL=XMIN-(XMAX-XMIN)/24.0                                          S42*0059
```

```
      XMAX=ABS(XH)
      XMIN=ABS(XL)
      YMIN=ABS(YL)
      YMAX=ABS(YH)
      IF(XMIN.GT.XMAX)XMAX=XMIN
      IF(YMIN.GT.YMAX)YMAX=YMIN
      XMAX=ALOG10(XMAX)
      YMAX=ALOG10(YMAX)
      IF(XMAX.LT.0.0)XMAX=XMAX-1.0
      IF(YMAX.LT.0.0)YMAX=YMAX-1.0
      MX=-XMAX
      MY=-YMAX
      WRITE(LOG1,250)MX,MY
250   FORMAT(20X,46HSCALES - ≠X≠ IS SHOWN TIMES 10 TO THE POWER OF,I3,40
     1H   ≠Y≠ IS SHOWN TIMES 10 TO THE POWER OF,I3,/)
      YINC=(YH-YL)/54.0
      YINC2=YINC/2.0
      XRANGE=XH-XL
      DO 750 KLINE=1,55
      IF(KLINE.EQ.1.OR.KLINE.EQ.55)GO TO 350
      DO 265 L=2,120
265   LINE(L)=BLANK
      IF(KLINE.EQ.7.OR.KLINE.EQ.13.OR.KLINE.EQ.19.OR.KLINE.EQ.25.OR.KLIN
     1E.EQ.31.OR.KLINE.EQ.37.OR.KLINE.EQ.43.OR.KLINE.EQ.49)GO TO 300
      LINE(1)=XI
      LINE(121)=XI
      GO TO 400
300   LINE(1)=DASH
      LINE(121)=DASH
      GO TO 400
350   DO 360 L=2,120
360   LINE(L)=DASH
      LINE(1)=CROSS
      LINE(121)=CROSS
      DO 365 L=11,111,10
365   LINE(L)=XI
      GO TO 650
400   DO 600 I=1,IX
      GO TO(590,585,580,575,570,565,560,555,550,545,540,535,530,525,520,
     1515,510,505),IY
505   IF(Y18(I).GT.YH+YINC2.OR.Y18(I).LE.YH-YINC2)GO TO 510
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL(18)
510   IF(Y17(I).GT.YH+YINC2.OR.Y17(I).LE.YH-YINC2)GO TO 515
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL(17)
515   IF(Y16(I).GT.YH+YINC2.OR.Y16(I).LE.YH-YINC2)GO TO 520
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL(16)
520   IF(Y15(I).GT.YH+YINC2.OR.Y15(I).LE.YH-YINC2)GO TO 525
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL(15)
525   IF(Y14(I).GT.YH+YINC2.OR.Y14(I).LE.YH-YINC2)GO TO 530
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL(14)
530   IF(Y13(I).GT.YH+YINC2.OR.Y13(I).LE.YH-YINC2)GO TO 535
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL(13)
535   IF(Y12(I).GT.YH+YINC2.OR.Y12(I).LE.YH-YINC2)GO TO 540
      L=(X(I)-XL)/XRANGE*120.0+1.5
```

```
      LINE(L)=SYMBOL(12)
540   IF(Y11(I).GT.YH+YINC2.OR.Y11(I).LE.YH-YINC2)GO TO 545
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL(11)
545   IF(Y10(I).GT.YH+YINC2.OR.Y10(I).LE.YH-YINC2)GO TO 550
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL(10)
550   IF(Y 9(I).GT.YH+YINC2.OR.Y 9(I).LE.YH-YINC2)GO TO 555
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL( 9)
555   IF(Y 8(I).GT.YH+YINC2.OR.Y 8(I).LE.YH-YINC2)GO TO 560
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL( 8)
560   IF(Y 7(I).GT.YH+YINC2.OR.Y 7(I).LE.YH-YINC2)GO TO 565
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL( 7)
565   IF(Y 6(I).GT.YH+YINC2.OR.Y 6(I).LE.YH-YINC2)GO TO 570
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL( 6)
570   IF(Y 5(I).GT.YH+YINC2.OR.Y 5(I).LE.YH-YINC2)GO TO 575
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL( 5)
575   IF(Y 4(I).GT.YH+YINC2.OR.Y 4(I).LE.YH-YINC2)GO TO 580
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL( 4)
580   IF(Y 3(I).GT.YH+YINC2.OR.Y 3(I).LE.YH-YINC2)GO TO 585
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL( 3)
585   IF(Y 2(I).GT.YH+YINC2.OR.Y 2(I).LE.YH-YINC2)GO TO 590
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL( 2)
590   IF(Y 1(I).GT.YH+YINC2.OR.Y 1(I).LE.YH-YINC2)GO TO 600
      L=(X(I)-XL)/XRANGE*120.0+1.5
      LINE(L)=SYMBOL( 1)
600   CONTINUE
      IF(KLINE.EQ.1.OR.KLINE.EQ.7.OR.KLINE.EQ.13.OR.KLINE.EQ.19.OR.KLINE
     1.EQ.25.OR.KLINE.EQ.31.OR.KLINE.EQ.37.OR.KLINE.EQ.43.OR.KLINE.EQ.49
     2.OR.KLINE.EQ.55)GO TO 650
      WRITE(LOG1,610)LINE
610   FORMAT(8X,121A1)
      GO TO 750
650   YNUM=YH*10.0**MY
      WRITE(LOG1,655)YNUM,LINE
655   FORMAT(1X,F6.3,1X,121A1)
750   YH=YH-YINC
      XNUM(1)=XL*10.0**MX
      XINC=((XH-XL)/12.0)*10.0**MX
      DO 800 I=2,13
800   XNUM(I)=XNUM(I-1)+XINC
      WRITE(LOG1,820)XNUM
820   FORMAT(6X,12(F6.3,4X),F6.3)
      RETURN
900   WRITE(LOG1,910)
910   FORMAT(//,35X,54HNO PLOT HAS BEEN MADE BECAUSE ≠X≠ OR ≠Y≠ RANGE IS
     1 ZERO)
      RETURN
      END
```

Section IV

CONCLUSIONS & RECOMMENDATIONS

The program objective of demonstrating a computer program to predict compressor performance with combined radial and circumferential distortion has been achieved. Limited use of the program has, as yet, been made, and several work areas now present themselves. These include the establishment of the validity of the results given by the computer program for a variety of compressor types, running at various conditions, and then the use of the program to determine the type of compressor design best suited to various applications involving distorted flow. Of course, the effect of any distortion pattern upon a particular existing compressor design may be assessed.

Some further program development could be done, mainly to establish the optimum, that is cheapest, manner in which the solution to any problem may be obtained.

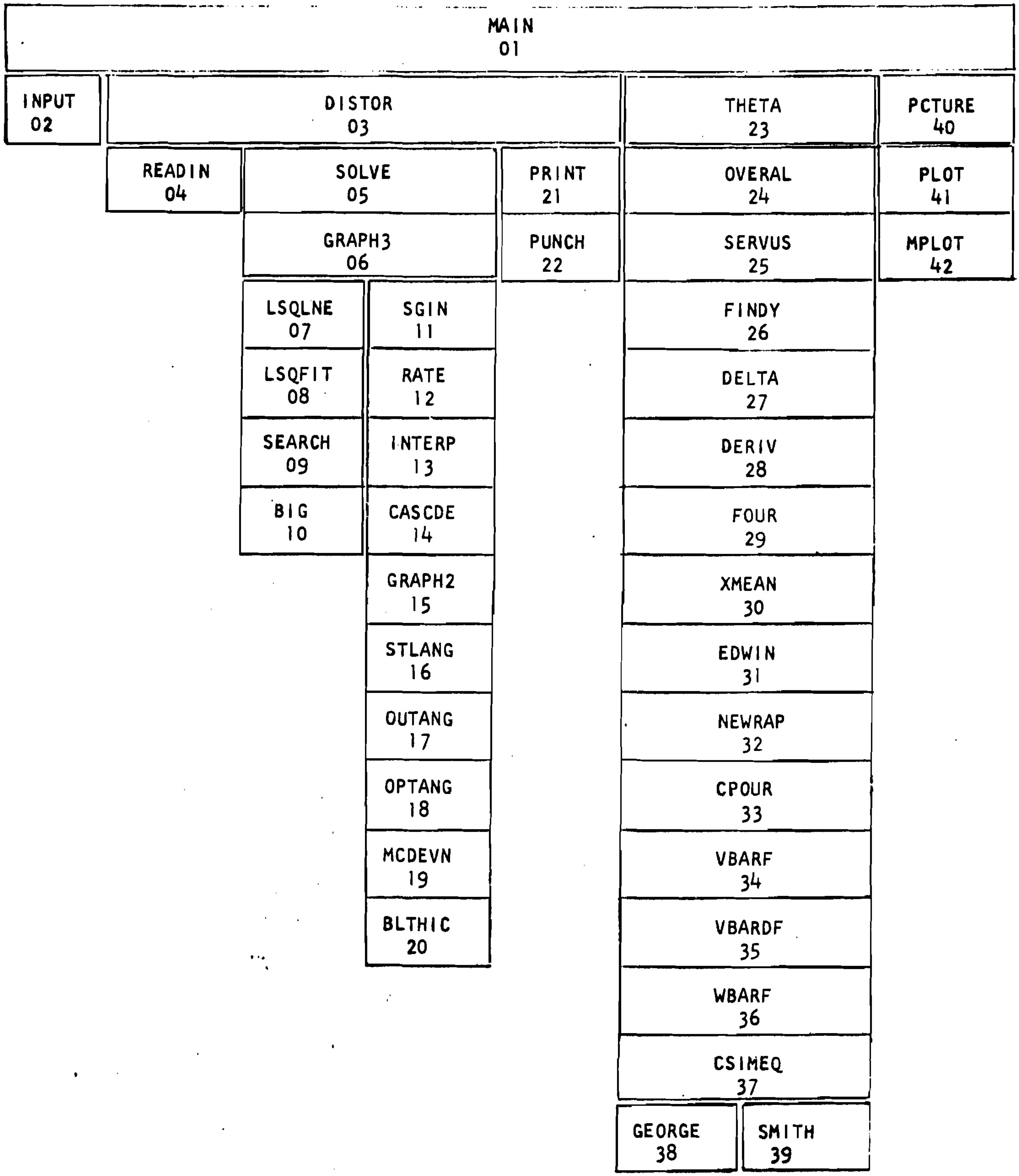


FIGURE 1 - OVERLAY STRUCTURE

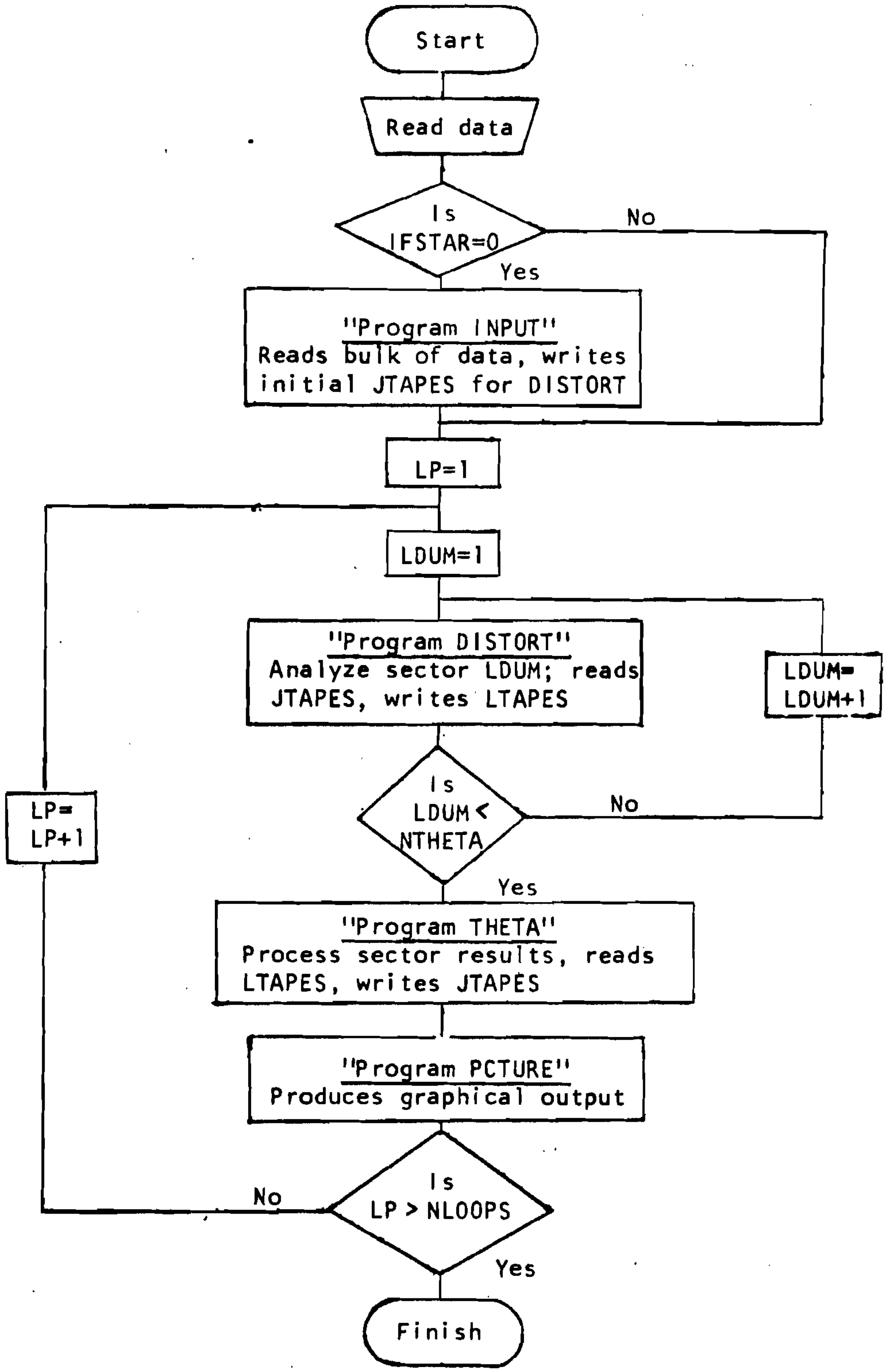


FIGURE 2 - OUTLINE FLOW CHART

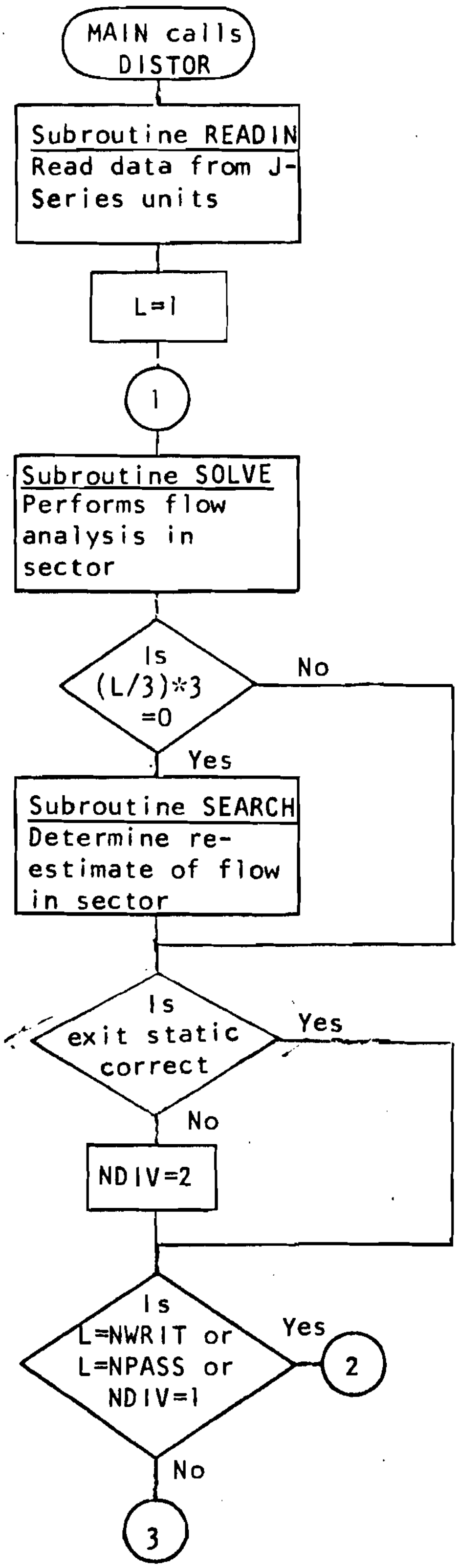


FIGURE 3 - FLOW CHART FOR "PROGRAM DISTORT"

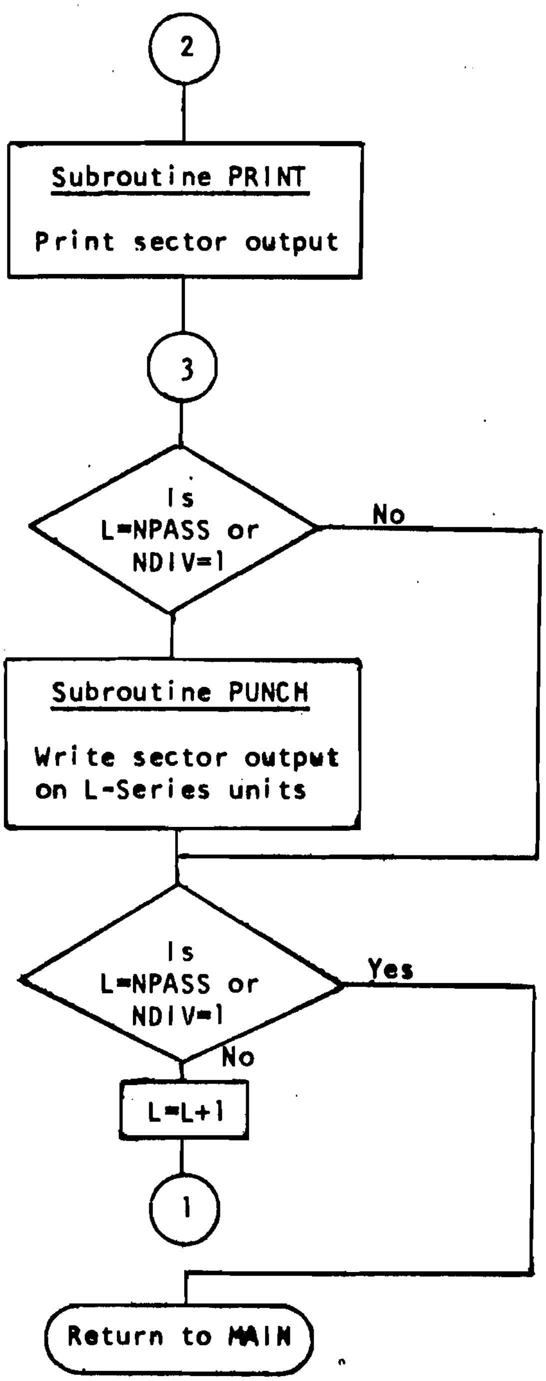


FIGURE 3 - FLOW CHART FOR "PROGRAM DISTORT" (CONTINUED)

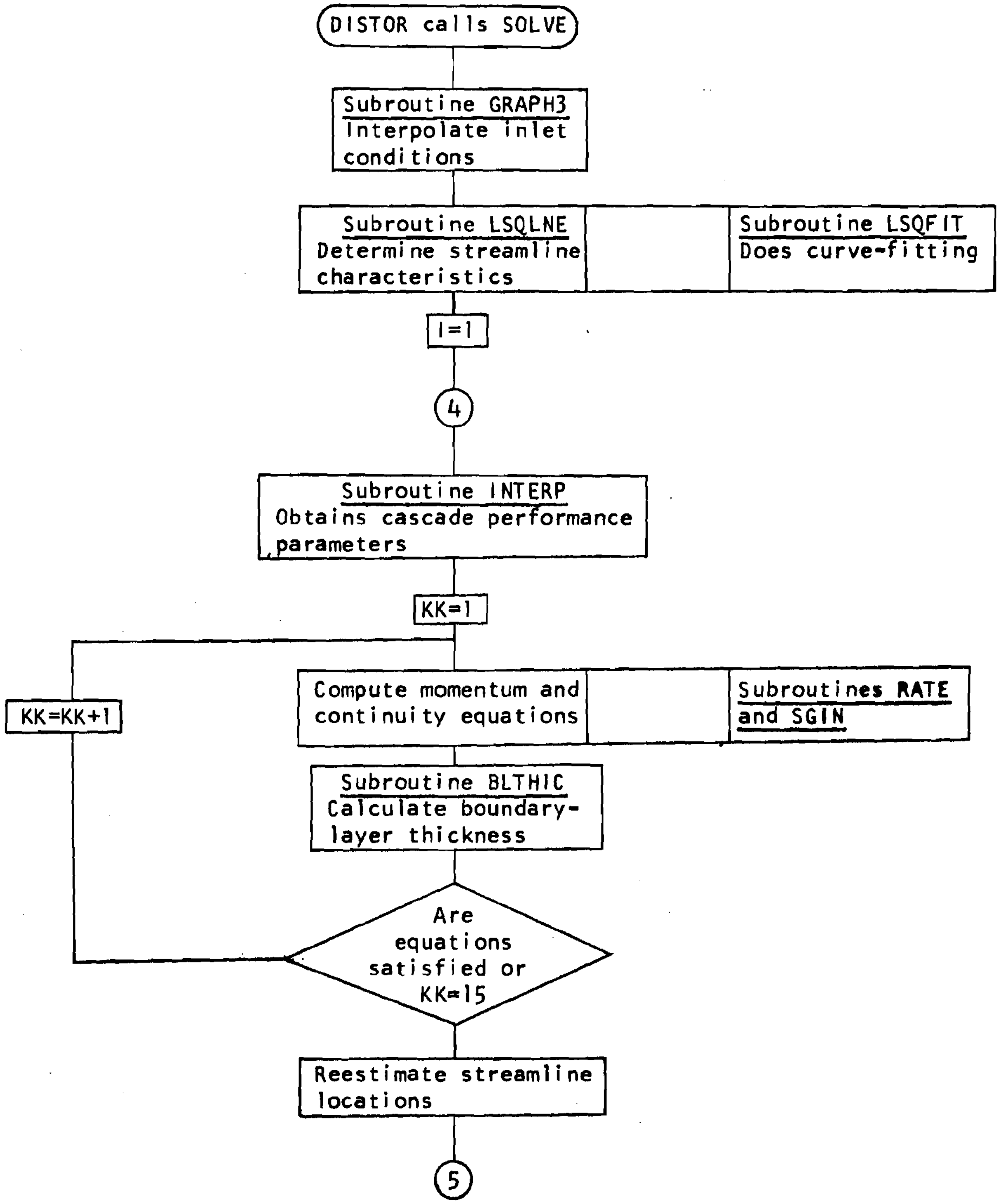


FIGURE 3 - FLOW CHART FOR "PROGRAM DISTORT" (CONTINUED)

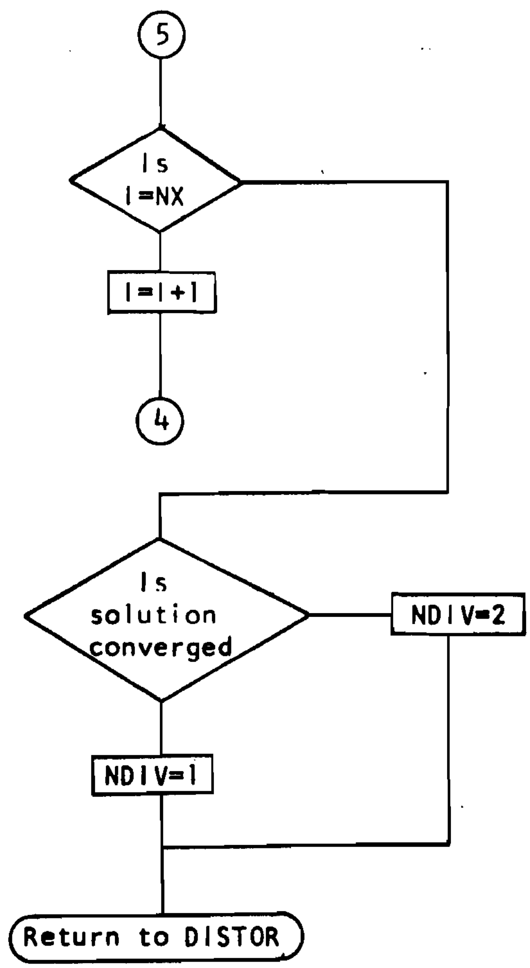


FIGURE 3 - FLOW CHART FOR "PROGRAM DISTORT" (CONTINUED)

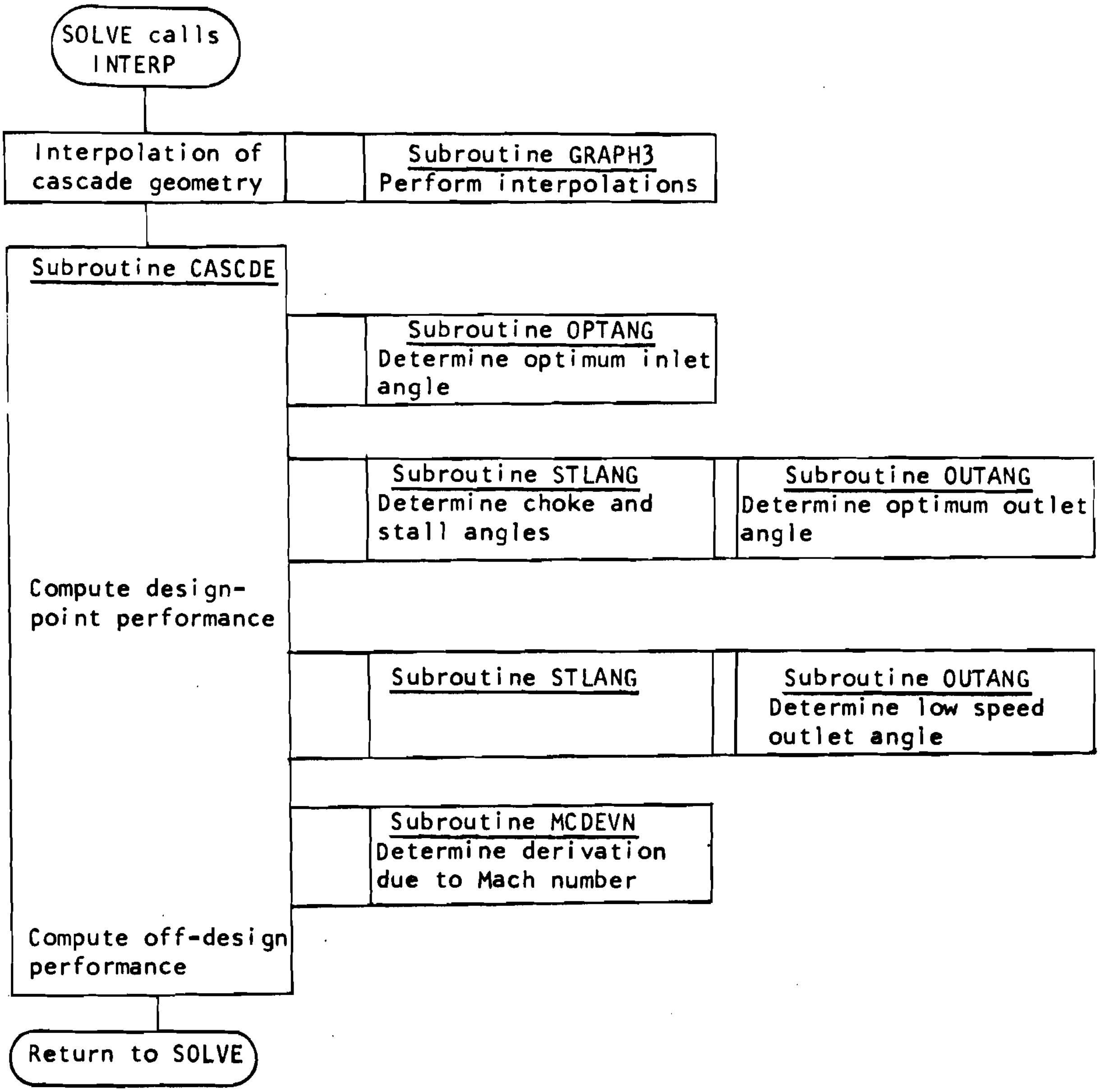


FIGURE 3 - FLOW CHART FOR "PROGRAM DISTORT" (CONTINUED)

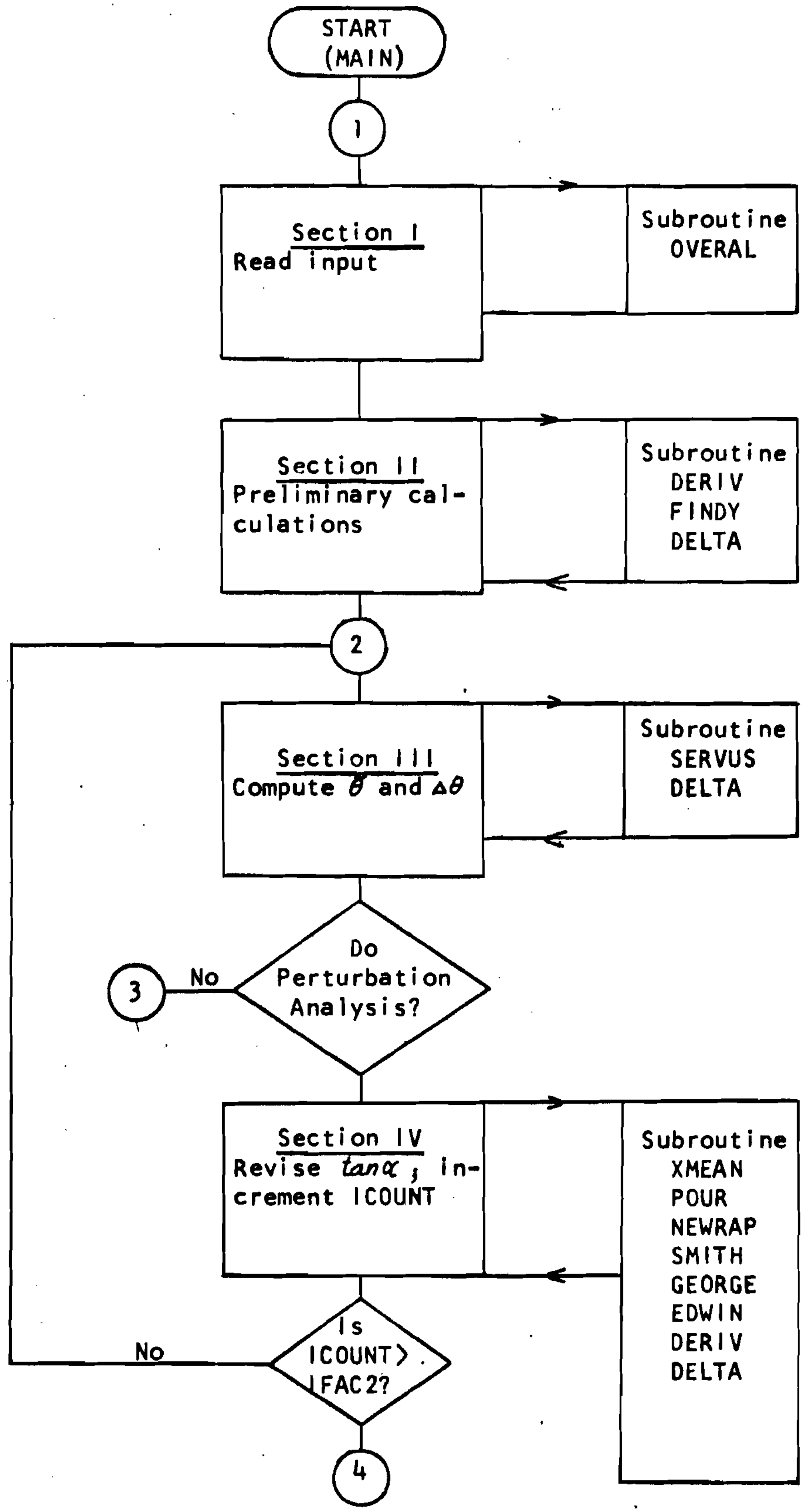


FIGURE 4 - FLOW DIAGRAM FOR "PROGRAM THETA"

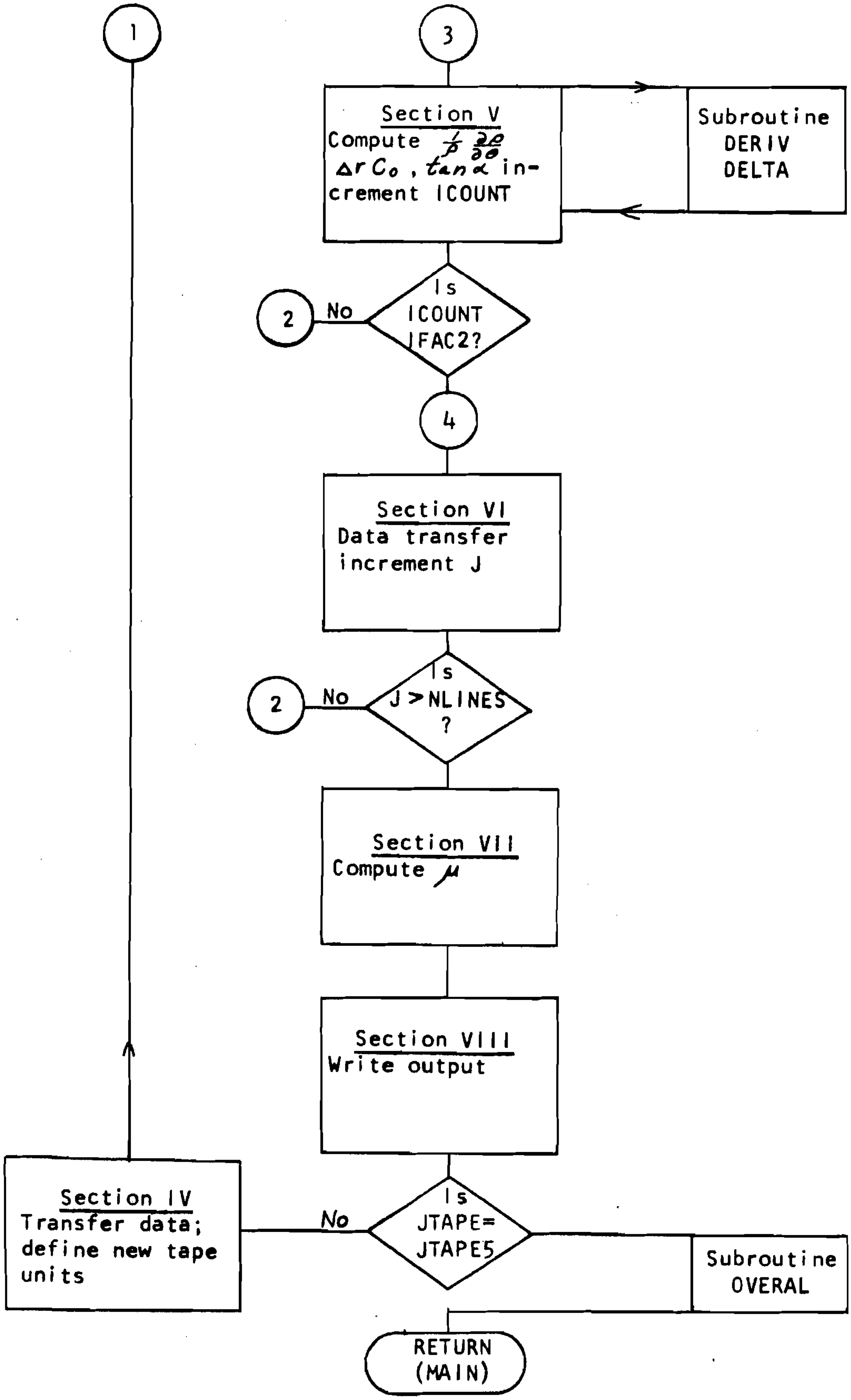


FIGURE 4 - FLOW DIAGRAM FOR "PROGRAM THETA" (CONTINUED)

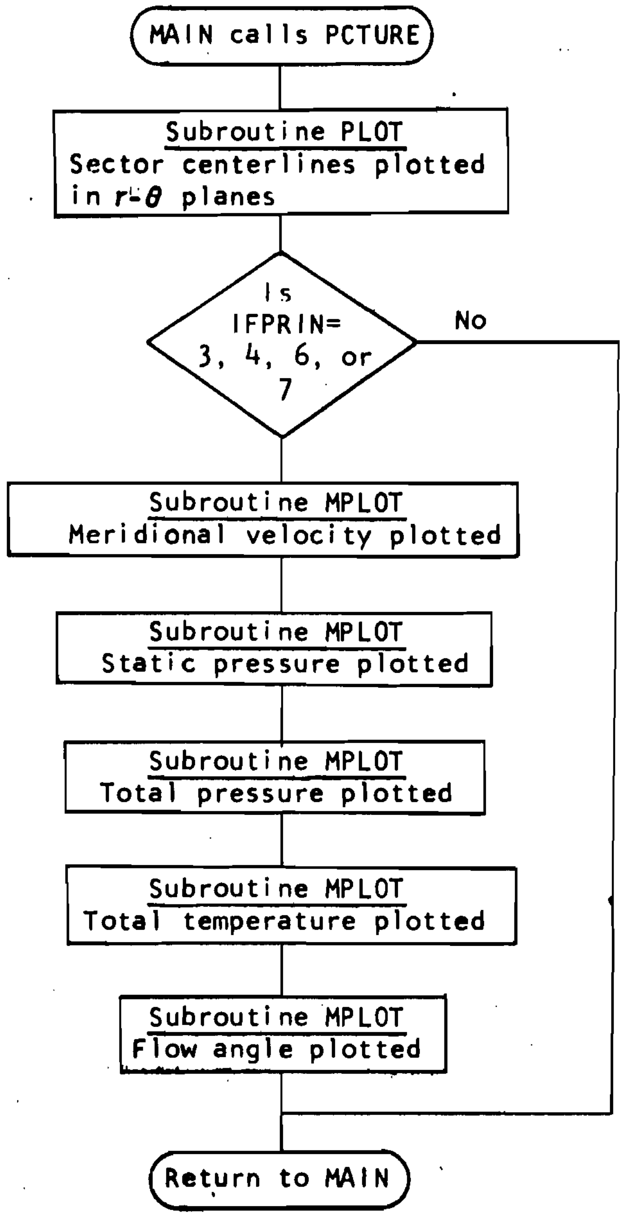


FIGURE 5 - FLOW CHART FOR "PROGRAM PICTURE"

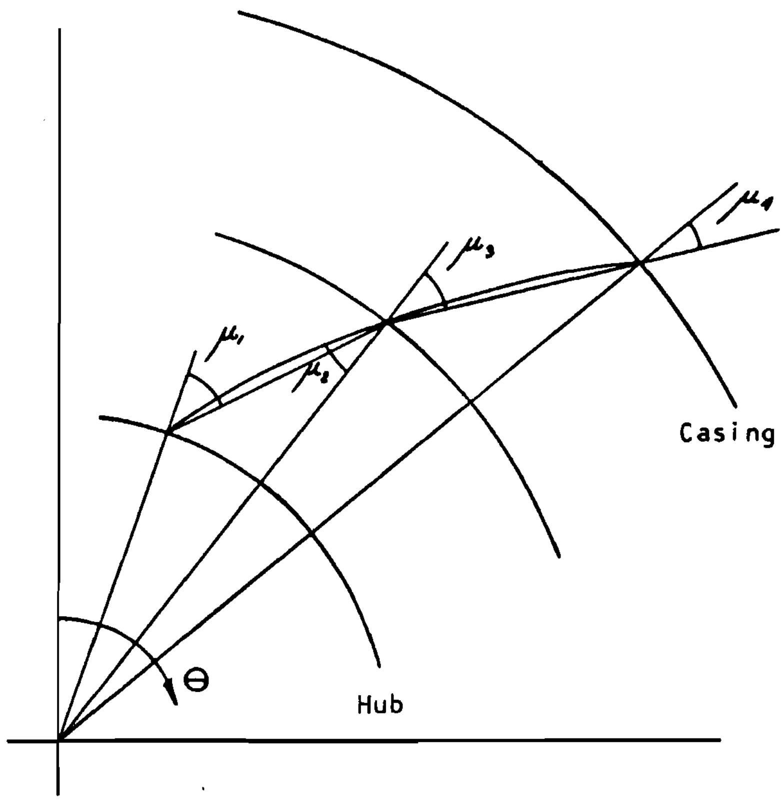


FIGURE 6 - CALCULATION OF μ

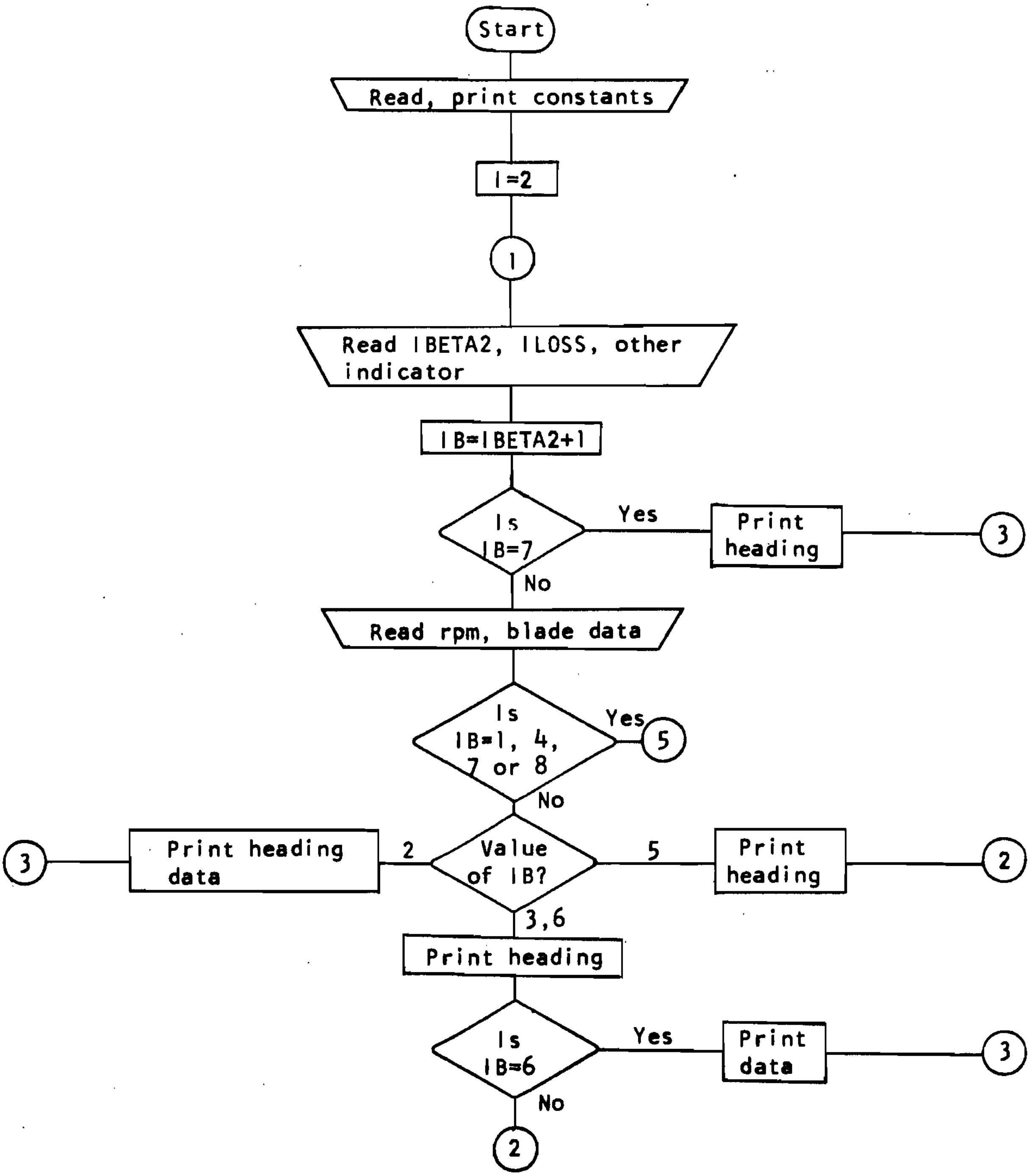


FIGURE 7 - FLOW CHART FOR SUBROUTINE INPUT

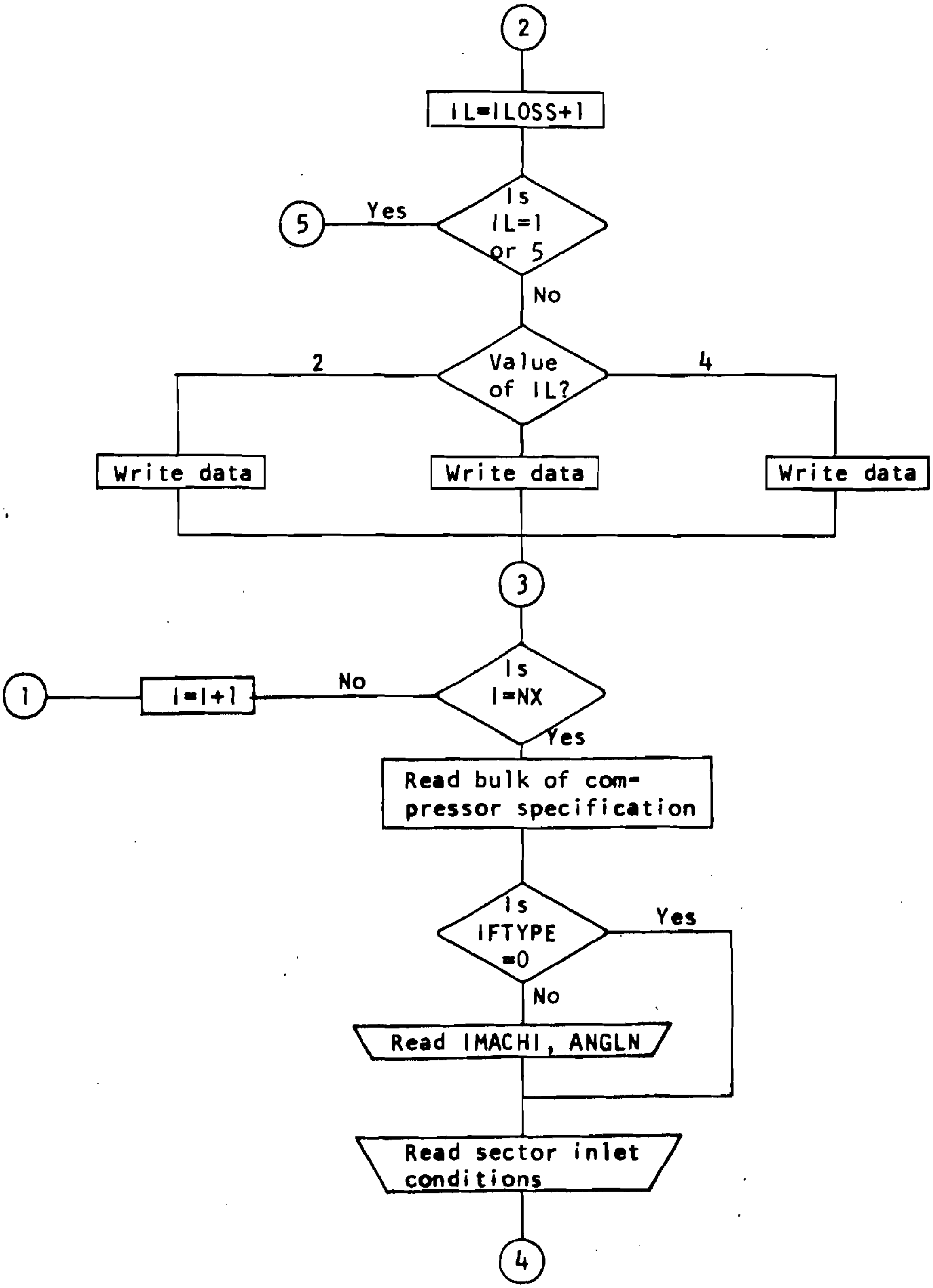


FIGURE 7 - FLOW CHART FOR SUBROUTINE INPUT (CONTINUED)

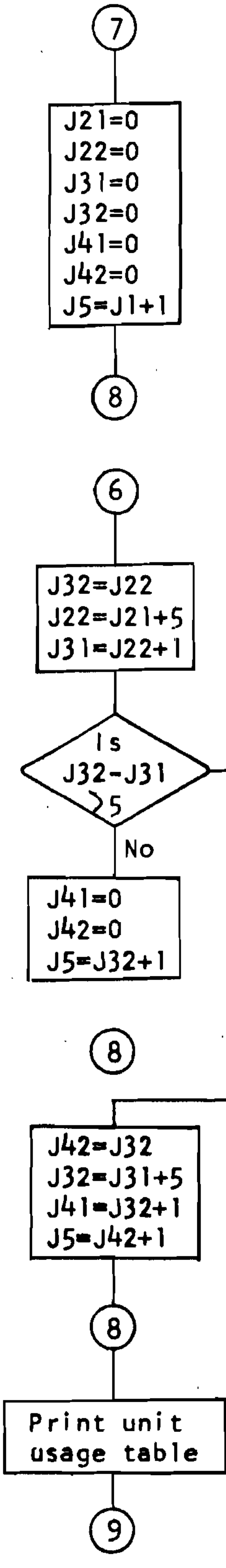


FIGURE 7 - FLOW CHART FOR SUBROUTINE INPUT (CONTINUED)

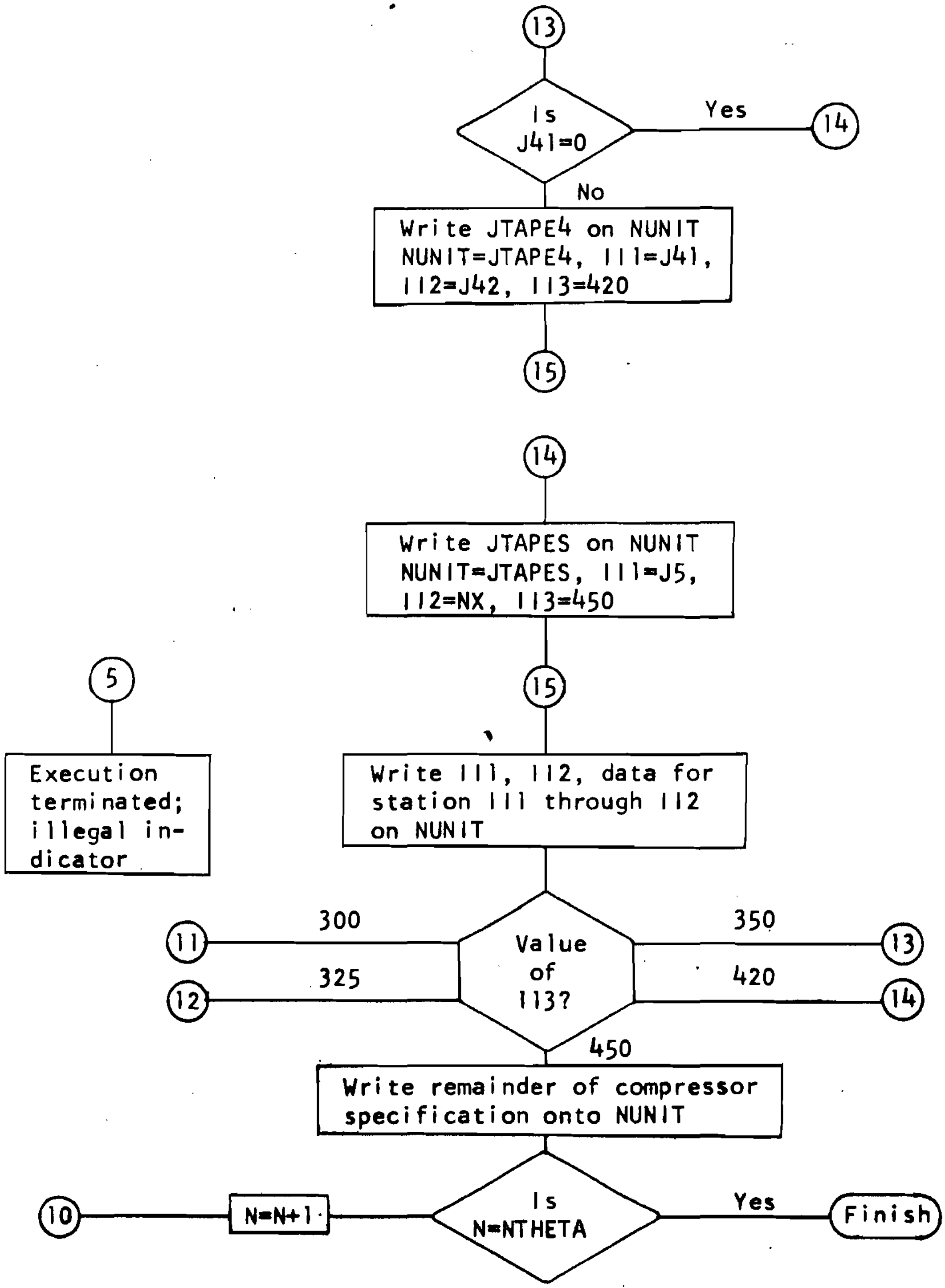


FIGURE 7 - FLOW CHART FOR SUBROUTINE INPUT (CONTINUED)

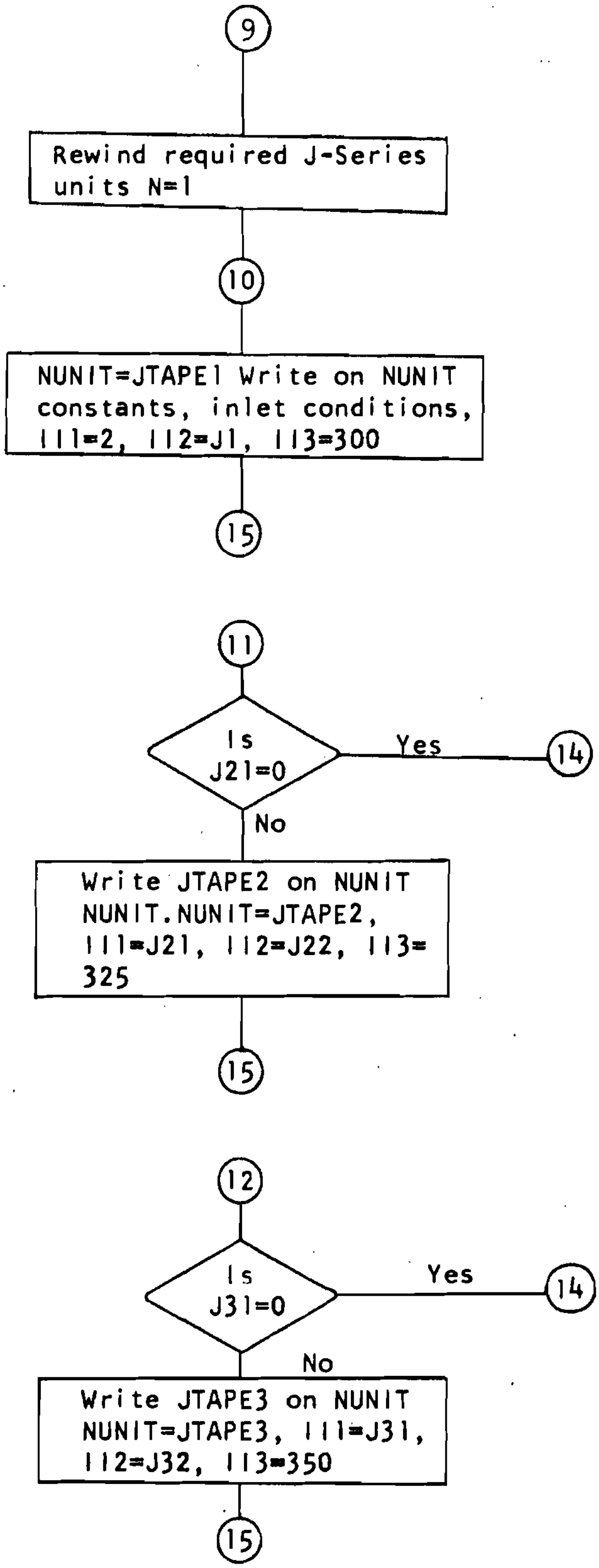


FIGURE 7 - FLOW CHART FOR SUBROUTINE INPUT (CONTINUED)

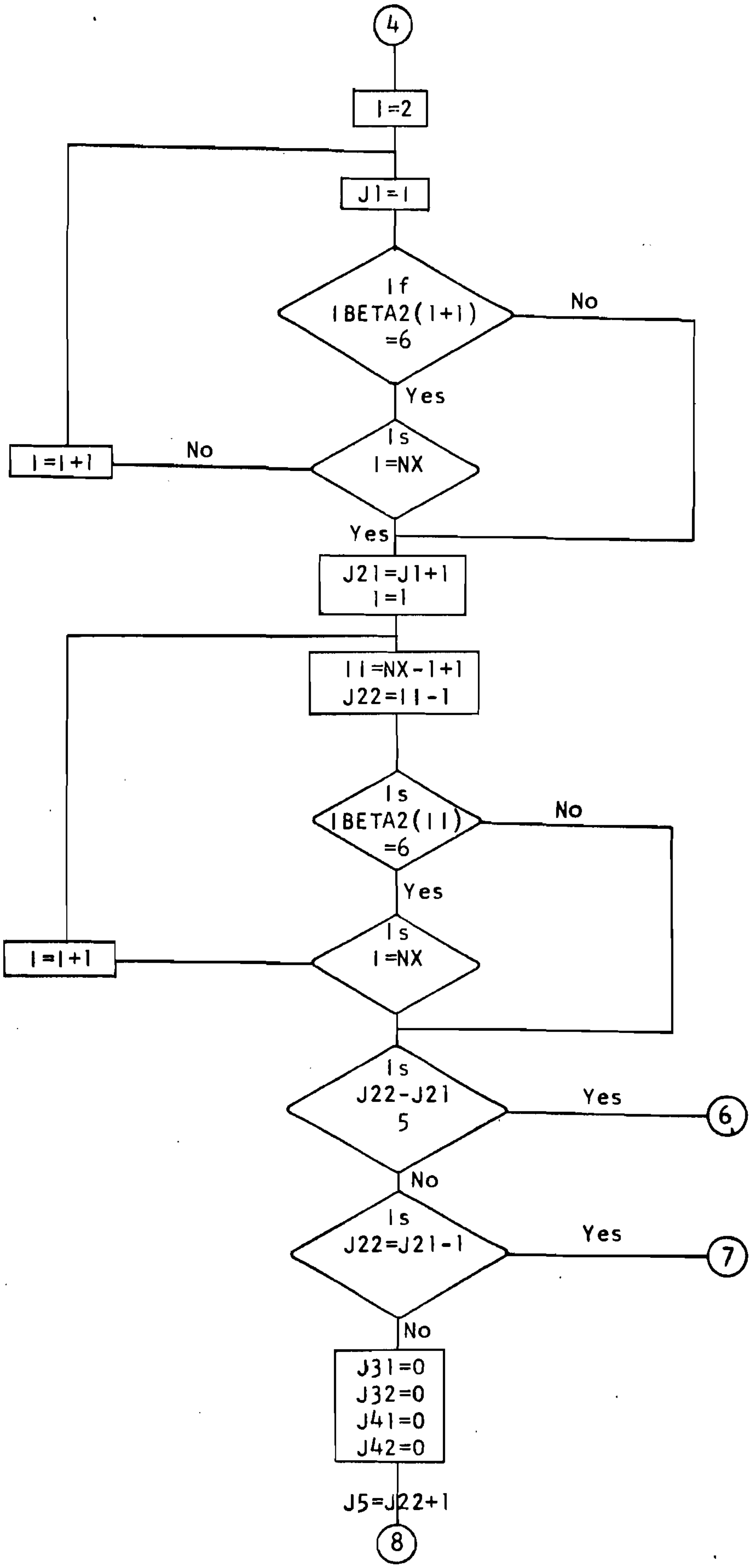


FIGURE 7 - FLOW CHART FOR SUBROUTINE INPUT (CONTINUED)

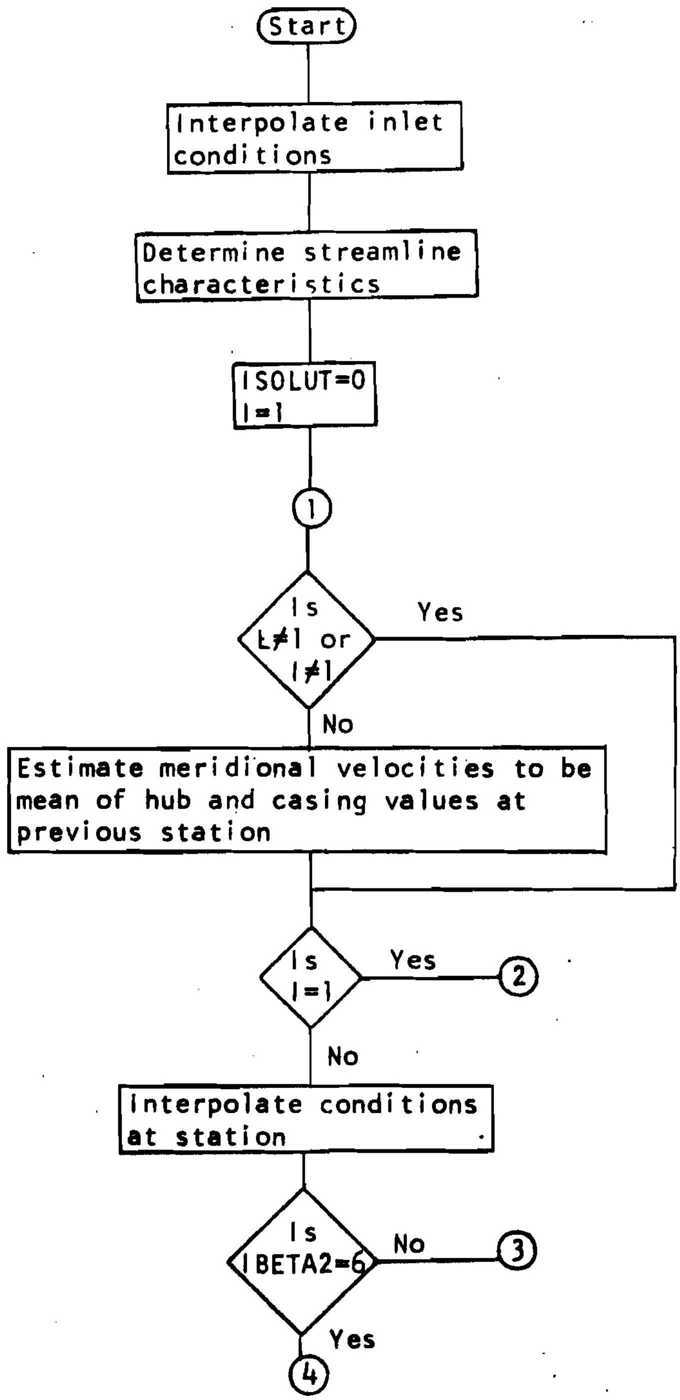


FIGURE 8 - FLOW CHART FOR SUBROUTINE SOLVE

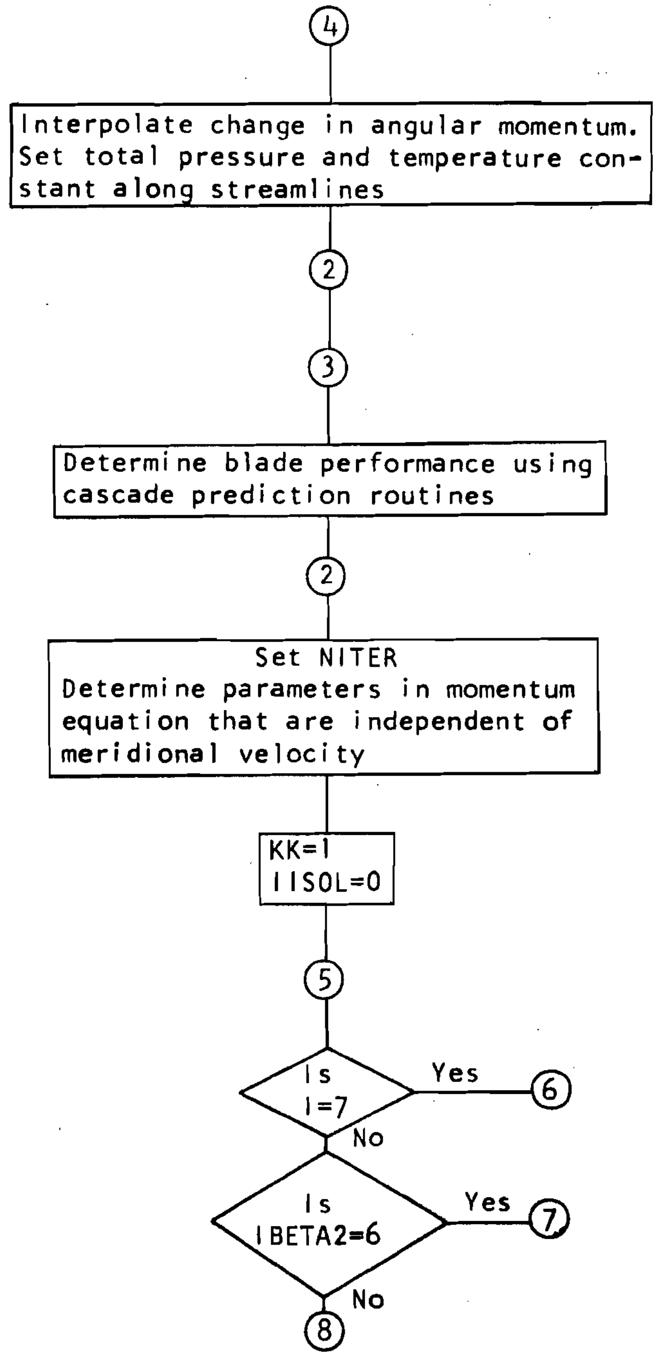


FIGURE 8 - FLOW CHART FOR SUBROUTINE SOLVE (CONTINUED)

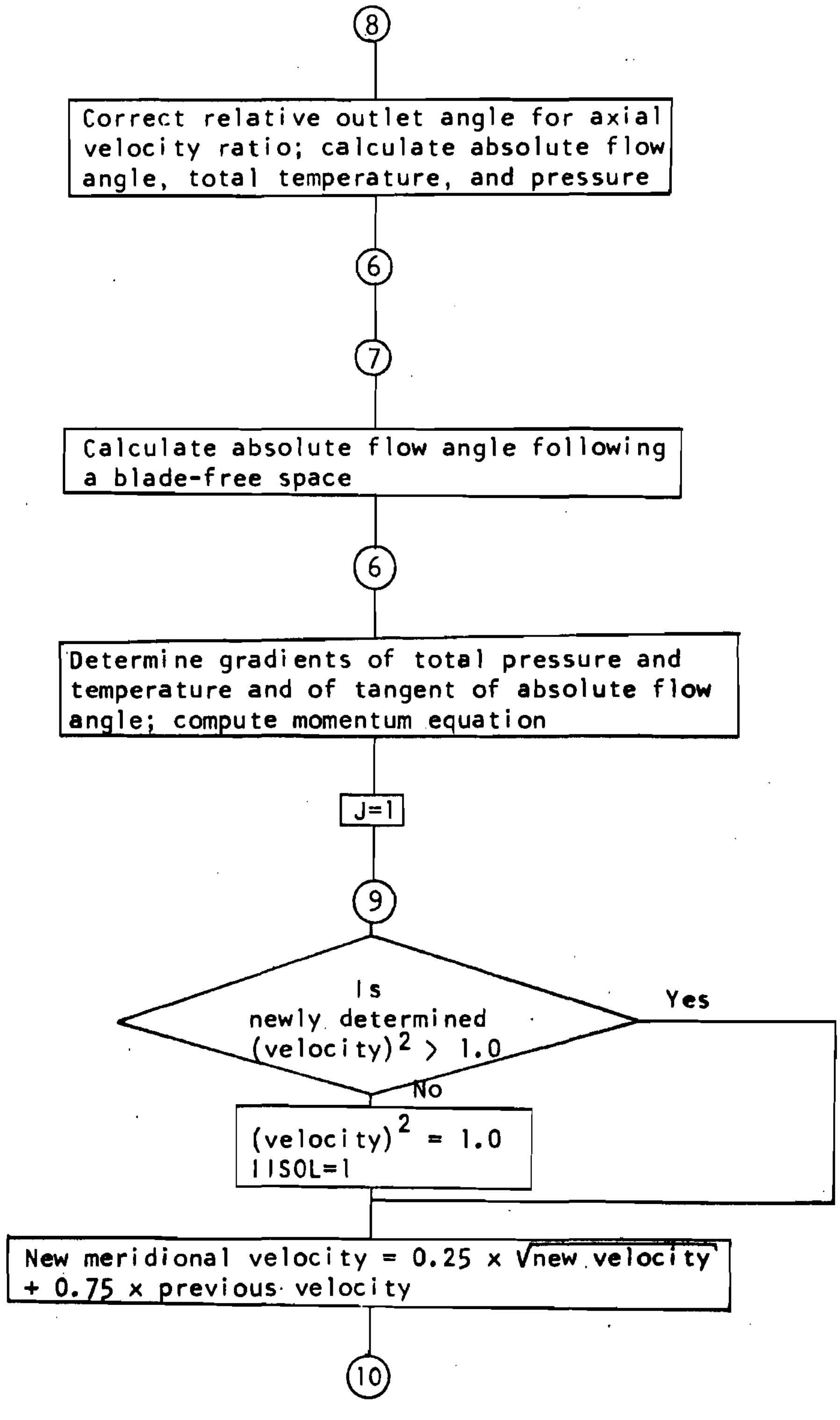


FIGURE 8 - FLOW CHART FOR SUBROUTINE SOLVE (CONTINUED)

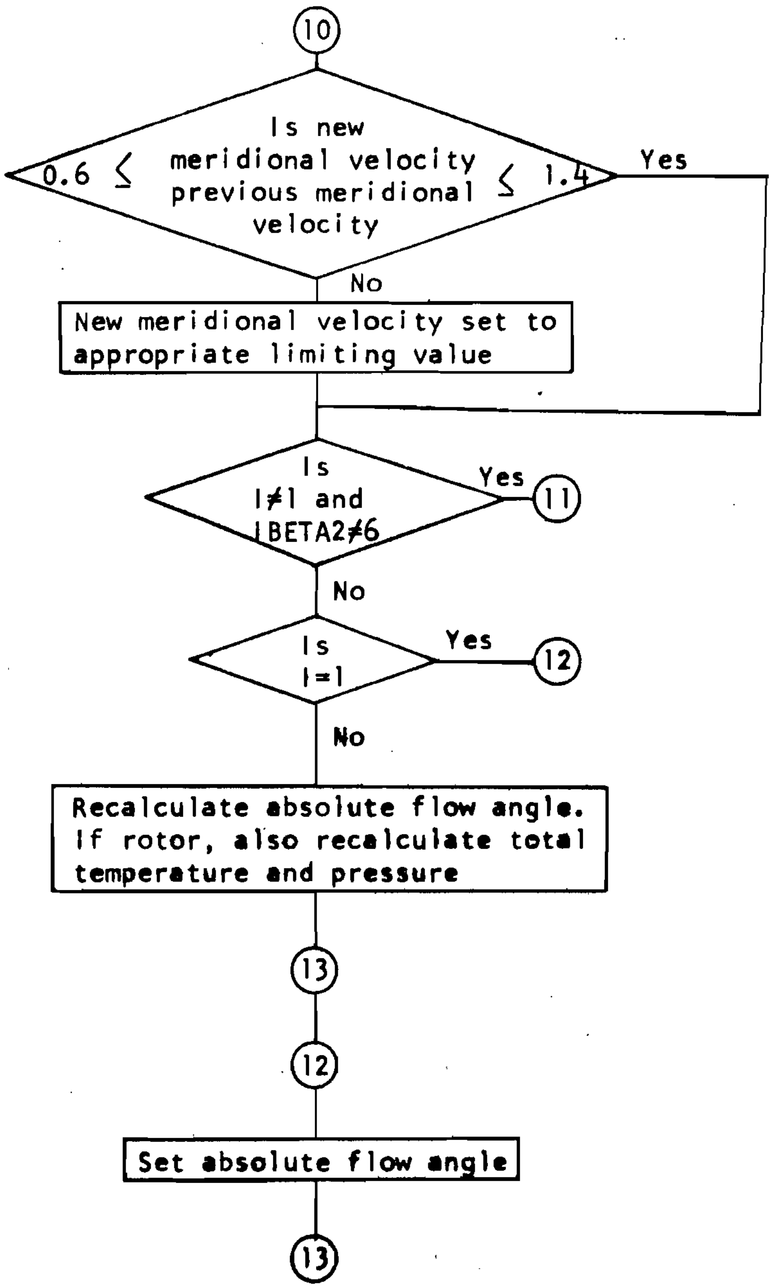


FIGURE 8 - FLOW CHART FOR SUBROUTINE SOLVE (CONTINUED)

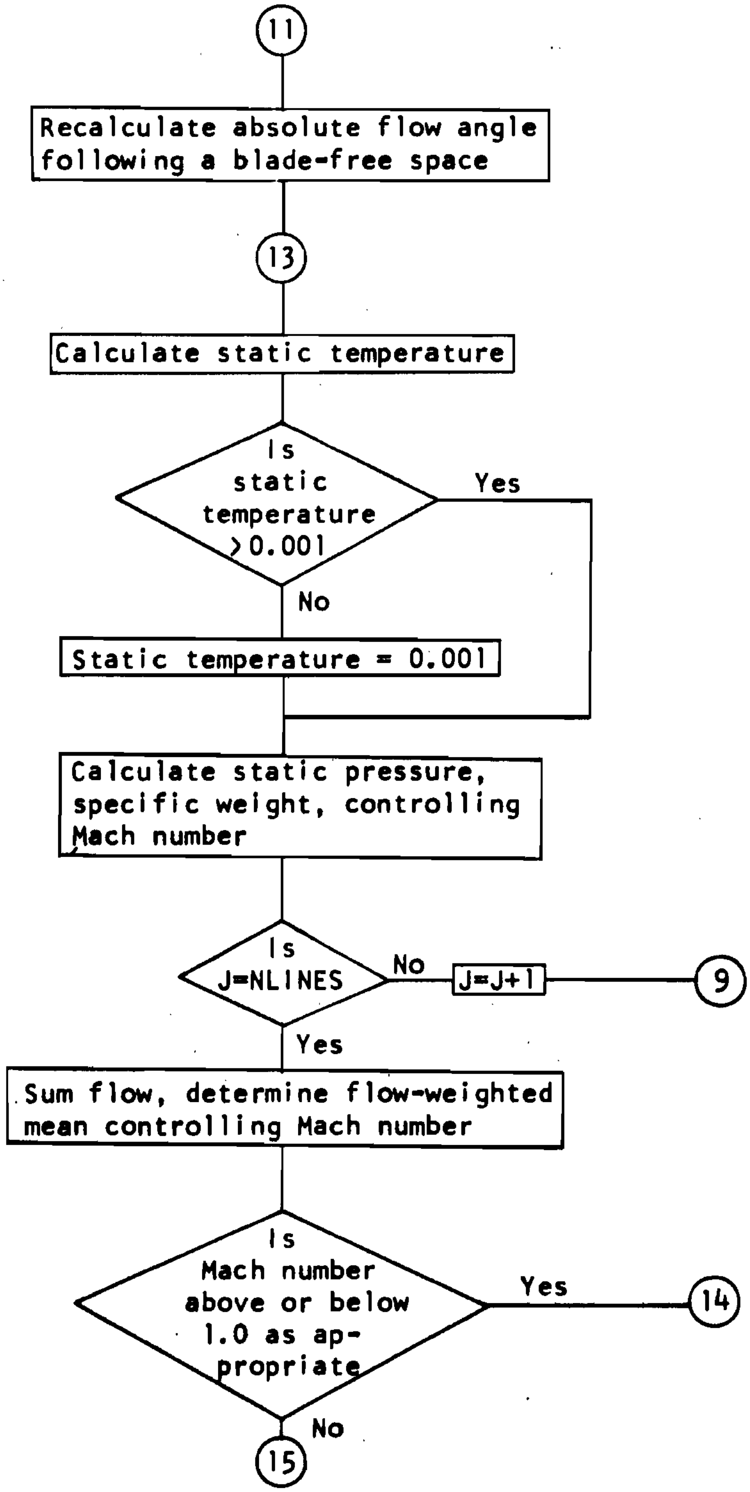


FIGURE 8 - FLOW CHART FOR SUBROUTINE SOLVE (CONTINUED)

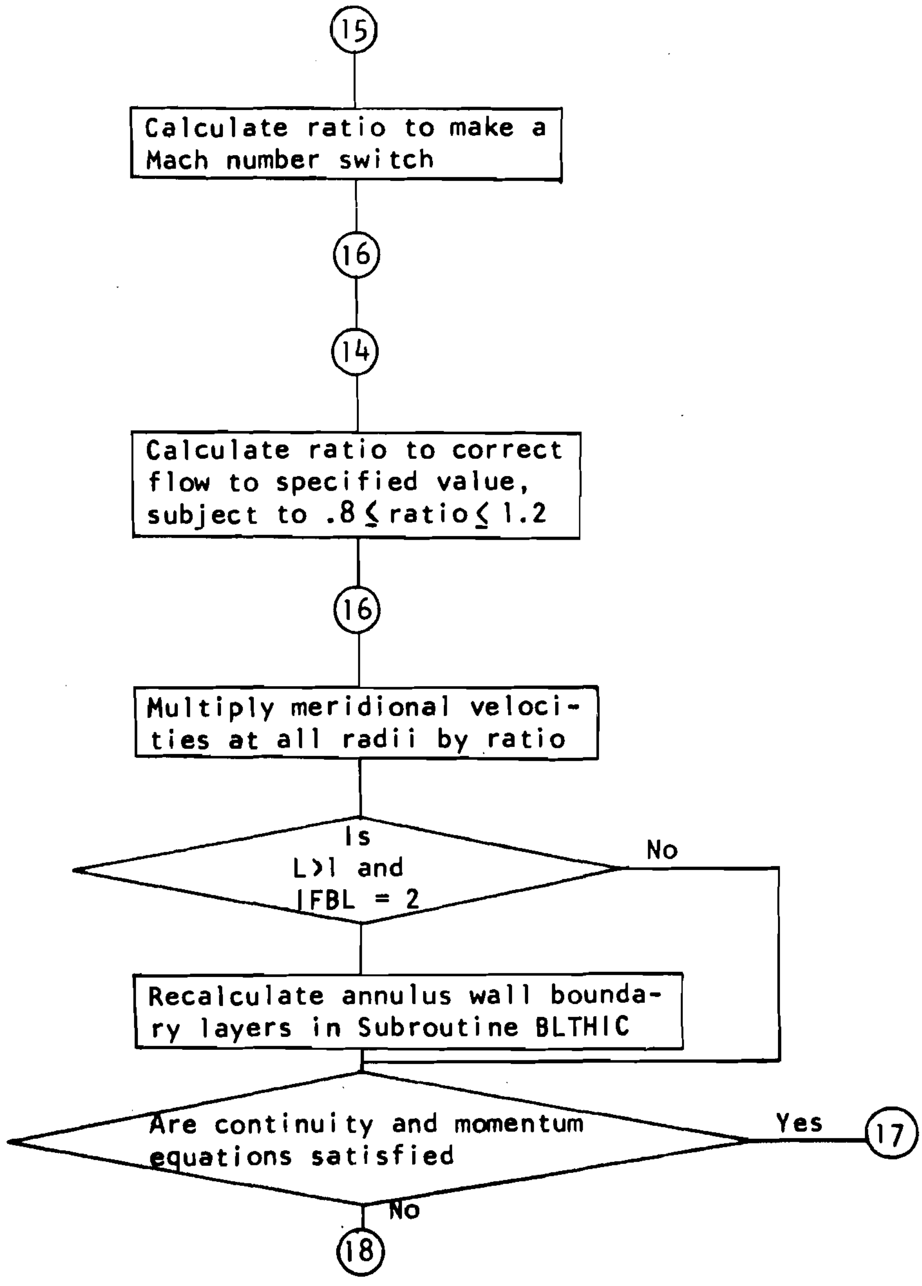


FIGURE 8 - FLOW CHART FOR SUBROUTINE SOLVE (CONTINUED)

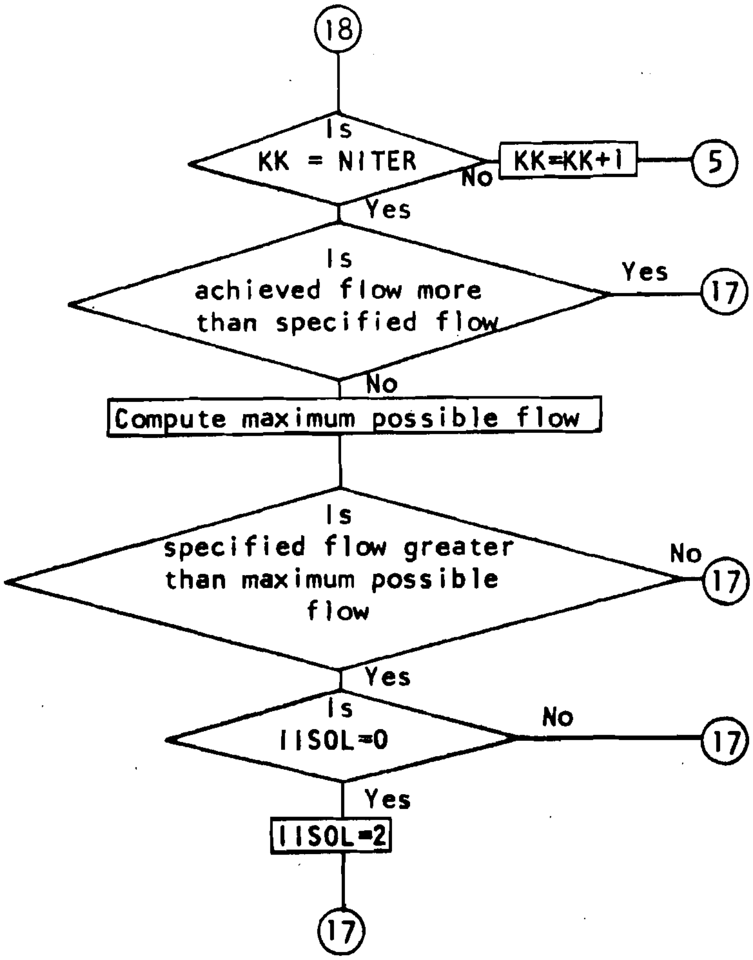


FIGURE 8 - FLOW CHART FOR SUBROUTINE SOLVE (CONTINUED)

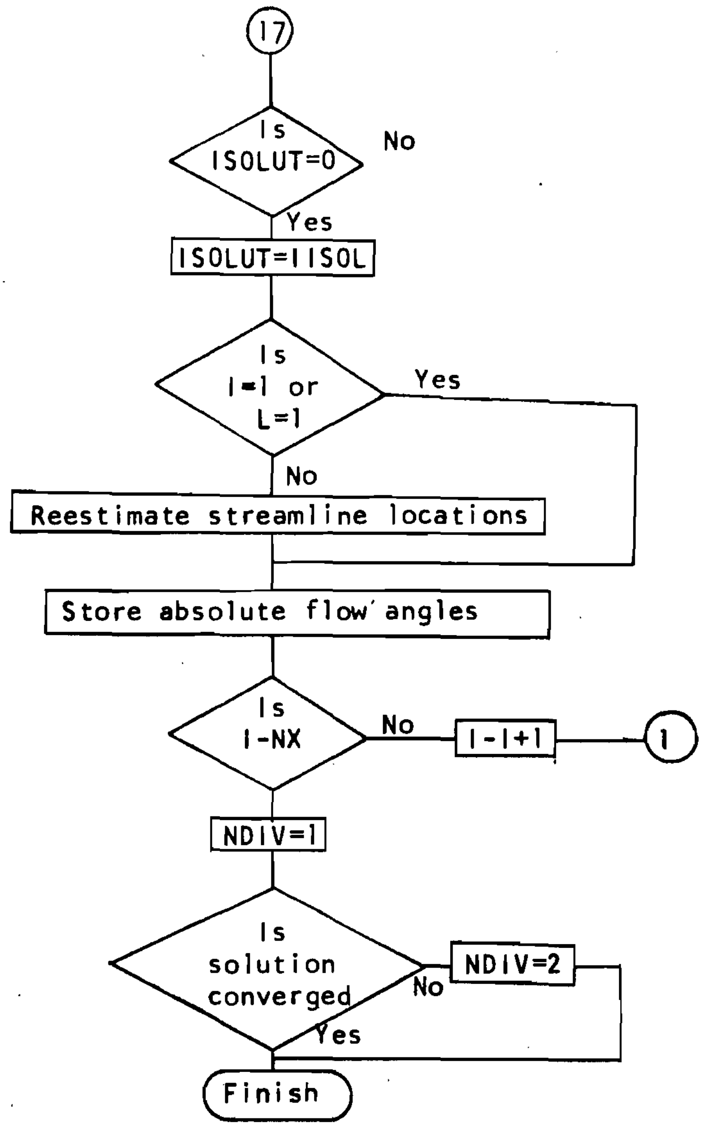


FIGURE 8 - FLOW CHART FOR SUBROUTINE SOLVE (CONTINUED)

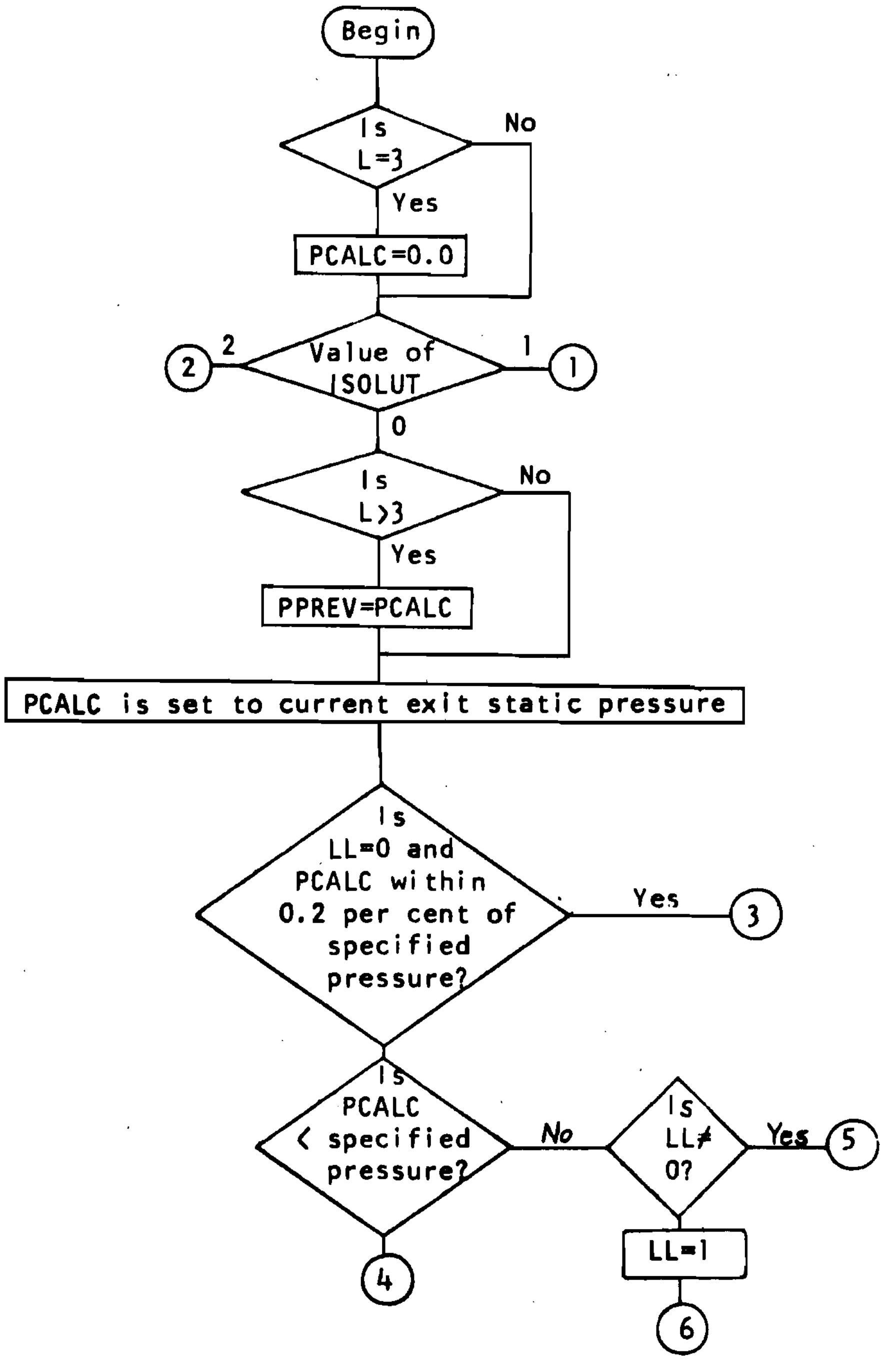


FIGURE 9 - FLOW CHART FOR SUBROUTINE SEARCH

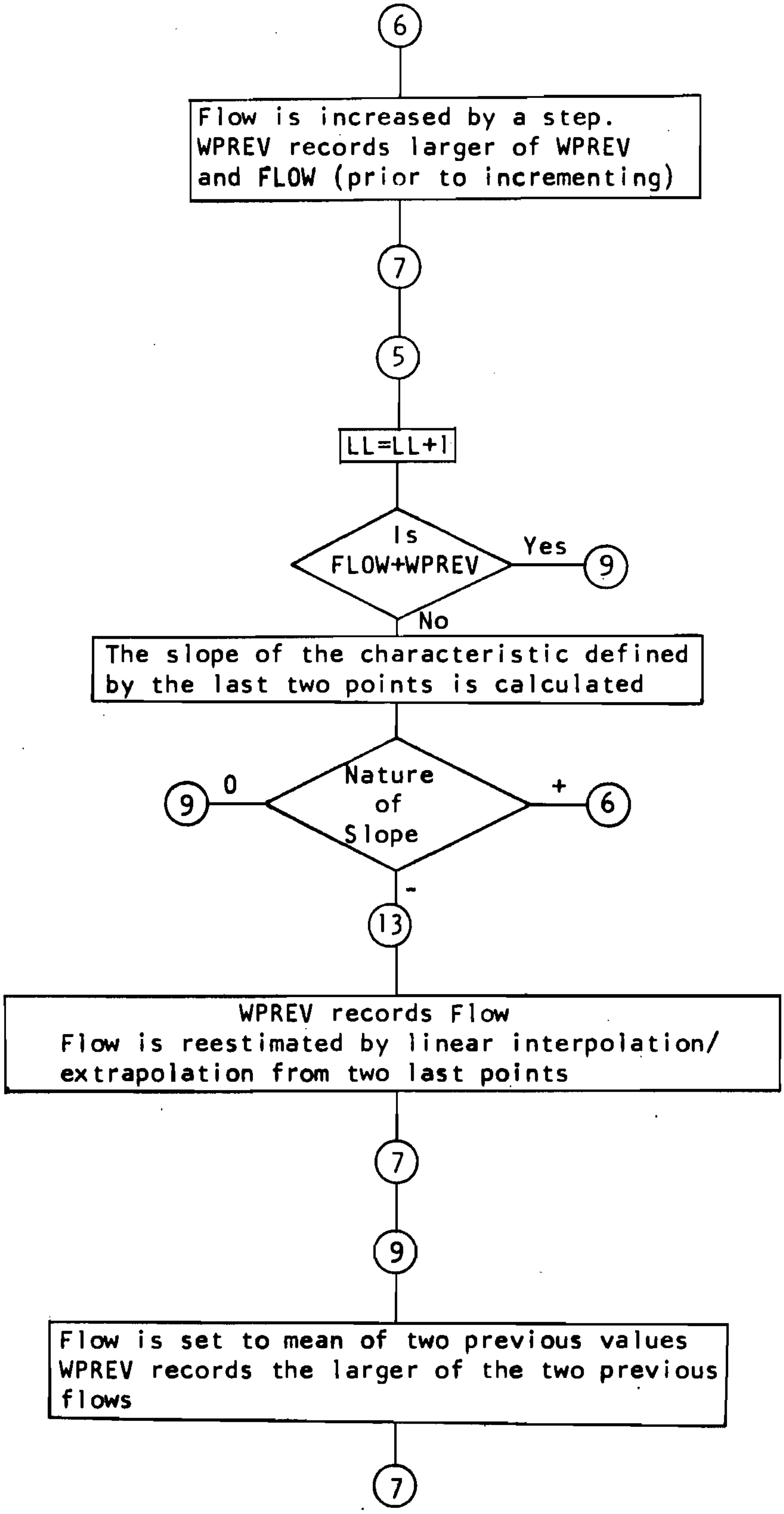


FIGURE 9 - FLOW CHART FOR SUBROUTINE SEARCH (CONTINUED)

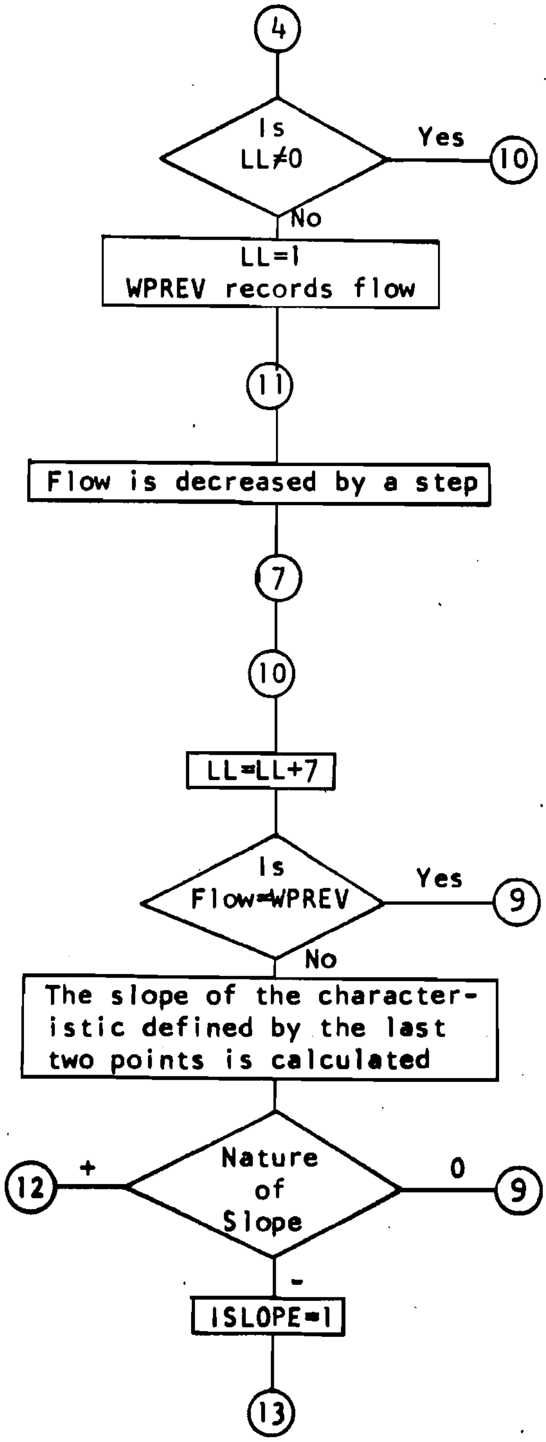


FIGURE 9 - FLOW CHART FOR SUBROUTINE SEARCH (CONTINUED)

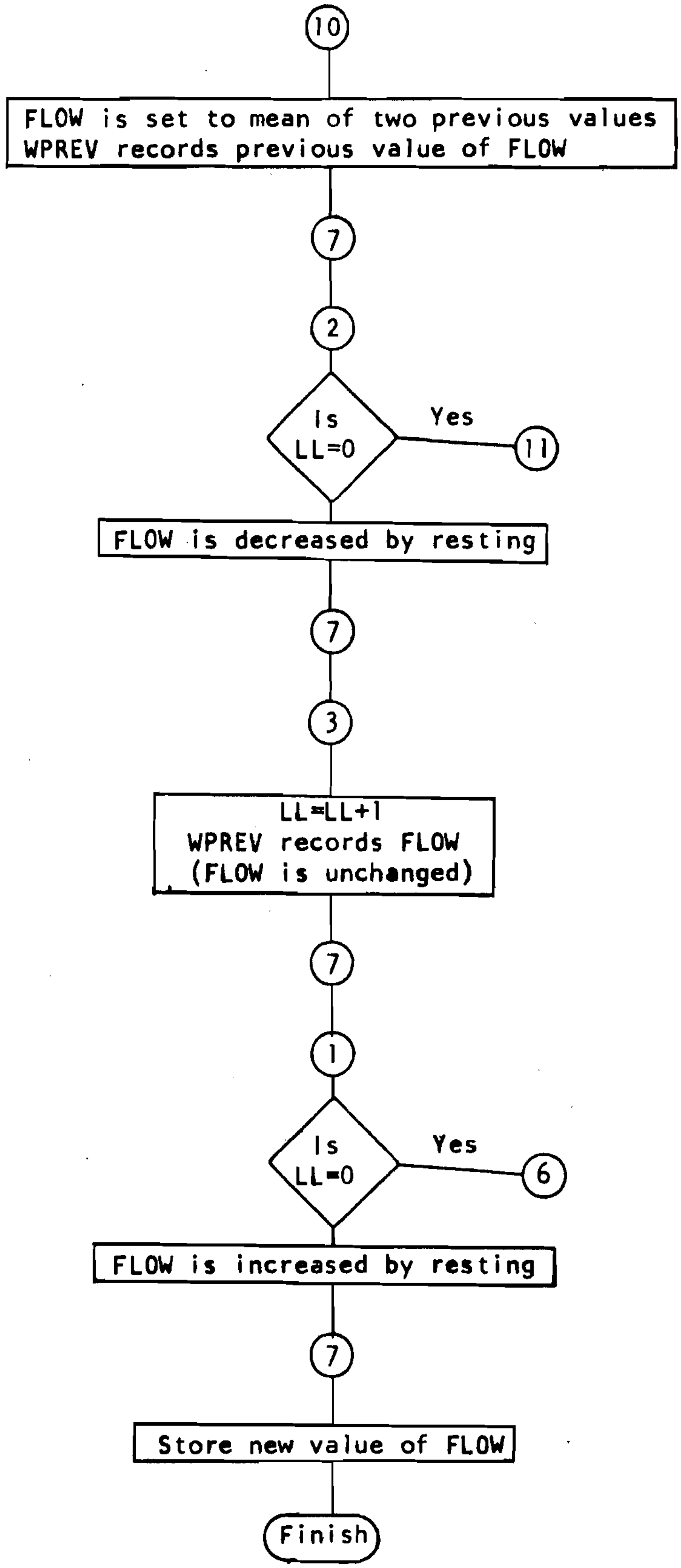


FIGURE 9 - FLOW CHART FOR SUBROUTINE SEARCH (CONTINUED)

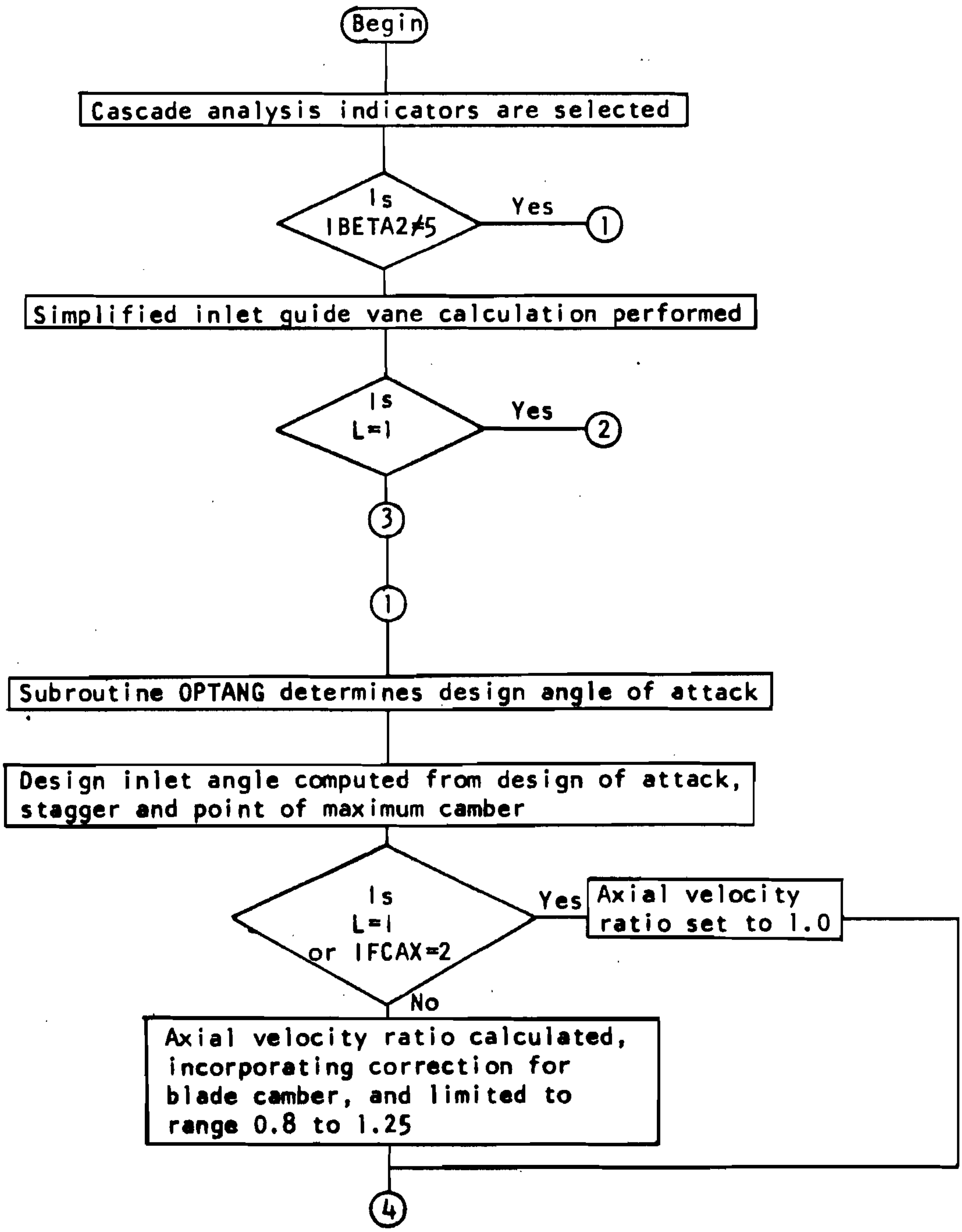


FIGURE 10 - FLOW CHART FOR SUBROUTINE CASCDE

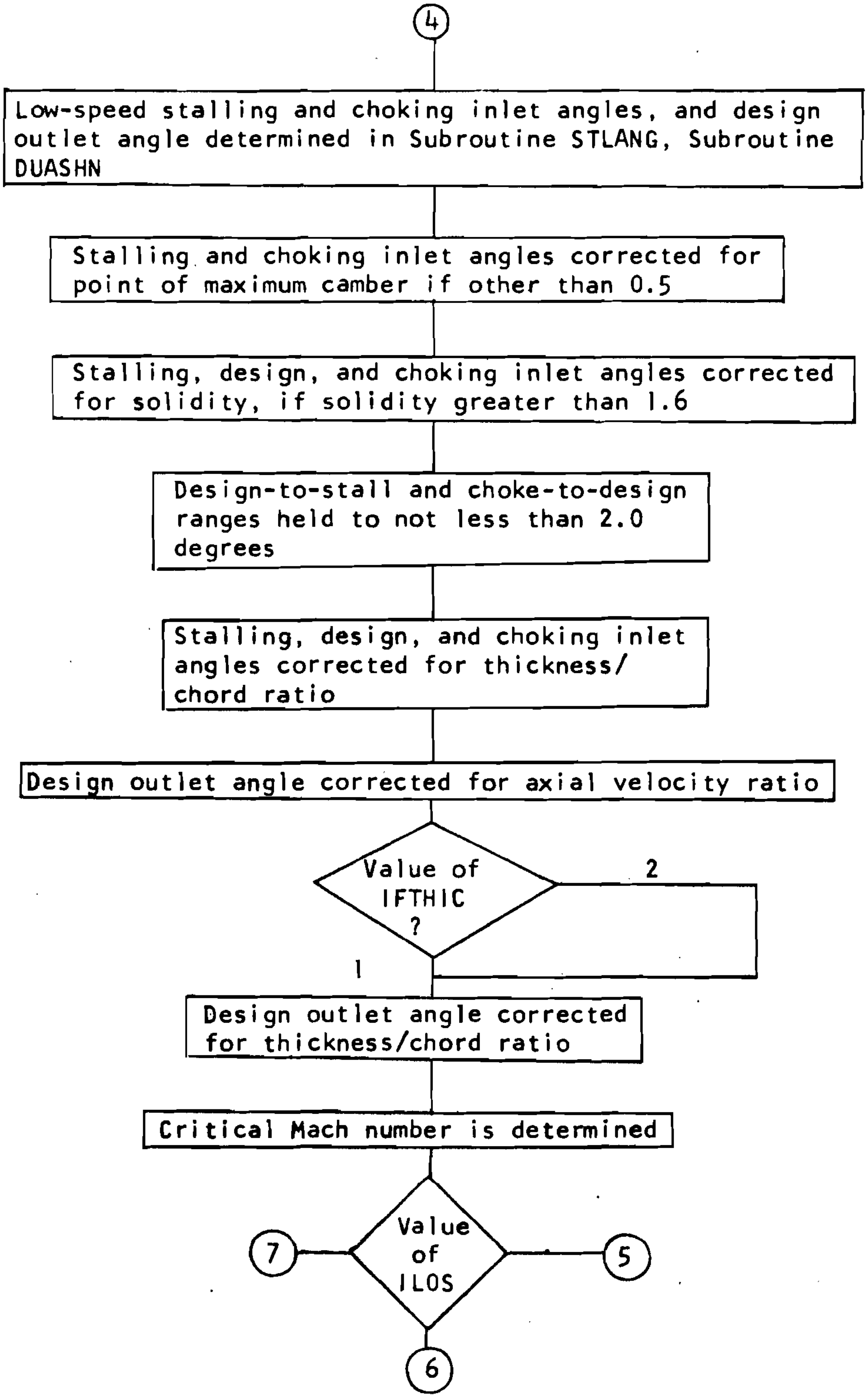


FIGURE 10 - FLOW CHART FOR SUBROUTINE CASCDE (CONTINUED)

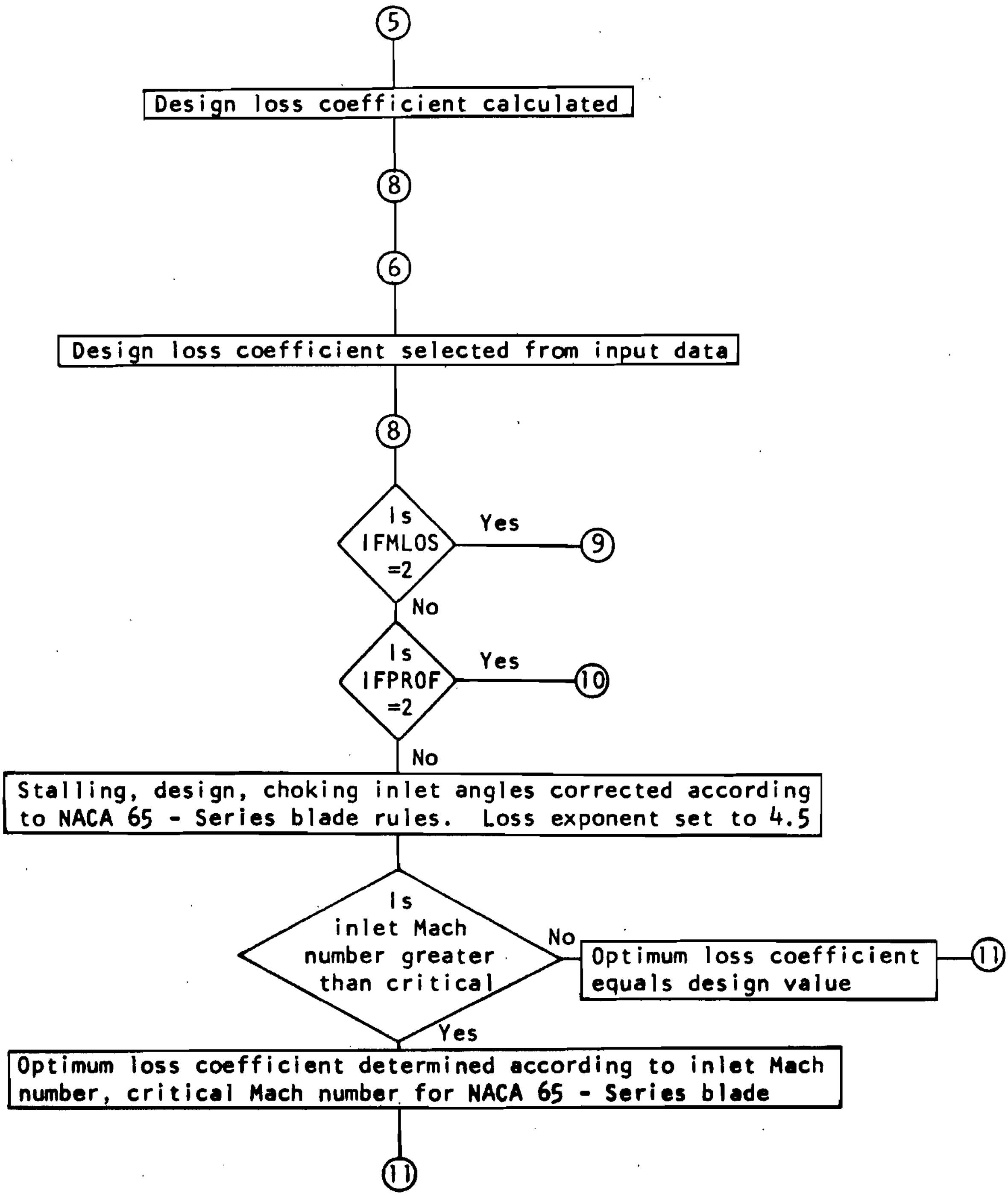


FIGURE 10 - FLOW CHART FOR SUBROUTINE CASCDE (CONTINUED)

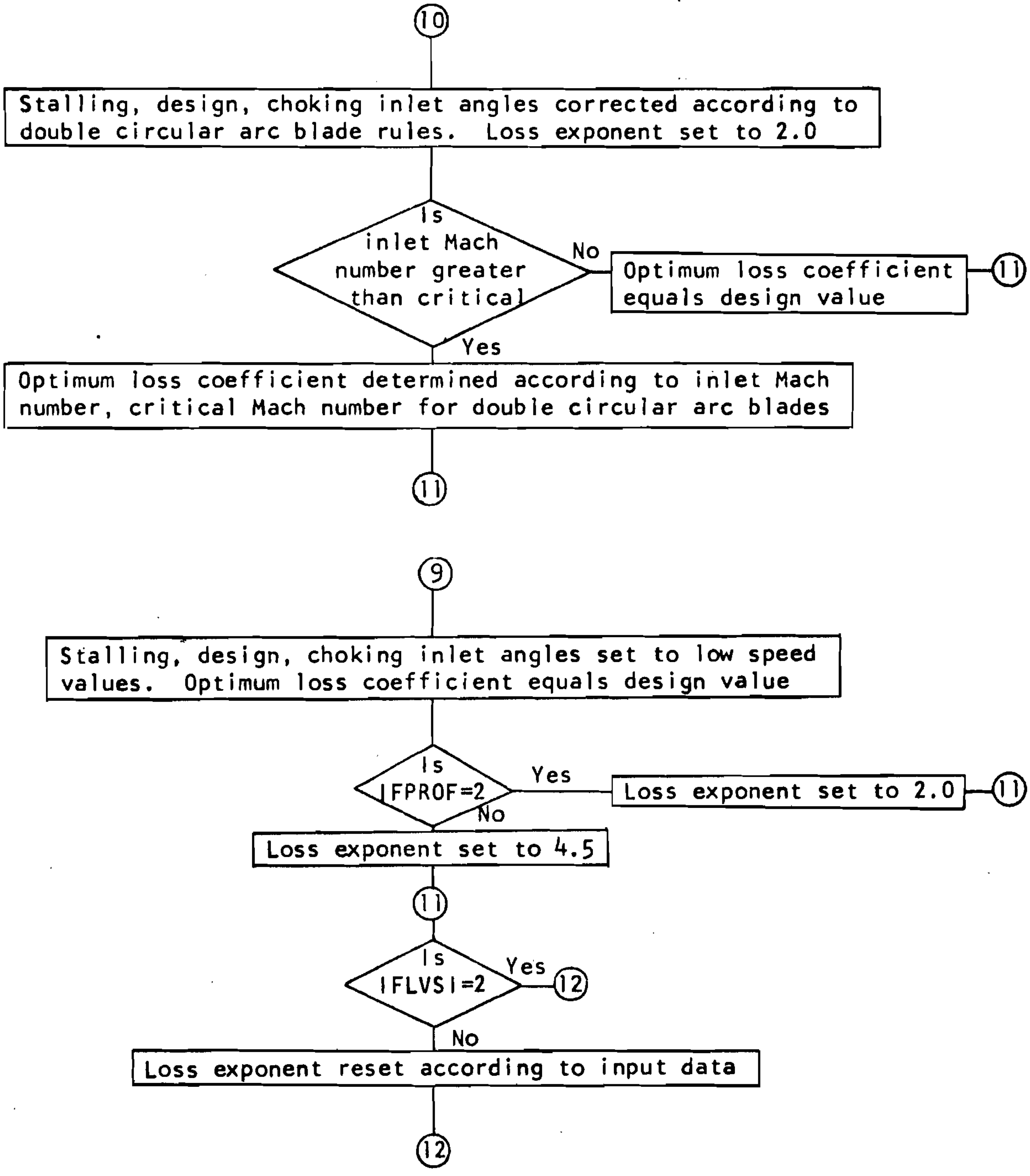


FIGURE 10 - FLOW CHART FOR SUBROUTINE CASCDE (CONTINUED)

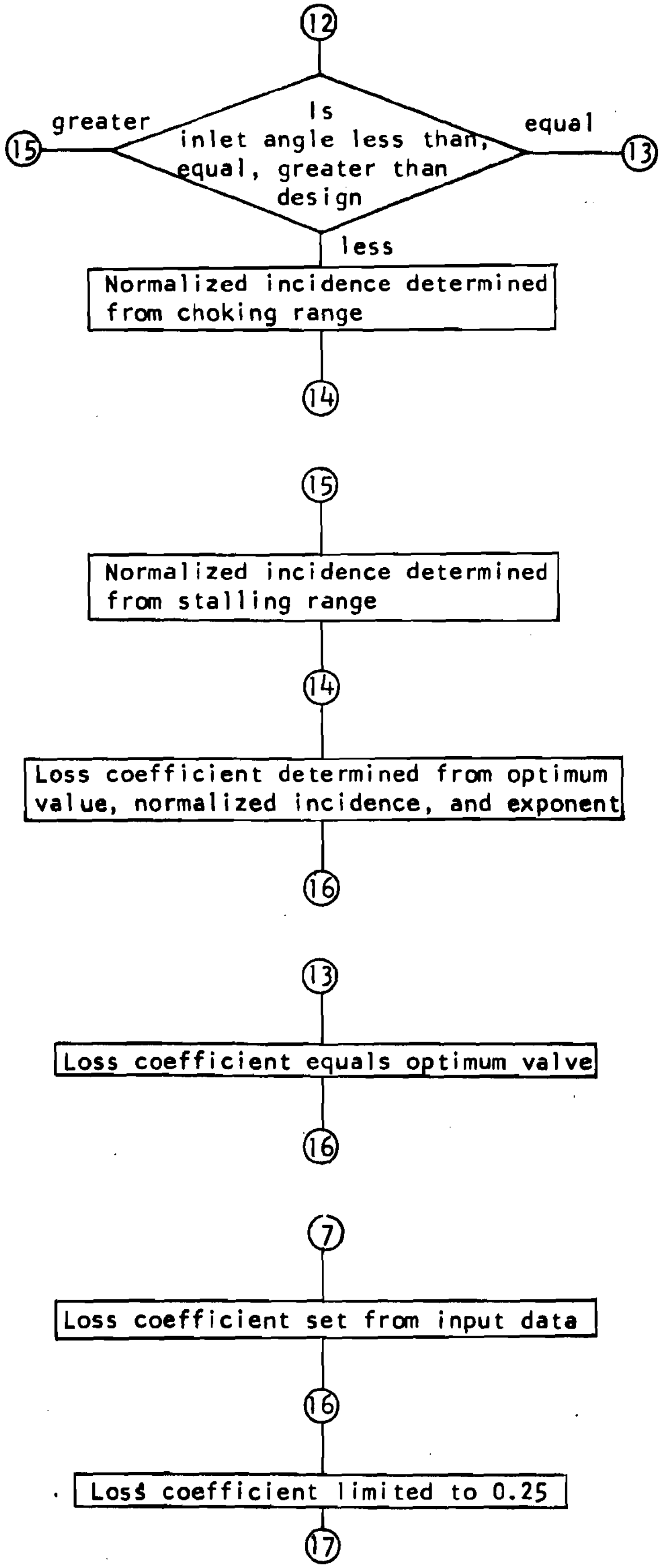


FIGURE 10 - FLOW CHART FOR SUBROUTINE CASCDE (CONTINUED)

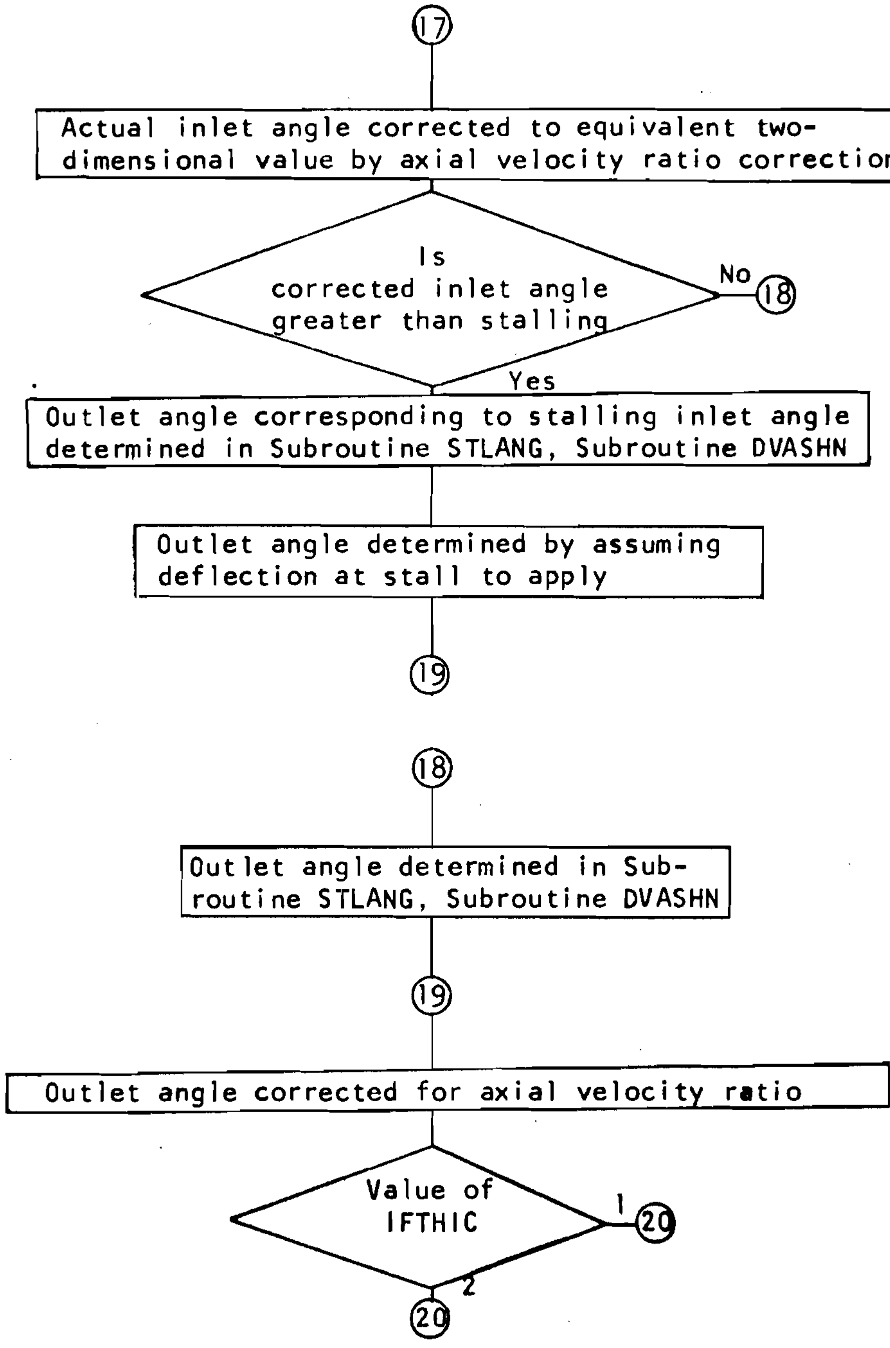


FIGURE 10 - FLOW CHART FOR SUBROUTINE CASCDE (CONTINUED)

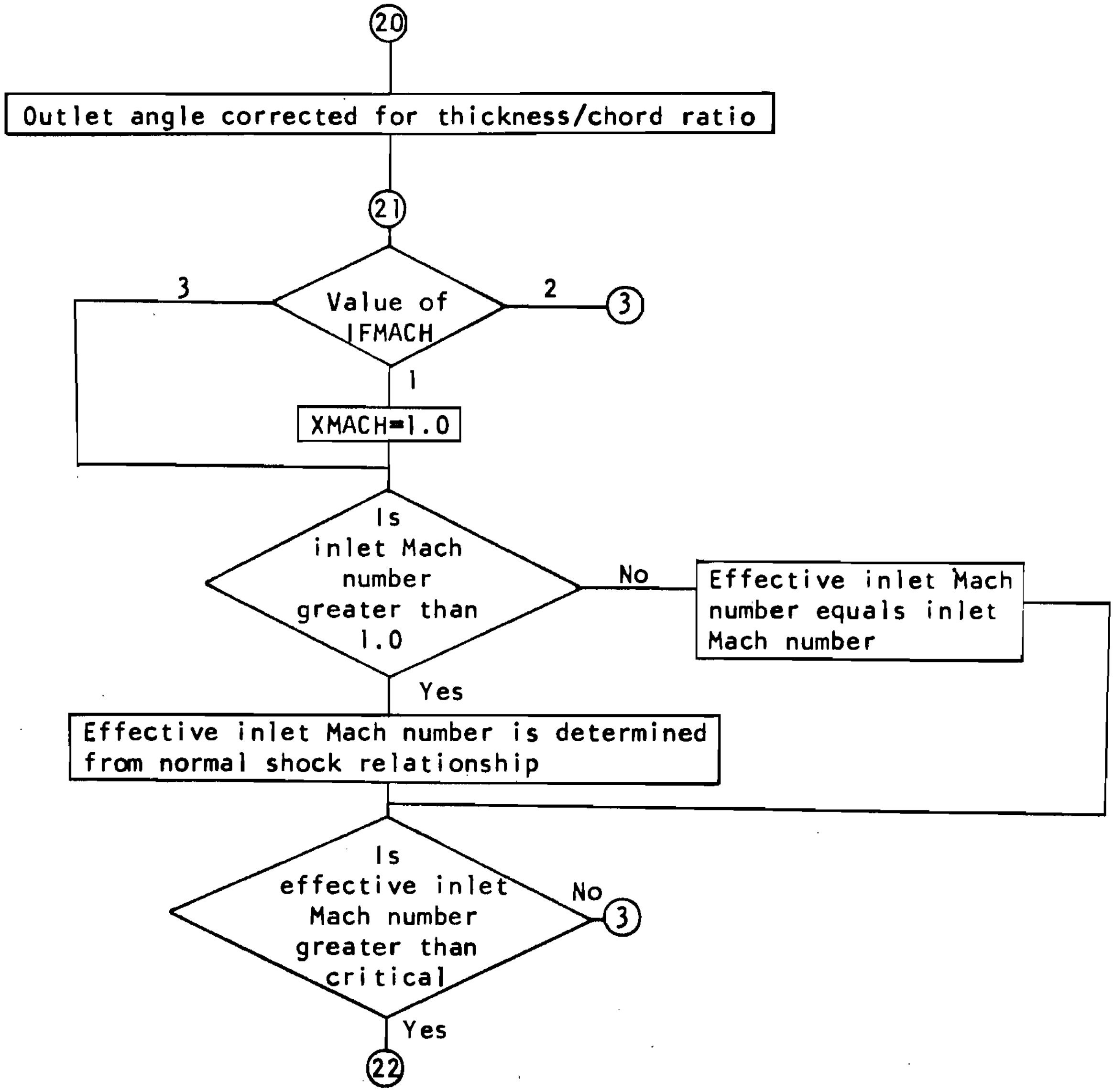


FIGURE 10 - FLOW CHART FOR SUBROUTINE CASCDE (CONTINUED)

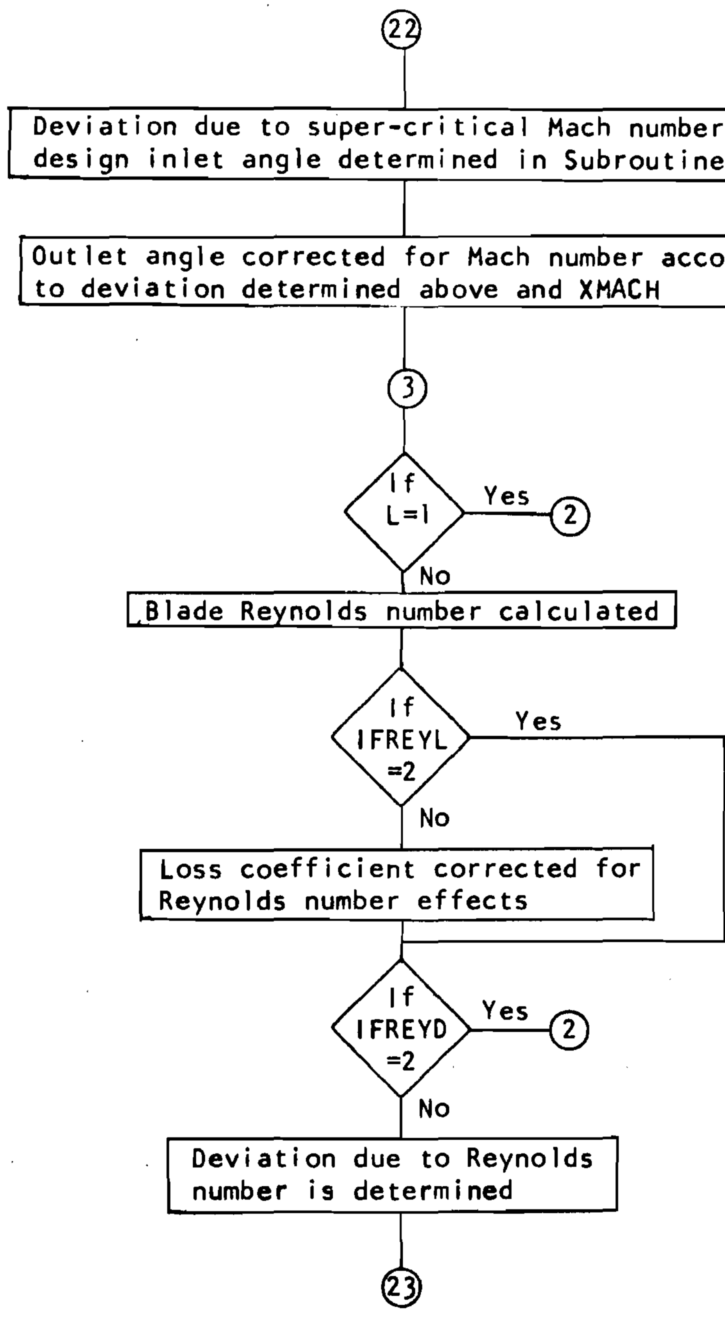


FIGURE 10 - FLOW CHART FOR SUBROUTINE CASCDE (CONTINUED)

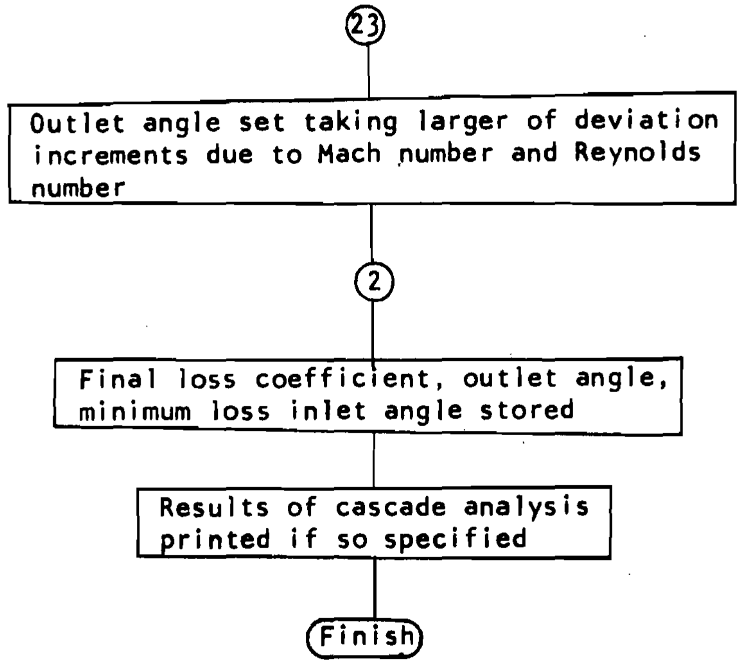


FIGURE 10 - FLOW CHART FOR SUBROUTINE CASCDE (CONTINUED)

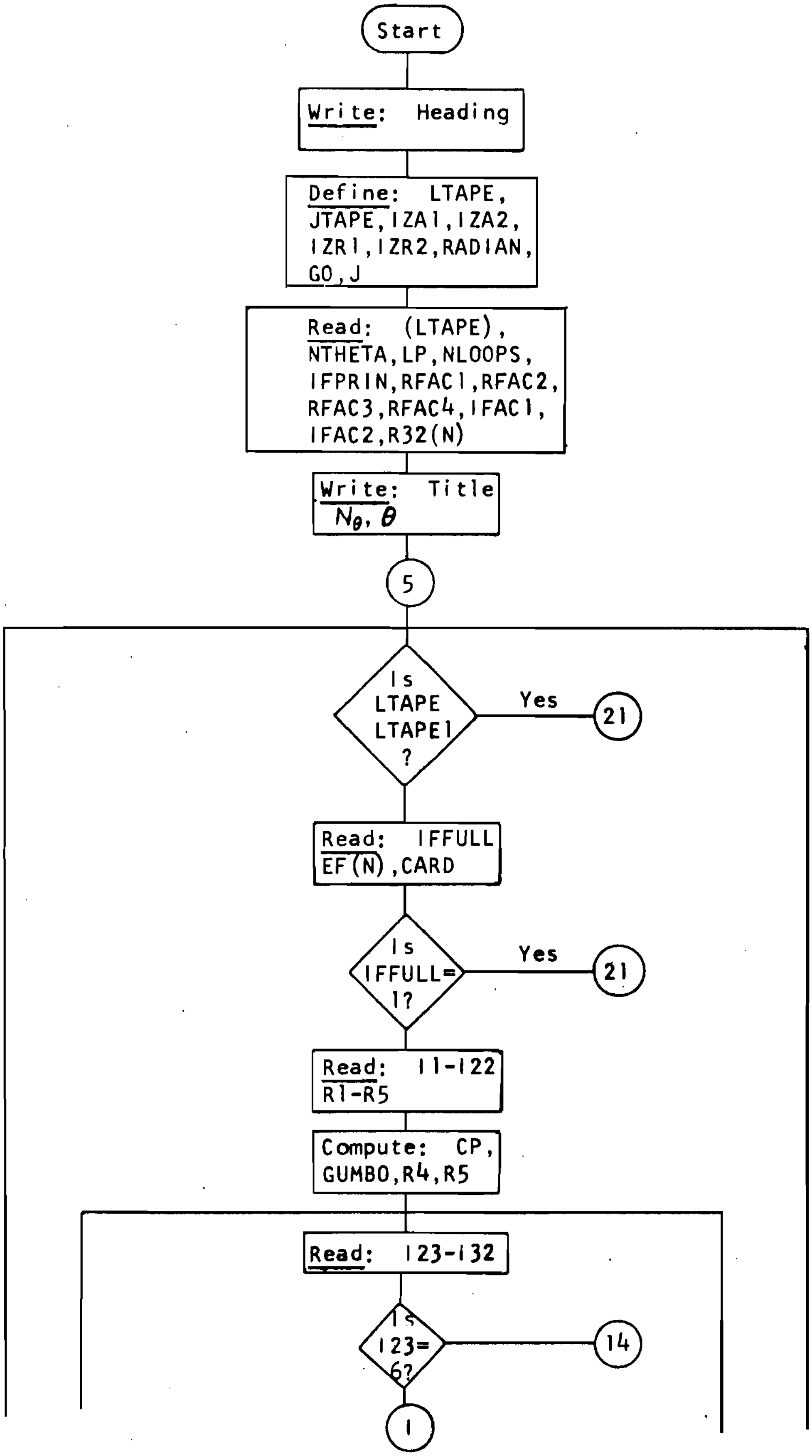


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA

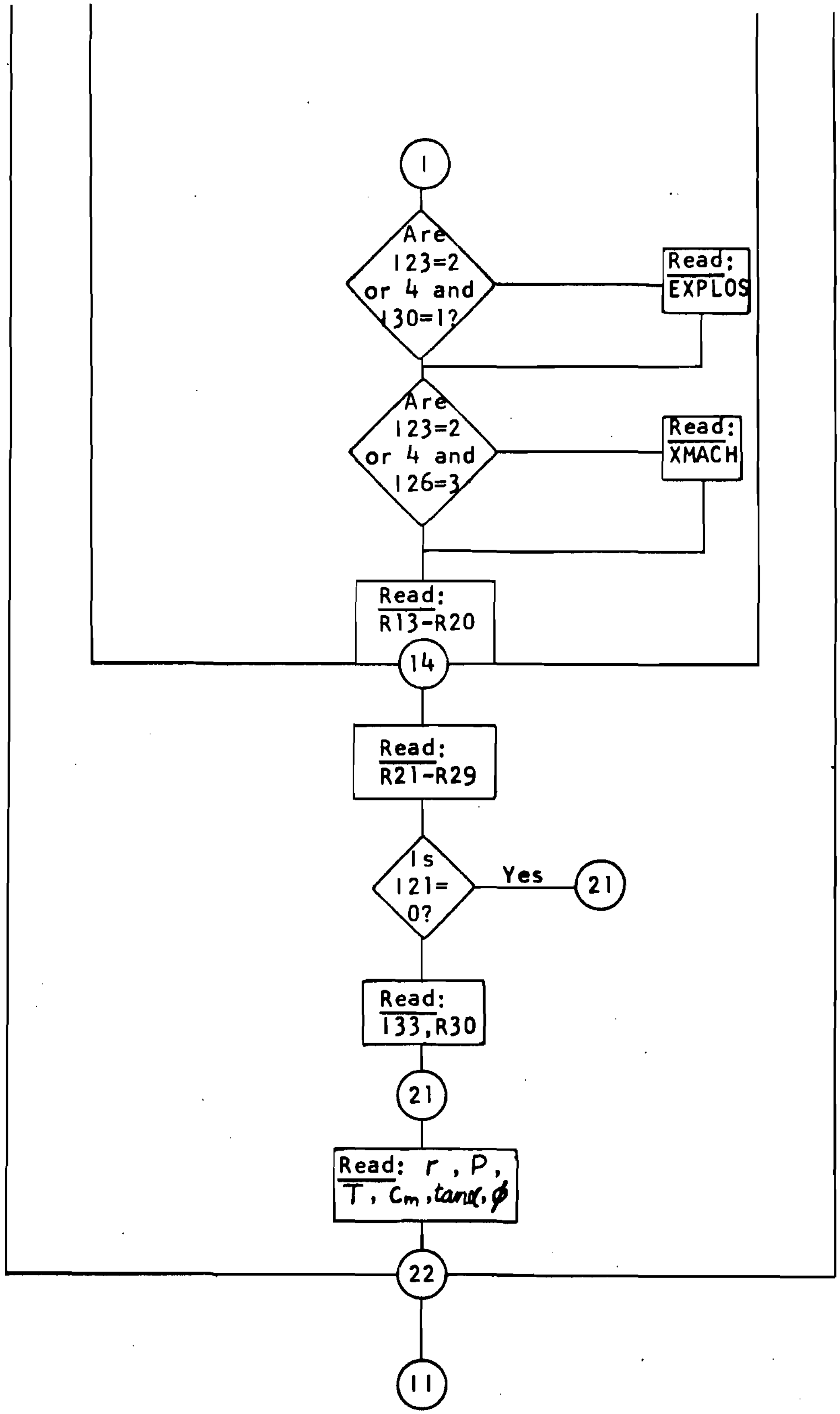


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA (CONTINUED)

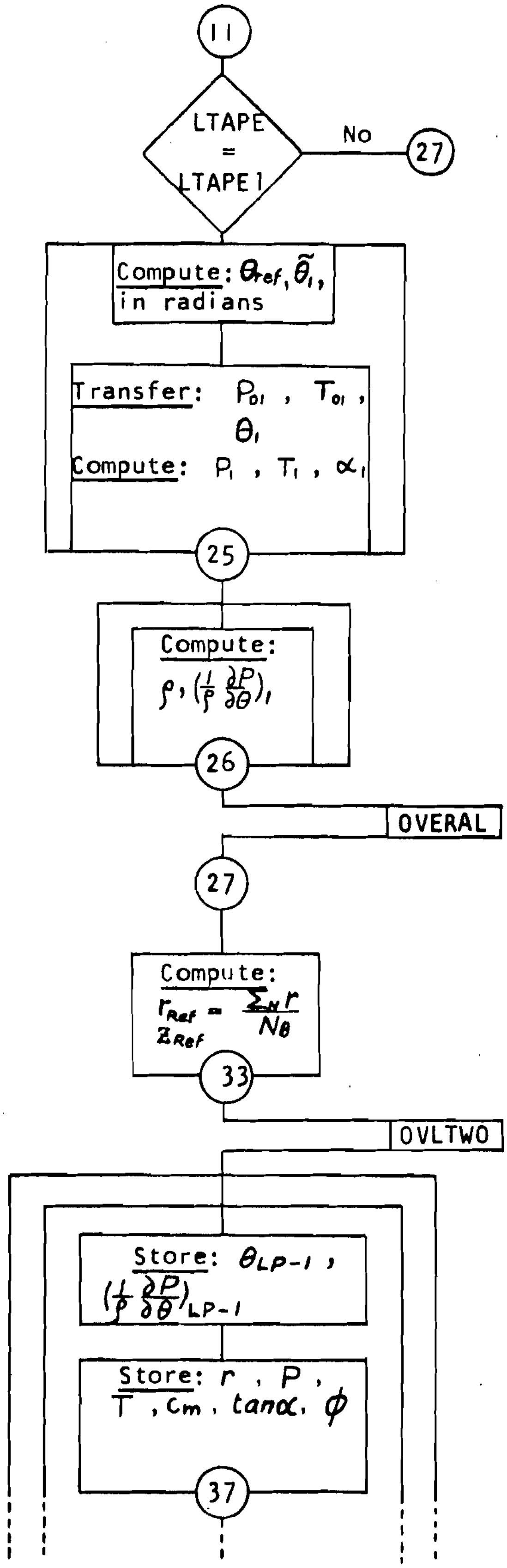


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA (CONTINUED)

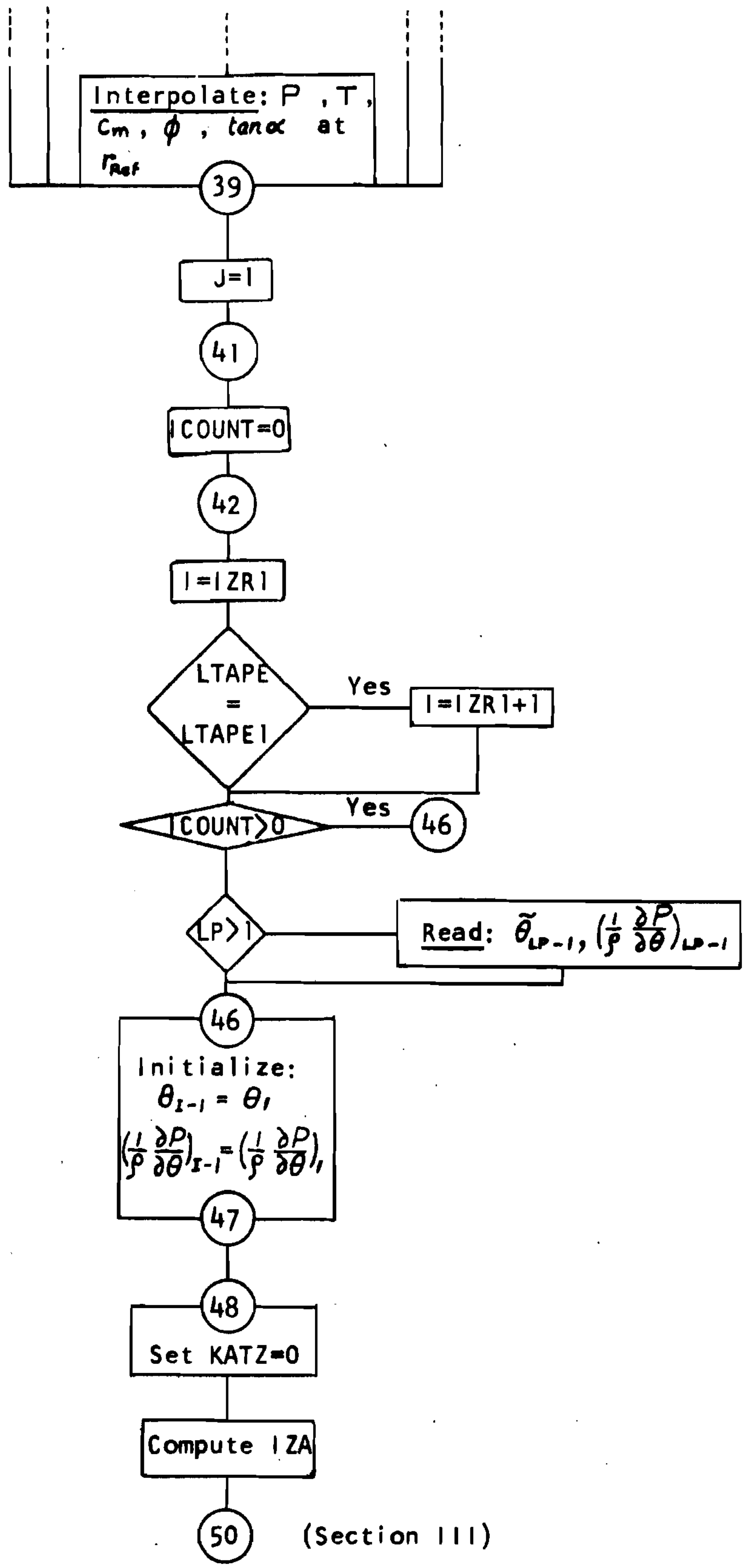


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA (CONTINUED)

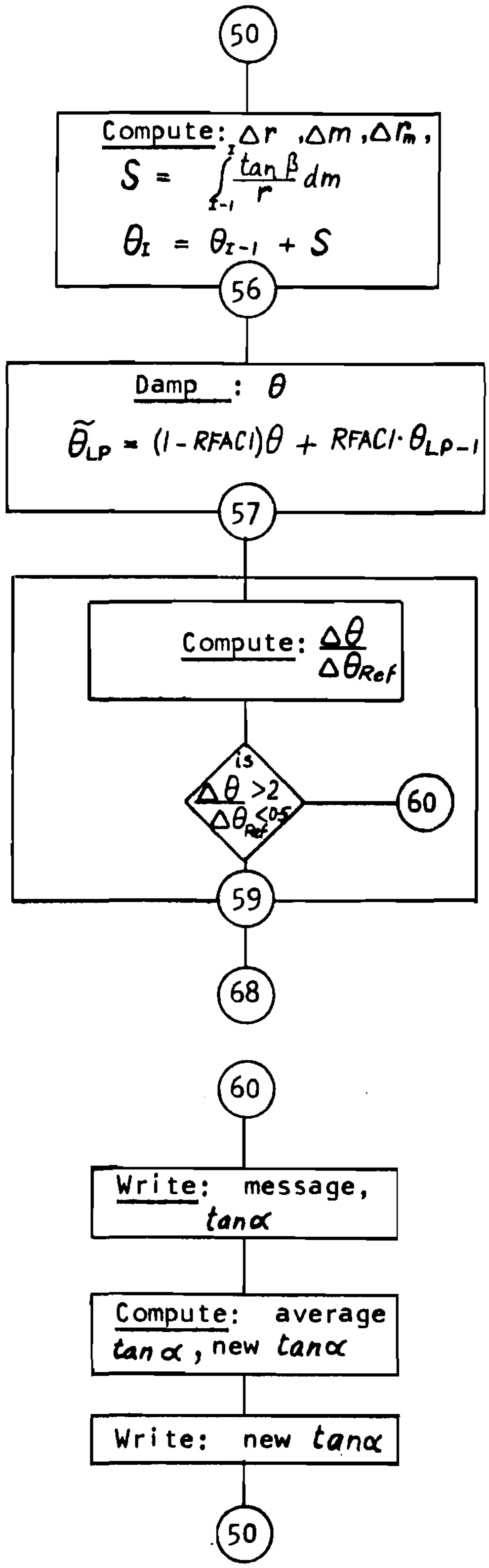


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA (CONTINUED)

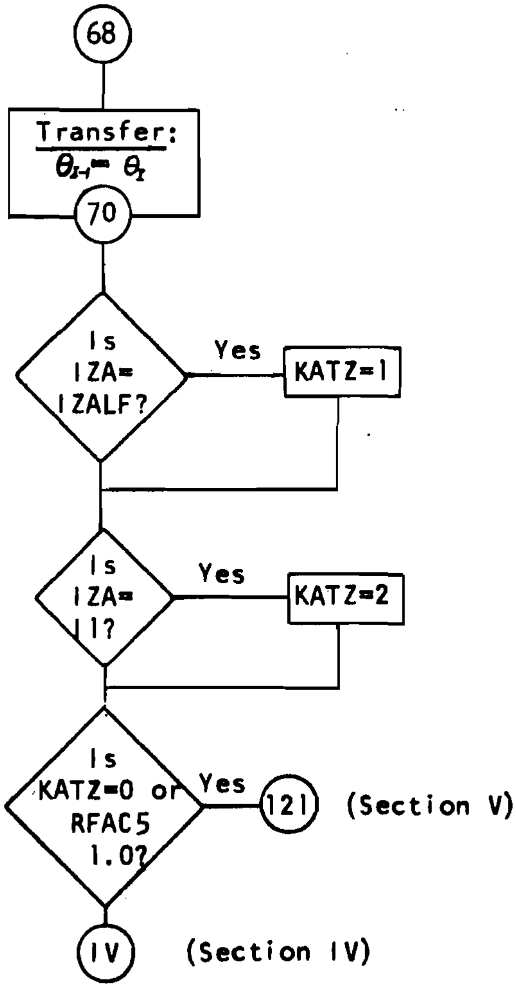


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA (CONTINUED)

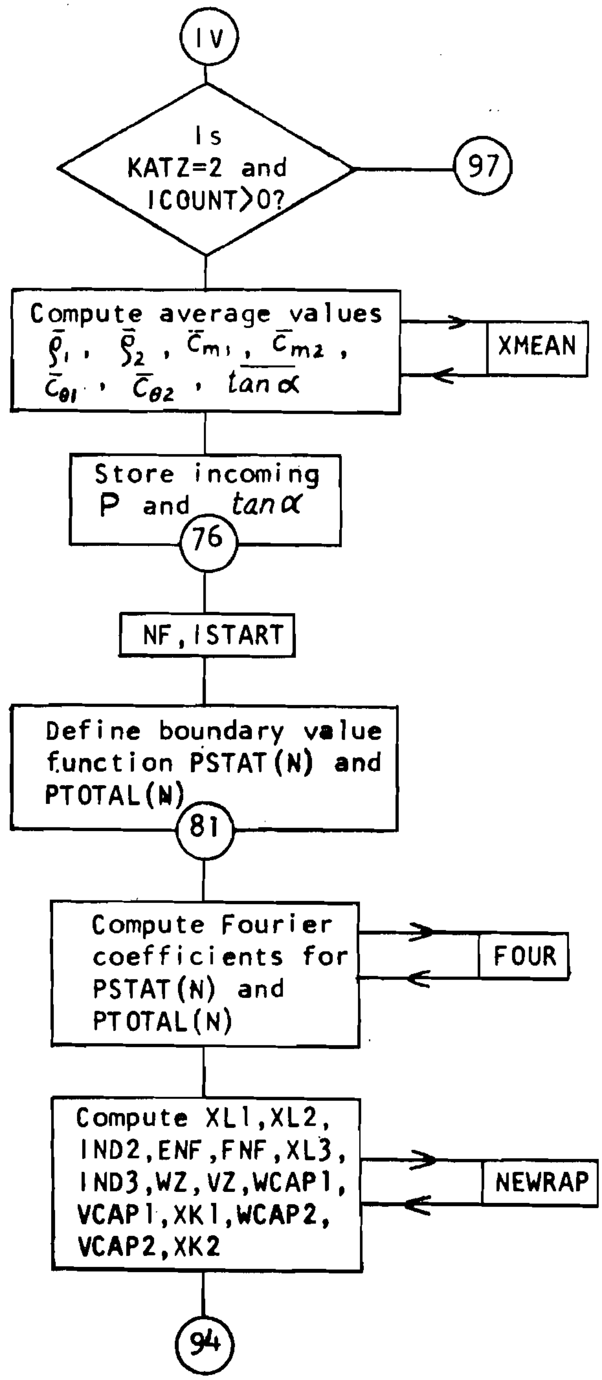


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA (CONTINUED)

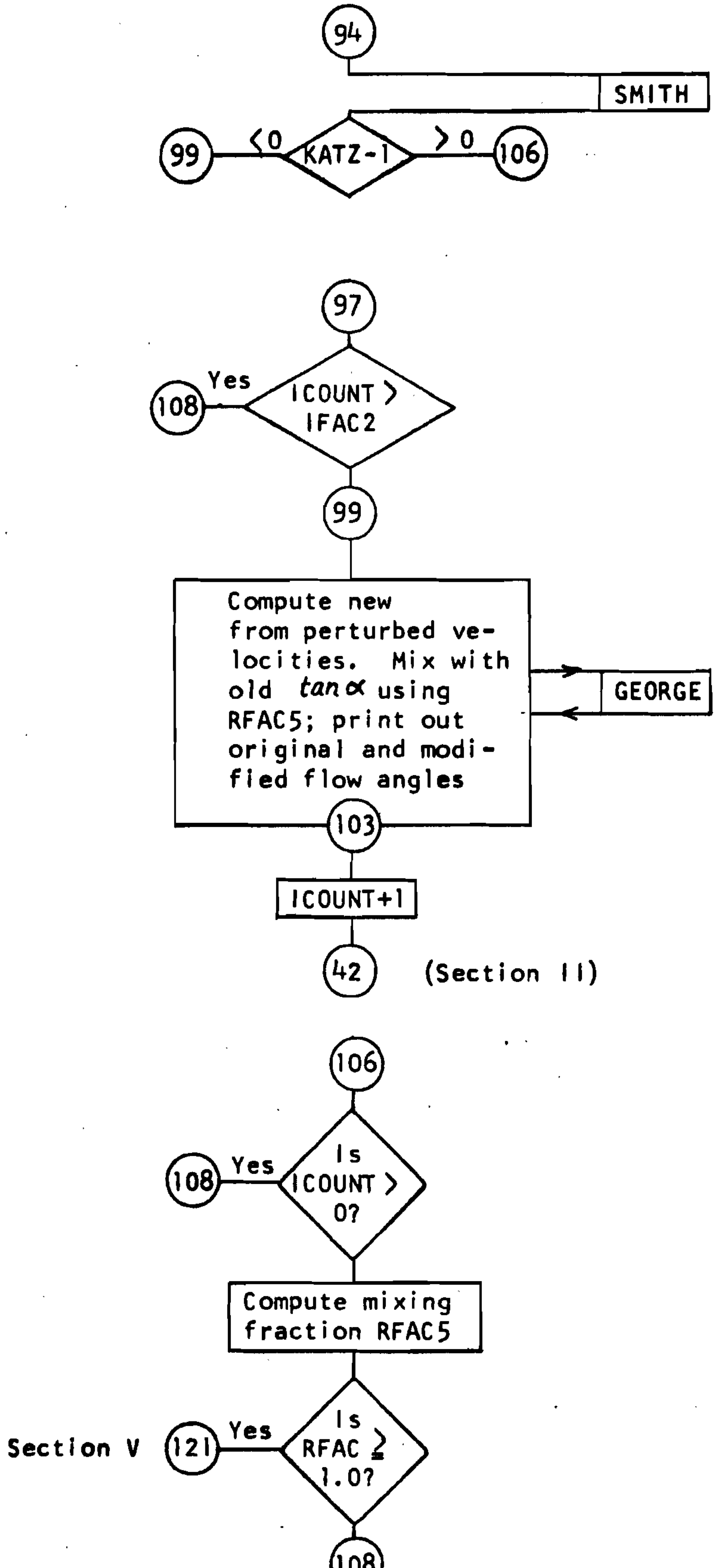


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA (CONTINUED)

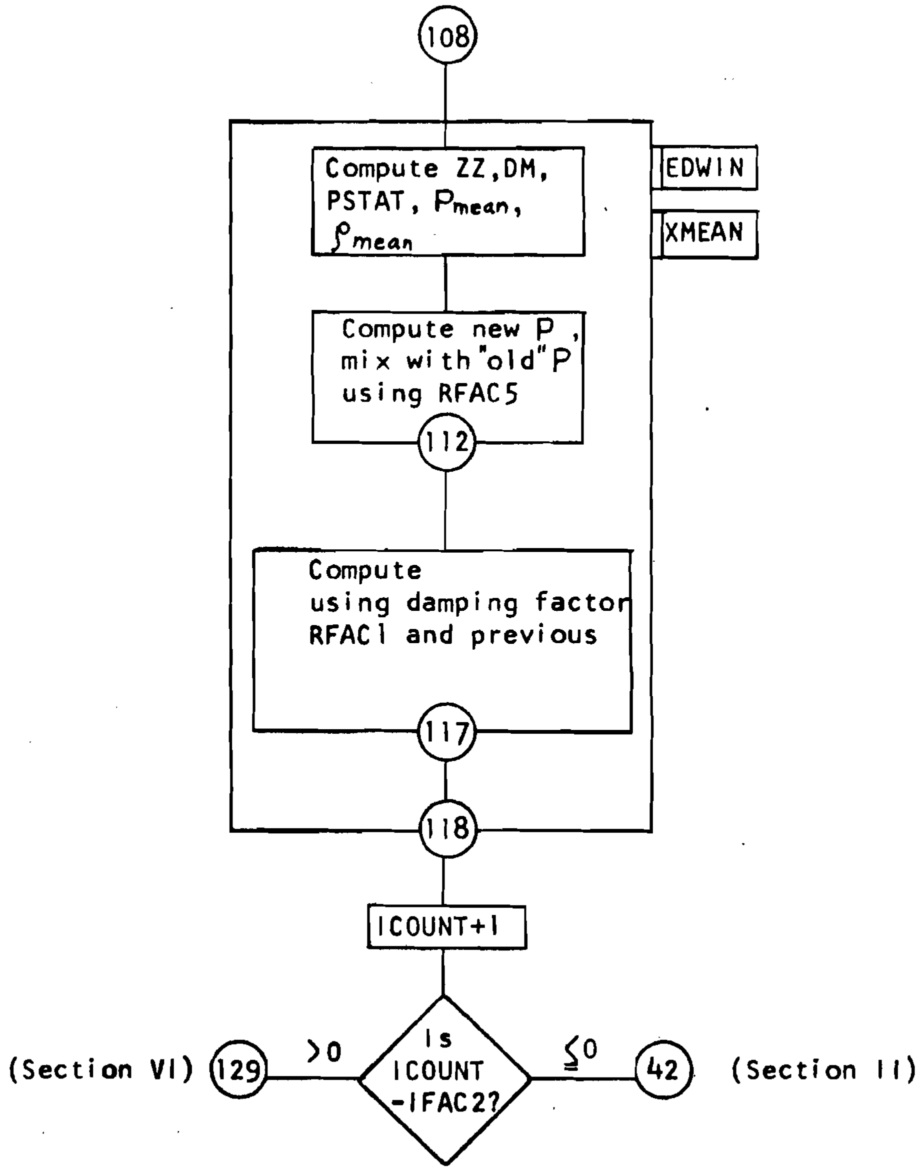


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA (CONTINUED)

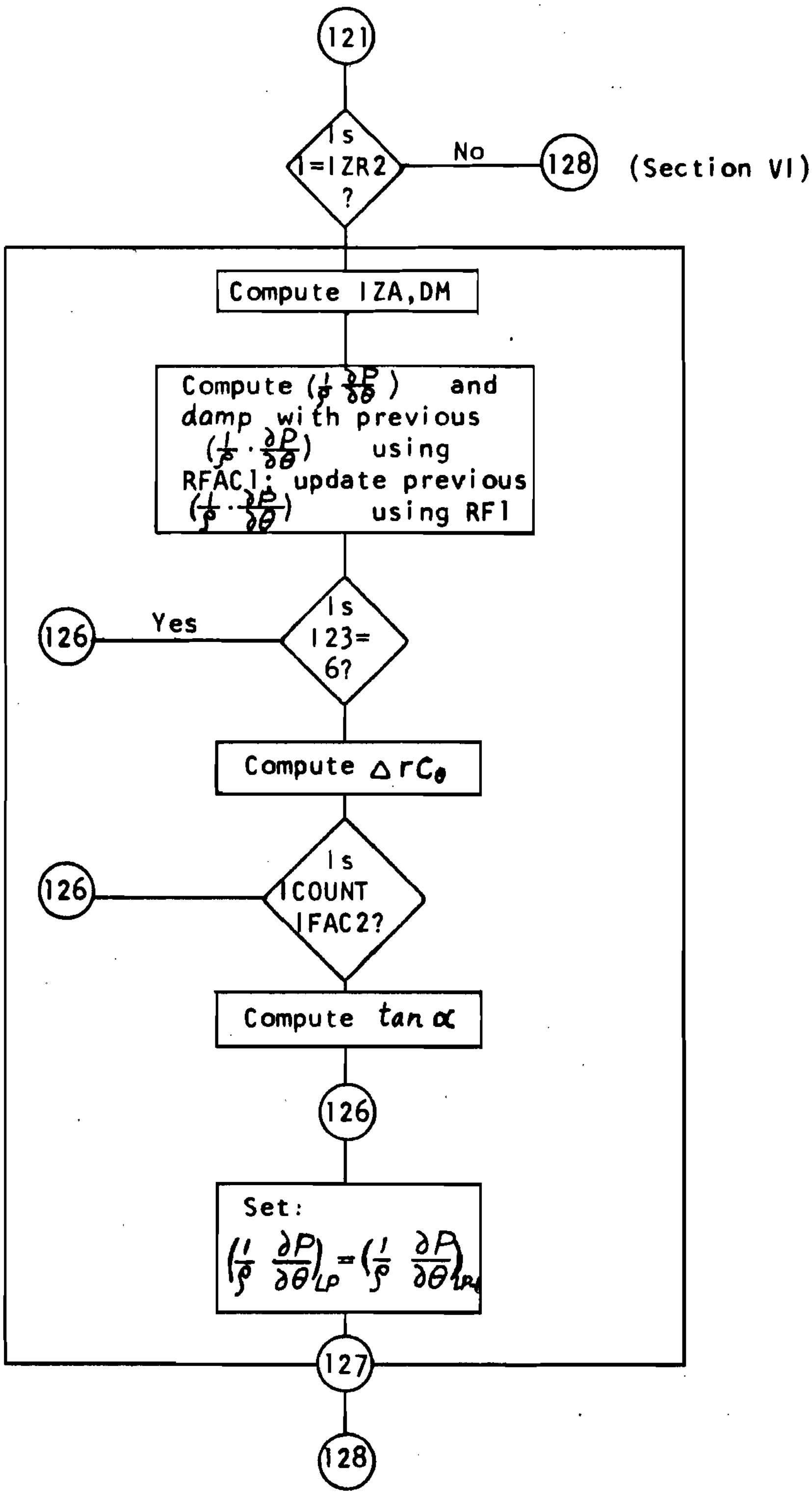


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA (CONTINUED)

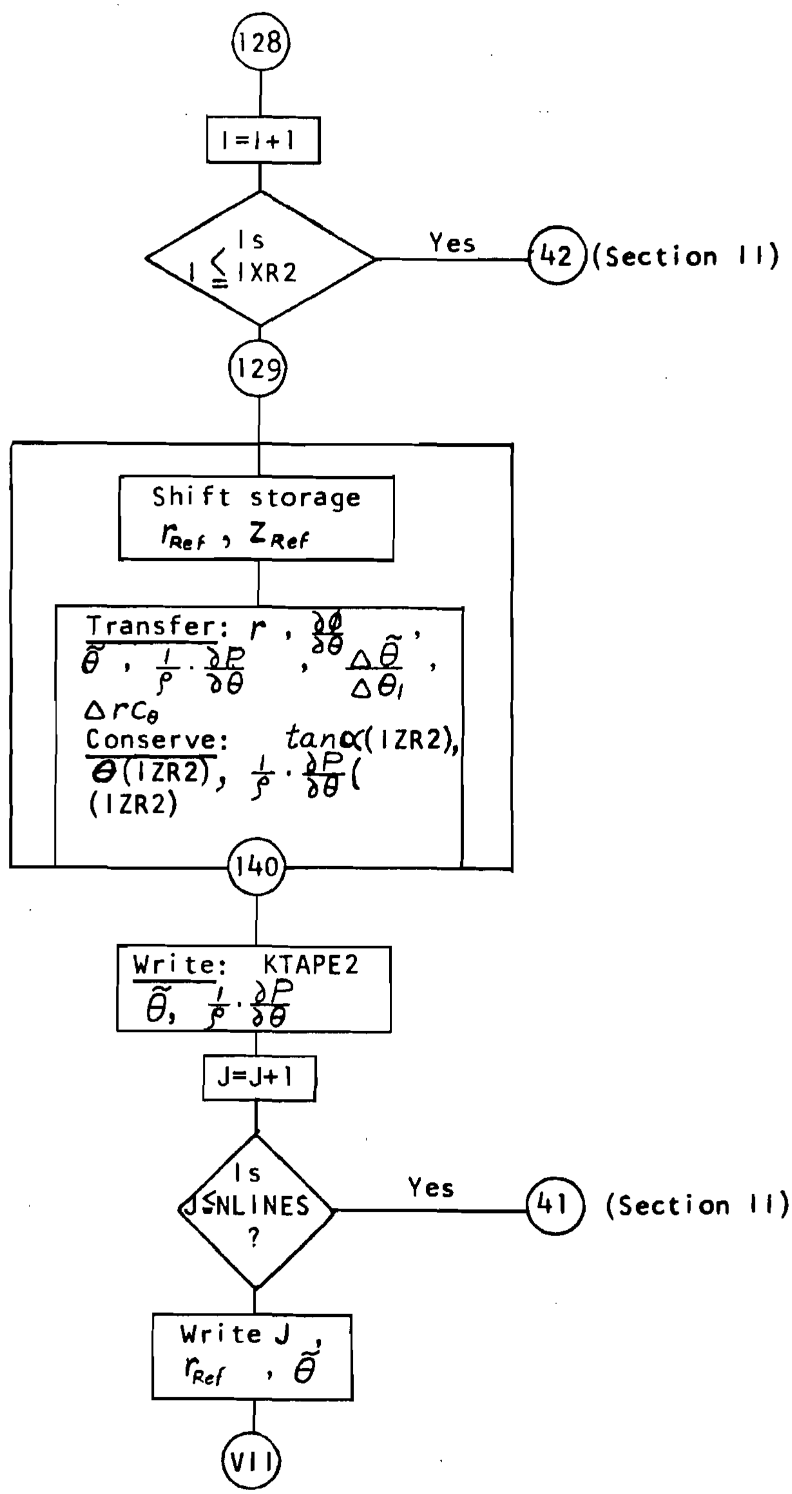


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA (CONTINUED)

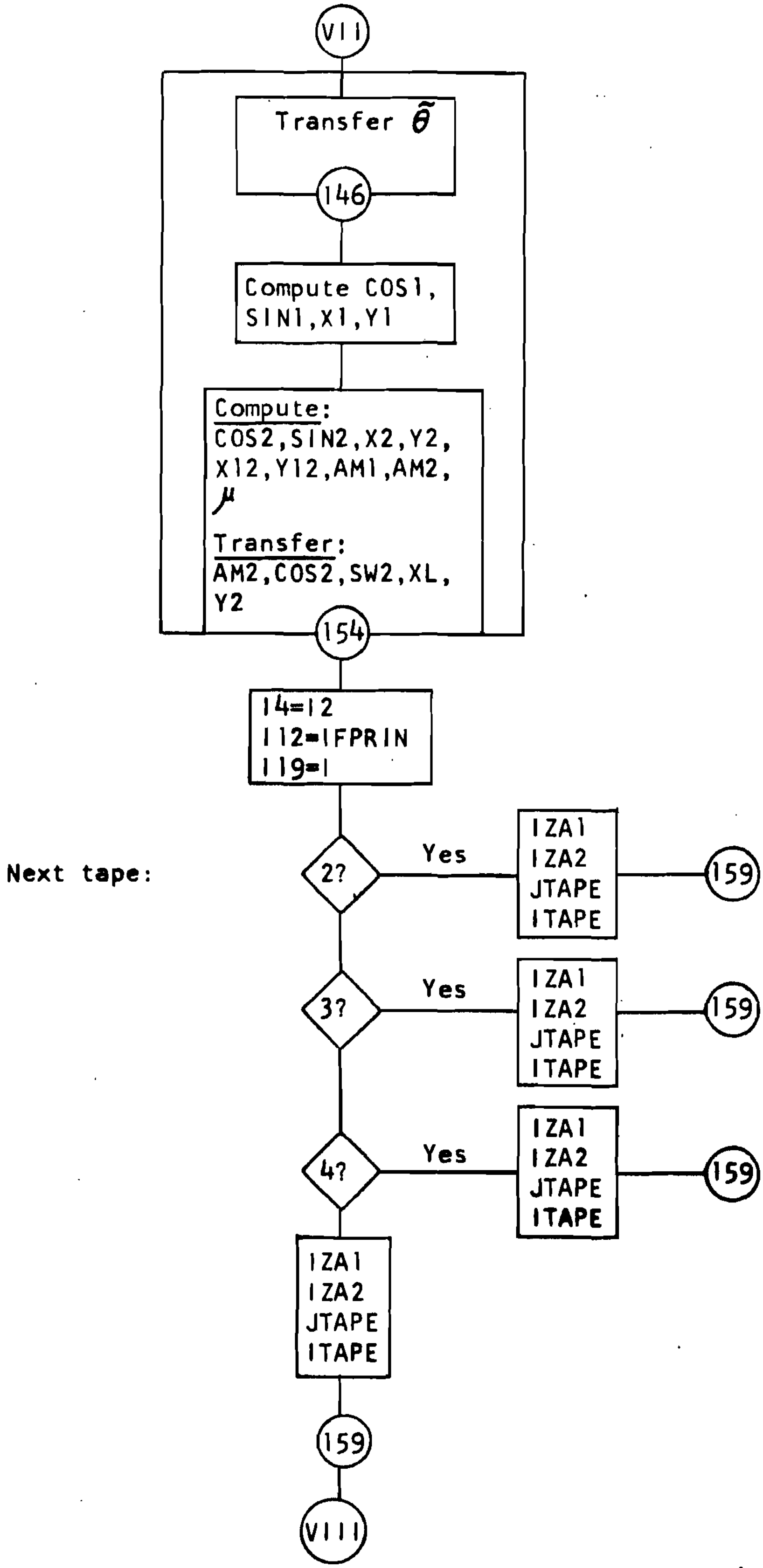


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA (CONTINUED)

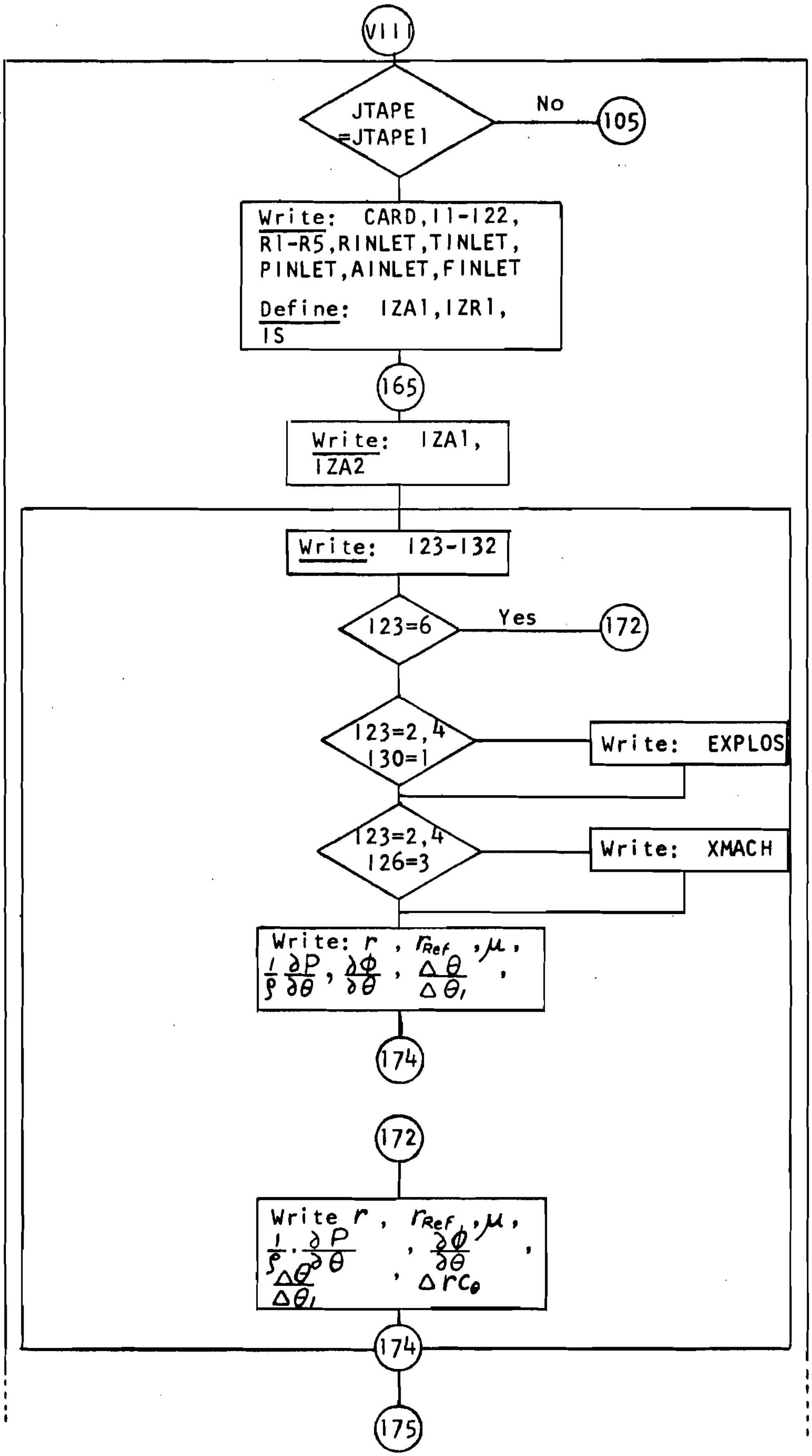


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA (CONTINUED)

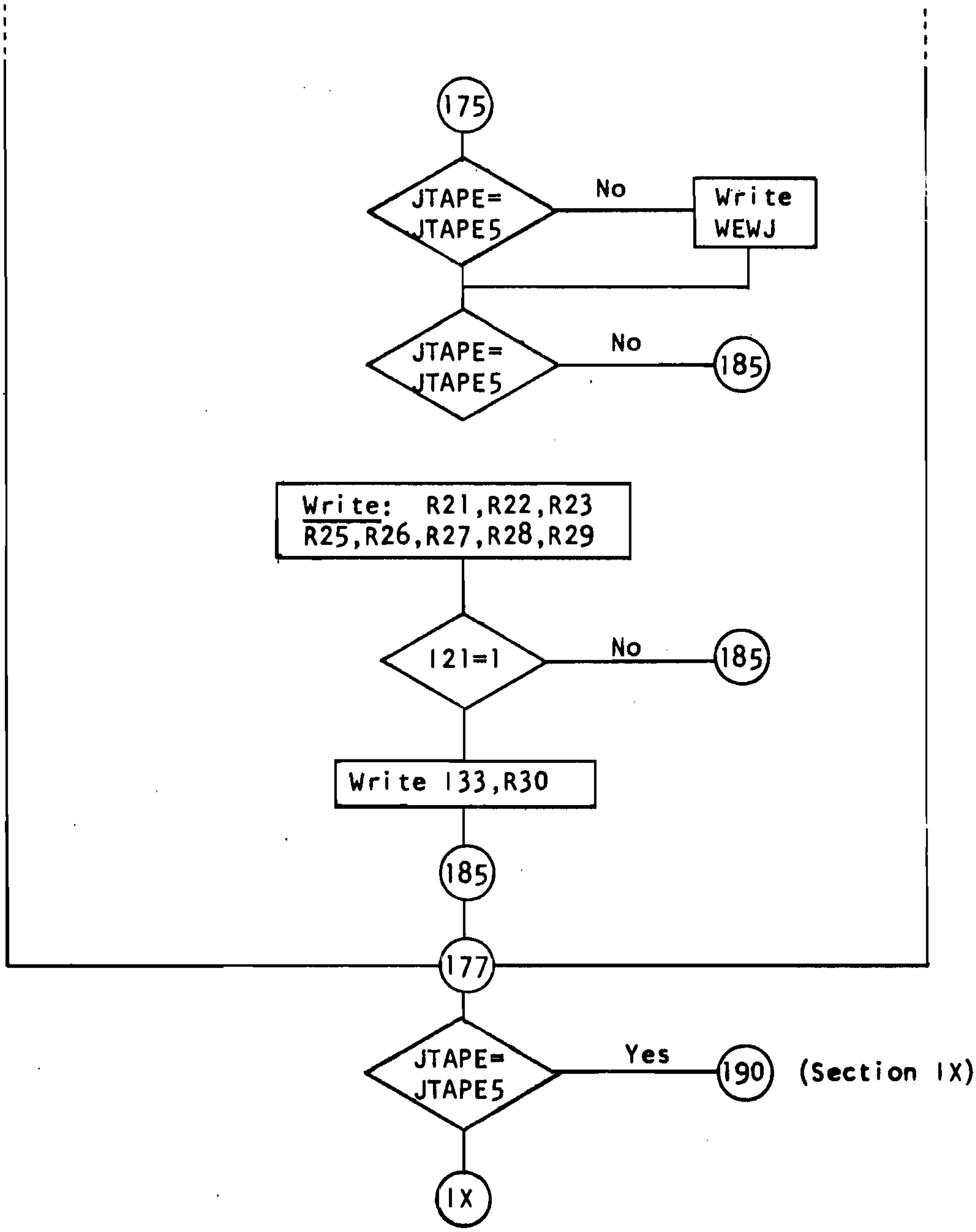


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA (CONTINUED)

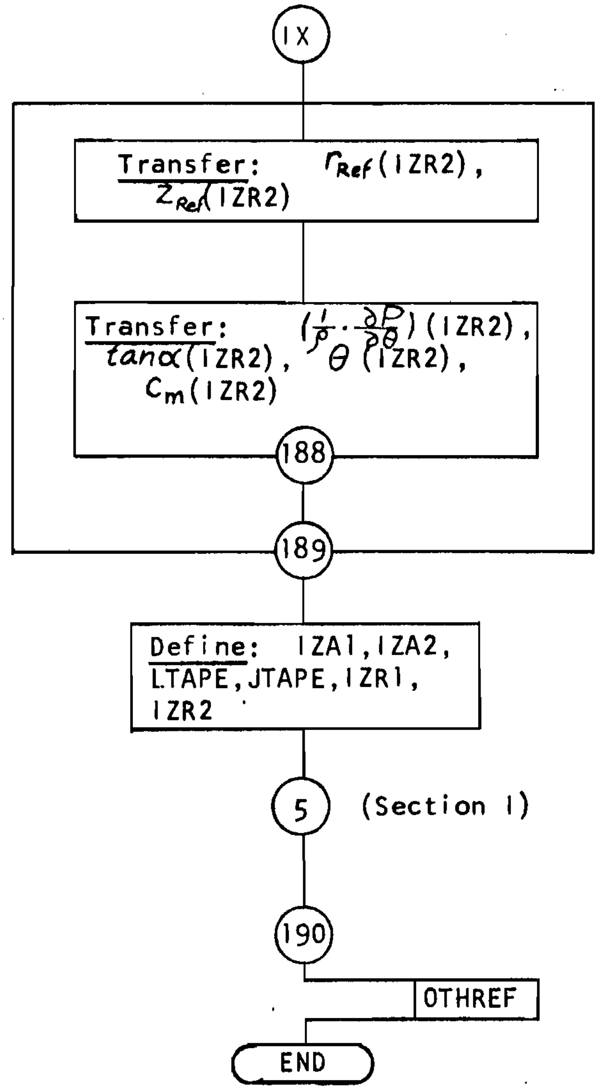


FIGURE 11 - FLOW CHART FOR SUBROUTINE THETA (CONTINUED)

TABLES

TABLE I - UNIT NAMES AND APPLICATIONS

Unit Name	Mode	Maximum Record Length (Words)	Use, Recommended Unit
ITAPE	BCD		Standard Input, Card Reader
NTAPE	BCD		Standard Output, Printer
JTAPE1	BCD	7xNLINES, 105	Data for first part of Program (DISTORT), Recommend Magnetic Tape
JTAPE2	BCD	7xNLINES, or 8xNBETA, 120	
JTAPE3	BCD		
JTAPE4	BCD		
JTAPE5	BCD	7xNLINES, or 3+3xNX, 105	
LTAPE1	BCD	8xNBETA, or 3xNX, or 6xNLINESxZ1, 540	Data for Second Part of Program (THETA) Recommend Disc
LTAPE2	BCD	6xNLINESxZ2, 540	
LTAPE3	BCD	6xNLINESxZ3, 540	
LTAPE4	BCD	6xNLINESxZ4, 540	
LTAPE5	BCD	6xNLINESxZ5, 540	
KTAPE1	BIN	2xNTHETAxZMAX, 156	Data written on even number cycles, read on odd by "THETA", Recommend Tape
KTAPE2	BIN	2nXTHETAxZMAX, 156	Data written on even number cycles, read on even by "THETA", Recommend Disc

Notes: NLINES is number of streamlines
NBETA is number of blade data radii
NX is total number of stations
Zx is number of stations on unit LTAPEx
NTHETA is number of sectors
ZMAX is largest value of Zx

TABLE II - RECOMMENDED UNIT NUMBERS

Unit Name	Fortran Logical Unit Number	Symbolic Unit Number	Notes
ITAPE	5		No action required
NTAPE	6		No action required
JTAPE1	1	A(1)	Set up mag tape
JTAPE2	2	A(2)	Set up mag tape if required
JTAPE3	3	A(3)	Set up mag tape if required
JTAPE4	4	A(4)	Set up mag tape if required
JTAPE5	7	A(7)	Set up mag tape
LTAPE1	8	A(8)	Allow to reside on disk
LTAPE2	9	A(9)	Allow to reside on disk
LTAPE3	10	B(0)	Allow to reside on disk
LTAPE4	11	B(1)	Allow to reside on disk
LTAPE5	12	B(2)	Allow to reside on disk
KTAPE1	13	B(3)	Set up mag tape
KTAPE2	14	B(4)	Allow to reside on disk

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Northern Research and Engineering Corporation (NREC)	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED
	2b. GROUP ———

3. REPORT TITLE
PROPULSION SYSTEM FLOW STABILITY PROGRAM (DYNAMIC)
PART VIII. MODEL FOR PREDICTING COMPRESSOR PERFORMANCE WITH COMBINED CIRCUMFERENTIAL AND RADIAL DISTORTION - COMPUTER PROGRAM

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Phase I Final Technical Report

5. AUTHOR(S) (First name, middle initial, last name)
R. A. Novak and R. M. Hearsey, et.al. (NREC)

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1968	i - xii plus 304	———

8a. CONTRACT OR GRANT NO. F33615-67-C-1848	9a. ORIGINATOR'S REPORT NUMBER(S) EDR5908, Part VIII
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	AFAPL-TR-68-142, Part VIII

10. DISTRIBUTION STATEMENT This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of The Air Force Aero Propulsion Laboratory (APTA), Air Force Systems Command, Wright-Patterson Air Force Base, Ohio.

11. SUPPLEMENTARY NOTES ———	12. SPONSORING MILITARY ACTIVITY Air Force Aero Propulsion Laboratory Air Force Systems Command Wright-Patterson Air Force Base, Ohio 45433

13. ABSTRACT

This part describes in detail a computer program designed to enable the performance of axial flow compressors operating under asymmetric inlet conditions to be predicted. The underlying theory is given in Part VII; a comparison between results obtained and experimental data is made in Part IX.

The report is divided into a number of sections, the principal ones being those giving a Functional Description of the program, and an Operational Description of the program. The Functional Description is intended to enable the program to be used. The Operational Description of the program is intended to enable the methods used in the program to be understood in detail, hence allowing the program to be modified if desired.

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Asymmetric Inlet						
Axial Flow Compressor Performance						
Combined Circumferential and Radial Distortion						
Compressor Performance Computer Program						
Compressor Performance Prediction Model						
Propulsion System Flow Stability						

UNCLASSIFIED

Security Classification

Accession Number : AD0851903

Title : Propulsion System Flow Stability Program (Dynamic). Part VIII. Model for Predicting Compressor Performance with Combined Circumferential and Radial Distortion - Computer Program.

Descriptive Note : Final technical rept. 20 Jun 67-30 Sep 68 on Phase 1,

Corporate Author : NORTH AMERICAN ROCKWELL CORP LOS ANGELES CA LOS ANGELES DIV

Personal Author(s) : Novak, R A ; Hearsey, R M

Report Date : Dec 1968

Pagination or Media Count : 312

Abstract : The part describes in detail a computer program designed to enable the performance of axial flow compressors operating under asymmetric inlet conditions to be predicted. The underlying theory is given in Part VII; a comparison between results obtained and experimental data is made in Part IX. The report is divided into a number of sections, the principal ones being those giving a Functional Discription of the program, and an Operational Description of the program. The Functional Description is intended to enable the program to be used. The Operational Description of the program is intended to enable the methods used in the program to be understood in detail, hence allowing the program to be modified if desired. (Author)

Descriptors : *TURBOFAN ENGINES , *TURBOJET INLETS , FLOW SEPARATION , FLOW CHARTING , PREDICTIONS , COMPUTER PROGRAMS , ASYMMETRY , BODIES , ENGINE SURGE , STABILITY , AIRCRAFT ENGINES , GAS TURBINES

Subject Categories : Fluid Mechanics
Jet and Gas Turbine Engines