NASA TM-79312

NASA-TM-79312 19800005795

NASA Technical Memorandum 79312

MODIFICATION OF AXIAL COMPRESSOR STREAMLINE PROGRAM FOR ANALYSIS OF ENGINE TEST DATA

Jeffrey G. Williams
Lewis Research Center
Cleveland, Ohio

November 1979

MODIFICATION OF AXIAL COMPRESSOR STREAMLINE PROGRAM FOR ANALYSIS OF ENGINE TEST DATA

by Jeffrey G. Williams

E-268

SUMMARY

This report describes modifications of an existing ax al compressor streamline analysis computer program to allow input of measured radial pressure and temperature profiles obtained from engine or cascade data. The proposed modifications increases the input flexibility and are accomplished without changing the computer program's input format. The computer program was written by Richard M. Hearsey under a grant from the Aerospace Research Laboratory at Wright-Patterson Air Force Base. Since this report is intended to supplement the above computer program, the reader is referred to Hearsey's reports for theory, complete program listings, and detailed user's instructions.

INTRODUCTION

Detailed flow field information is important in both axial compressor analysis and design. In fan and compressor designs, this information permits evaluation of proposed improvements through efficiency calculations. In flutter research, accurate flow information gives blade incidence or angle of attack which specifies the steady state pressure distribution and hence internal blade stresses. Likewise, in stall investigations, accurate blade incidence is of primary concern.

The precision required determines which two or three-dimensional axisymmetric analysis to use. Two dimensional programs (radial velocities equal to zero) can be reasonably accurate for hub-to-tip radius ratios less than 0.8. However three-dimensional analyses must be utilized whenever radial velocities become significant. There are several such analyses that are worthy of consideration. Radial equilibrium is quite popular but requires different equations for axial stations located interior and exterior to blade rows (ref. 1). Actuator Disk Theory, on the other hand, is rather difficult to use in practice because of the closeness of 'disks' in an axial compressor (ref. 2). The Streamline Curvature technique is extremely powerful but requires at least medium computer capability (ref. 3).

Of the several streamline curvature analyses in existence, Richard Hearsey's analysis and computer pro-

N80-14051 #

gram (refs. 4 & 5) can satisfy a variety of user requirements. A partial listing of these program possibilities can be found in Appendix B. The computer program is written in three sections which include the aerodynamic streamline analysis section and two blade design sections. All parts of the program are well documented and written in standard FORTRAN IV. Reference 4 presents the theory and detailed user's instructions while reference 5 gives program listing and two examples.

Despite the versatility of Hearsey's program, the aerodynamic section of the program does not accept direct input of engine or cascade test data. This report is intended to supplement reference 4 and describes modifications to several of the aerodynamic section subroutines to allow direct input of radial pressure and temperature profiles. In addition, modifications to the CALCOMP subroutine are proposed to provide compatability with an earlier version of the plotting package. Finally, an example is prepared and run to show that the modifications are acceptable.

DISCUSSION

Streamline Computational Technique

Streamline curvature analysis is a means to solve for the velocity triangles in an axial compressor. The typical two-dimensional velocity triangles for rotors and stators (fig. 1) are extended to the radial direction to account for radial velocities which can be significant. Symmetry is assumed in the circumferential direction to simplify the analysis.

The flow field grid is established by the intersection of streamlines with specified axial stations. These axial stations may be inserted at strategic locations in the flow field such as leading and trailing blade edges or instrumentation sites. Generally, however, several stations must be specified upstream and downstream of the region of interest to define inlet and outlet conditions. The program will produce output for all defined axial stations.

The compressible flow equations are restructured and written in terms of radial, circumferential and meridional components (ref. 4). The meridional direction, shown in figure 2, lies in the radial-axial plane. These flow equations, which contain streamline curvature and slope terms, are solved given an initial guess for the streamline pattern. The program iterates, each time improving the streamline pattern until the specified tolerance is achieved. Program output consists of velocity triangle components, flow angle, streamline curvature, and thermodynamic information as a function of radius.

Aerodynamic Subroutine Modification

The aerodynamic section of Richard Hearsey's computer program (ref. 4) allows a variety of input data combinations for the rotor and stator axial stations.* 'Work' input (DATA1) can be expressed as: Total pressure, total enthalpy, absolute angular momentum, absolute whirl velocity or relative flow angle. 'Loss' input (DATA2) can be of the form: Relative total pressure loss coefficient, isentropic efficiency, or

---

*The author assumes that the reader has an understanding of the aerodynamic (or streamline) section of the reference 4 computer program and is familiar with the terminology and organization of the input deck.

entropy rise. The appropriate values of DATA1 and DATA2 are input at each axial location as a function of radius or passage height.

Typically, a researcher has correlated engine or cascade data in terms of radial pressure and temperature profiles behind rotors and pressures and flow angles (blade metal angle plus deviation angle) behind stators. Figure 3 shows the typical instrumentation of a compressor cascade. To use Hearsey's program in its present form the most favorable input combination would then be: Total pressure (DATA1, NWORK = 1) and isentropic efficiency (DATA2, NLOSS = 2) as input behind rotors and flow angle (DATA1, NWORK = 5, 6, or 7) and relative total pressure loss coefficient (DATA2, NLOSS = 1) as input behind stators.* This input selection minimizes the conversion from 'raw' data to program compatible data.

Modifications can be made to the computer program to allow direct input of station radial pressures and temperatures obtained from the cascade instrumentation. This will permit: Total pressure (DATA1) and total temperature (DATA2) as input behind rotors and flow angle (DATA1) and total pressure (DATA2) as input behind stators. Clearly these are the most advantageous input combinations when raw cascade data is available. The program modifications should be such that the original input combinations remain available to those researchers who might wish to use them. The modification scheme detailed below allows for this input flexibility.

Aerodynamic section subroutines UD0302 and UD0307 must be modified to allow total temperature input (DATA2) behind rotors and total pressure input (DATA2) behind stators. Subroutine UD0302 reads the input combinations at each station and does appropriate units conversions. Subroutine UD0307 converts the station data into streamline entropy and enthalpy to be used by the main driving program, UD03AR.

The modification concept involves sensing the magnitude of the input data and directing subroutine UD0307 to the correct computation of entropy and enthalpy utilizing the available thermodynamic subrou-

---

*The term 'relative' is misleading here but is used extensively throughout the reference 4 program instructions. Appendix C shows that the relative loss coefficient reduces to the absolute total pressure loss coefficient in stator passages.

tines, UDG1 through UDG9. For example, for stations behind rotors, total pressure (DATA1, NWORK = 1) is an acceptable input. Also, the magnitude of the isentropic efficiency (DATA2, NLOSS = 2) should never exceed 1.0. The modified program checks the magnitude of this 'loss' and if greater than 1.0, assumes the input is absolute total temperature. Subroutine UD0307 can then compute enthalpy and entropy for these stations using th rmodynamic routines UDG2 and UDG3 respectively. Hence, pressure and temperature data become a suitable input combination behind rotors. A listing of the modified subroutine, UD0307, which contains this limit checking sequence appears in Appendix D. Lines 61 - 65 and 69 - 72 contain the specific changes for rotor stations.

Similar modifications are incorporated in subroutine UD0307 to handle stations behind stators. At these locations, flow angle (DATA1, NWORK = 5, 6, or 7) is an acceptable input. The relative total pressure loss coefficient (DATA2, NLOSS = 1) should be a small pc itive number. In fact, for efficient compressor blading, the relative total pressure loss coefficient should be less than 0.3. In this instance, however the magnitude of the loss coefficient does not have an upper limit as did the isentropic efficiency. An arbitrary limit value of 2.0 was chosen assuming the magnitude of the total pressure (in any chosen set of program acceptable units) should never be smaller than this value. Again the magnitude checking procedure is called upon to scan for input data greater than the limit value of 2.0. If this condition is satisfied for all of the station input data, the program assumes total pressure as input instead of loss coefficients. The modified UD0307 subroutine can then calculate entropy from the total pressure input and the enthalpy of the previous station using subroutine UDG3 (i.e. enthalpy is assumed constant across stator passages). Sul outine UD0302, the input routine, also needs the li it checking sequence because a units conversion may be required depending on the value of the scaling factor, SCLFAC. Subroutine UD0307, lines 142 - 146 and 158 - 160, and subroutine UD0302, lines 190 - 198 contain the specific modifications to allow the input combination of flow angle and total pressure behind stationary blades. A source listing of UD0302 may be found in Appendix E.

CALCOMP Subroutine Modification

Several different versions of the well-known CALCOMP plotting package exist. The main difference in the versions is the number of arguments in the subroutine calls to the individual CALCOMP routines. Hearsey's program was written utilizing the 1970 version of the plotting package (ref. 6).

The 1964-65 Version 5 CALCOMP package requires additional arguments in the 'CALL' statements to routines AXIS and LINE. The AXIS routine needs one additional argument, DV, which indicates the number of divisions per ten inches of paper to be drawn on the axis. The LINE routine requires four additional arguments - XMIN, DX, YMIN, and DY. XMIN and YMIN are minimum coordinate values on the graph. DX and DY represent the division mark incremental value for the respective axis.

Subroutine UD0312, the plotting routine in the aerodynamic or streamline section of Hearsey's program, has several CALCOMP calls to routines AXIS and LINE. This subroutine is responsible for plotting static pressure distributions (NPLOT = 1) and final streamline mesh (NPLOT = 2). A 1964-65 Version 5 compatible listing of subroutine UD0312 may be found in Appendix F.

Example

Steady state data from a NASA-Lewis full scale engine test is used as input to the modified version of Richard Hearsey's aerodynamic program (ref. 4) detailed in this report and an industry developed streamline program. The industry program makes for a good comparison because it was developed for analysis of engine test data and hence accepts radial pressure and temperature distributions directly. This industry program, however, is not as flexible as the reference 4 program nor is it available to the general public.

The analysis concentrates on the three stage fan module at the inlet of a typical modern turbofan engine. The axial stations for computer output are defined and numbered in Figure 4. Stations at leading and trailing edges of blades follow the blade lean in the axial direction. Radial total pressure and temperature profiles were available behind the first two rotors and behind the flow straightener (F.S.). Flow angle was measured at only one axial location - behind the inlet guide vane (IGV). The deviation angles behind the remaining stationary blades are assumed to be zero; that is, the flow angle equals the trailing edge blade metal angle at these locations. From calibration plots for the flow angle probe and individual probe measurements, total pressure can be obtained thus allowing pressure loss input for the IGV (ref. 7). Figure 5 shows the non-dimensional pressure profile behind the IGV. Because of flow angle probe structural limitations data was only obtained in the upper 40 percent of the annulus. The remainder of the curve represe^-ts an approximate profile. Total pressure profiles at other locations are shown in Figures 6, 7, and 8; total temperature profiles are shown in Figure 9. In all figures zero percent span corresponds to the fan hub. Approximate stator losses for stator 1, stator 2, and the flow straightener were computed from blade geometry and NACA tables (ref. 8).

Station input data is entered into the aerodynamic section of the reference 4 computer program in the following manner:

Station 1:	Constant pressure, P_1 = 136.907 KPa (19.856 psia), and temperature, T_1 = 442.82K (797.08R), exist across engine inlet annulus. Whirl angle is assumed to be zero.
Stations 2-4	No data is input (NDATA = 0). These stations are used to establish the flow field at the inlet.
Station 5, IGV inlet:	Constant entropy (NWORK = 0) from previous station (NL1 = -1) assumed for five radial points (NDATA = 5).
Station 6, IGV outlet:	Twenty-one data points (NDATA = 21) specify flow angle (NWORK = 6) and total pressure (NLOSS= 1) as a function of radius. 136.392 < P < 136.795 KPa.

Station 7, first rotor inlet: No data is input.

Station 8, first rotor outlet: Twenty-one data points specify total pressure (NWORK = 1) and total temperature (NLOSS = 2) as a function of radius. $473.82 < T < 495.96$ K.

Notice that the modified reference 4 program is called upon to accept total pressure input for a stator station (station 6) and total temperature input for a rotor station (station 8). At station 6, the magnitude of all DATA2 (NLOSS = 1) input is greater than the arbitrary limit value of 2.0, hence total pressure input is assumed. At station 8, the magnitude of all DATA2 (NLOSS = 2) input is greater than the limit value of 1.0, hence total temperature input is assumed. The next nine stations (9 - 17) follow the pattern established by stations 5 - 8 above. The last three stations, 19 - 21, set the outlet conditions. Input data at all points is interpreted by a spline fit (NTERP = 0) and read in as a function of percent radial span (NDIMEN = 3). A constant annulus wall boundary layer thickness is assumed for all axial stations.

The subsonic solution (NMACH = 0) to the flow field was obtained at all specified axial stations. The modified reference 4 computer program converged to within a tolerance of 0.3 percent (TOLNCE = 0.003) after 53 iterations. Output for all twenty-one streamlines at each station consist of velocity components, radius of curvature, pressures, temperatures, and flow angles. The industry program was prepared and run with the same input information. The result was convergence to within tolerance of 0.25 percent after 38 iterations.

Computer output for stations 7 and 8 for the modified reference 4 program and the industry program is compared in figures 10 through 15. Figures 10, 11, and 12 compare total pressure, total temperature, and flow angle, respectively. Figures 13, 14, and 15 compare the velocity triangle components: Meridional, radial, and relative velocity. In these figures, the modified reference 4 program output is symbolized with triangles while the industry program output appears as circles. All data is expressed as normalized ratios with the normalization constant as indicated on each graph.

Relatively good agreement is obtained between the modified reference 4 computer program and the industry program. The greatest variations in the output of the two streamline analyses appears at the annulus boundaries. This is probably due to different methods of handling the boundary information or differences in the internal program interpolation (spline fitting) routines. This is best evidenced in Figures 10(b) and 11(b). In both figures the industry program output deviates from both the reference 4 program output and the specified pressure and temperature input profiles at the fan hub (zero percent span). This descrepancy leads to differences in velocity and flow angle which can be seen in the remaining figures.

The significance of this output discrepancy depends on the initial purpose of the analysis. As an example, the maximum difference in station 7 blade hub inlet flow angle, from Figure 12(a), is approximately 3.39° $(-3.08 < \alpha < 0.31^{\circ})$. This change in inlet flow angle at the hub will not greatly affect the overall blade pressure distribution because of the good agreement in flow angle output at spans greater than 10 percent. However, since the onset of stall flutter is sensitive to steady state blade incidence (ref. 9), this flow angle difference at the hub could be a concern.* Since the pressure and temperature output of the modified reference 4 computer program matches the pressure and temperature input profiles at stations 7 and 8, velocity components, flow angles, and remaining thermodynamic output is more credible than the industry program output.

Despite the differences in output from the two computer programs, a three dimensional analysis is much more accurate than a two-dimensional (radial velocity equal to zero) analysis. A two-dimensional program run by the author gave a station 7 flow angle of 14.15°. The modified reference 4 program shows that the true station 7 flow angle ranges from -3.08° at the hub to 14.56° near 75 percent span. The relative magnitudes of the velocity components, shown in Figures 13, 14, and 15 also indicate the significance of the radial velocity for this example.

---

*If this difference in blade inlet flow angle had occurred at the blade tip, a more serious situation would result because the blade is more critically loaded in this area.

Figures 16 and 17 demonstrate the modified CALCOMP plotting routine, UD0312, in the aerodynamic section of the modified reference 4 program. Figure 16 is a streamline plot which shows the streamline contraction through the fan module. The static pressure distribution for the hub, mid, and tip streamlines for the fan is shown in Figure 17.

SUMMARY OF RESULTS

The aerodynamic section of Richard Hearsey's axial compressor streamline computer program was successfully modified to allow direct input of measured radial pressure and temperature profiles obtained from engine or cascade data. Subroutines UD0302 and UD0307 contain the specific modifications to permit the input combination of flow angle and total pressure for stators and the input combination of total pressure and total temperature for rotors. All modifications were accomplished without changing the computer program's original input format.

The internal CALCOMP subroutine, UD0312, which plots streamline mesh and static pressure distributions, was modified to be compatible with an old version (1964-65, Version 5) of the plotting package. The user must decide which version of UD0312 satisfies his computer facility requirements.

The included example demonstrated how to construct the input deck for the aerodynamic section of the modified Hearsey program. The modified program was run and the results were compared with an industry program which accepted radial pressure and temperature profiles directly. Good agreement was obtained from the two analyses which indicates that the modifications to Hearsey's program were acceptable.

APPENDIX A

Symbols:

P	total pressure, Pa
Pr	relative total pressure, Pa
p	static pressure, Pa
T	total temperature, R
ρ	static density, Kg/m^3
$\bar{\omega}$	relative total pressure loss coefficient
α	absolute flow angle, deg
β	relative flow angle, deg
C	absolute velocity, m/sec
W	relative velocity, m/sec
U	'bucket' or rotor velocity, m/sec

Subscripts:

x	axial direction
y	tangential direction
r	radial direction
m	meridional direction
A	rotor inlet
B	rotor outlet
1-21	axial station identifier

APPENDIX B

The following list highlights the flexibility built into Richard Hearsey's Axial Compressor Program, references 4 and 5. Some of the specific program capabilities are:

1. Versatility to use program for design, off-design, and analysis computations.
2. Allow intrablade station locations.
3. Two forms of momentum equation to calculate subsonic and supersonic solutions of the flow field.
4. Allow input in any set of consistent units.
5. Ability to specify inlet flow distributions.
6. Ability to run program without specifying some nonzero loss criterion.
7. Multiple speed runs without inputting entire deck (just specify percent 'design' speed).
8. Plots of midstreamline position showing convergence or divergence of analysis versus iteration at each station location.
9. CALCOMP plots of final streamline mesh and static pressure distribution.
10. NASTRAN compatible output (pressure difference for calculation of blade stresses).
11. Code does blade design as well as aerodynamic streamline calculation.
12. Arbitrary blade shapes can be designed.

APPENDIX C

In stator passages, the relative total pressure loss coefficient reduces to the absolute total pressure loss coefficient as follows:

A subscript implies stator inlet.
B subscript implies stator outlet.

Symbol definitions from Hearsey's program instructions, reference 4 and figure 1:

Pr' - Insentropic relative total pressure
Pr - Relative total pressure
P - Total pressure
p - Static pressure
ρ - Density

$$\bar{\omega} = \frac{Pr' - Pr}{Pr_A - p_A}$$

For incompressible flow:

$$Pr' = Pr_A = p_A + 1/2\ \rho W_A^2$$

$$Pr = Pr_B = p_B + 1/2\ \rho W_B^2$$

$$\bar{\omega} = \frac{Pr_A - Pr_B}{1/2\ \rho W_A^{\ 2}}$$

$$\bar{\omega} = \frac{p_A + 1/2\ \rho W_A^{\ 2} - (p_B + 1/2\ \rho W_B^{\ 2})}{1/2\ \rho W_A^{\ 2}} \qquad (1)$$

$$P_A = p_A + 1/2\ \rho C_A^2 = p_A + 1/2\ \rho\,(Cx_A^2 + Cy_A^2)$$

$$P_A = p_A + 1/2\ \rho \left[Cy_A^2 + W_A^2 - Wy_A^2 \right]$$

$$P_A = p_A + 1/2\ \rho \left[Cy_A^2 + W_A^2 - (U - Cy_A)^2 \right]$$

Or:

$$\left.\begin{aligned} &1/2\ \rho W_A^2 = P_A - p_A + \rho U\,(U/_2 - Cy_A) \\ &\text{Likewise:} \\ &1/2\ \rho W_B^2 = P_B - p_B + \rho U\,(U/_2 - Cy_B) \end{aligned}\right\} \quad (2)$$

Substituting (2) into (1):

$$\bar{\omega} = \frac{p_A - p_B + \left[\left[P_A - p_A + \rho U \left(\frac{U}{2} - Cy_A \right) \right] - \left[P_B - p_B + \rho U \left(\frac{U}{2} - Cy_B \right) \right] \right]}{P_A - p_A + \rho U \left(\frac{U}{2} - Cy_A \right)}$$

$$\bar{\omega} = \frac{P_A - P_B + \rho U\,(Cy_B - Cy_A)}{P_A - p_A + \rho U \left(\frac{U}{2} - Cy_A \right)}$$

For stators, U = 0, therefore:

$$\bar{\omega} = \frac{P_{\cdot} - P_B}{P_A - p_A} = \frac{P_A - P_B}{1/2\,\rho\,C_A{}^2} \qquad (3)$$

This is the definition of absolute total pressure loss coefficient.

APPENDIX D

Source listing of the modified subroutine UD0307, which supplies UD03AR with enthalpy and entropy, appears below. Modifications are on lines 61 - 65, 69 - 72, 142 - 146, and 158 - 160.

```
 1           SUBROUTINE UD0307                                                 $07$   2
 2           REAL LOSS,LAMI,LAMIP1,LAMIM1                                      $07$   3
 3           COMMON NSTNS,NSTRMS,NMAX,NFORCE,NBL,NCASE,NSPLIT,NREAD,NPUNCH,NPAG$07$   4
 4          1E,NSET1,NSET2,ISTAG,ICASE,IFAILO,IPASS,I,IVFAIL,IFFAIL,NMIX,NTRANS$07$   5
 5          2,NPLOT,ILOSS,LNCT,ITUB,IMID,IFAIL,ITER,LOG1,LOG2,LOG3,LOG4,LOG5,LO$07$   6
 6          3G6,IPRINT,NMANY,NSTPLT,NEQN                                       $07$   7
 7           COMMON NSPEC(30),NWORK(30),NLOSS(30),NDATA(30),NTERP(30),NMACH(30)$07$   8
 8          1,NL1(30),NL2(30),NDIMEN(30),IS1(30),IS2(30),IS3(30),NEVAL(30),NDIF$07$   9
 9          2F(4),NDEL(30),NLITER(30),NM(2),NRAD(2),NCURVE(30),NWHICH(30),NOUT1$07$  10
10          3(30),NOUT2(30),NOUT3(30),NBLADE(30)                               $07$  11
11           COMMON DM(11,5,2),WFRAC(11,5,2)                                   $07$  12
12           COMMON R(21,30),XL(21,30),X(21,30),H(21,30),S(21,30),VM(21,30),VW($07$  13
13          121,30),TBETA(21,30),DIFF(15,4),FDHUB(15,4),FDMID(15,4),FDTIP(15,4)$07$  14
14          2,TERAD(5,2)                                                       $07$  15
15           COMMON DATAC(100),DATA1(100),DATA2(100),DATA3(100),DATA4(100),DATA$07$  16
16          15(100),DATA6(100),DATA7(100),DATA8(100),DATA9(100),FLOW(10),SPEED($07$  17
17          230),SPDFAC(10),BBLOCK(30),BDIST(30),WBLOCK(30),WWBL(30),XSTN(150),$07$  18
18          3RSTN(150),DELF(30),DELC(100),DELTA(100),TITLE(18),DRDM2(30),RIM1(3$07$  19
19          40),XIM1(30),WORK(21),LOSS(21),TANEPS(21),XI(21),VV(21),DELW(21),LA$07$  20
20          5MI(21),LAMIM1(21),LAMIP1(21),PHI(21),CR(21),GAMA(21),SPPG(21),CPPG$07$  21
21          6(21),HKEEP(21),SKEEP(21),VWKEEP(21),DELH(30),DELT(30)             $07$  22
22           COMMON VISK,SHAPE,SCLFAC,EJ,G,TOLNCE,XSCALE,PSCALE,PLOW,RLOW,XMMAX$07$  23
23          1,RCONST,FM2,HMIN,C1,PI,CONTR,CONMX                                $07$  24
24           L1=I+NL1(I)                                                       $07$  25
25           L2=I+NL2(I)                                                       $07$  26
26           IW=NWORK(I)                                                       $07$  27
27           IL=NLOSS(I)                                                       $07$  28
28           XN=SPEED(I)*SPDFAC(ICASE)*PI/(30.0*SCLFAC)                        $07$  29
29           GO TO(100,250,270,290,440,440,440),IW                             $07$  30
30   100     GO TO(110,190,210,110),IL                                         $07$  31
31   110     IF(L2.NE.I)GO TO 150                                              $07$  32
32           DO 140 J=1,NSTRMS                                                 $07$  33
33           IF(IPASS.EQ.1.AND.ITER.EQ.0)GO TO 120                             $07$  34
34           IF(ITER.EQ.0)VV(J)=VM(J,I)                                        $07$  35
35           X1=H(J,I)-(VV(J)**2+VW(J,I)**2)/(2.0*G*EJ)                        $07$  36
36           X2=H(J,I)-(VW(J,I)**2-(VW(J,I)-XN*R(J,I))**2)/(2.0*G*EJ)          $07$  37
37           IF(X1.LT.HMIN)X1=HMIN                                             $07$  38
38           IF(X2.LT.HMIN)X2=HMIN                                             $07$  39
39           X3=1.0/(1.0+LOSS(J)*(1.0-UDG4(X1,S(J,I))/UDG4(X2,S(J,I))))        $07$  40
40           GO TO 130                                                         $07$  41
41   120     X3=1.0                                                            $07$  42
42   130     H(J,I)=UDG2(S(J,L1),WORK(J)/X3)                                   $07$  43
43   140     S(J,I)=UDG3(WORK(J),H(J,I))                                       $07$  44
44           GO TO 230                                                         $07$  45
45   150     DO 180 J=1,NSTRMS                                                 $07$  46
46           IF(IPASS.EQ.1.AND.L2.GT.I)GO TO 160                               $07$  47
47           X1=H(J,L1)-(VW(J,L1)**2-(VW(J,L1)-XN*R(J,L1))**2)/(2.0*G*EJ)+XN**2$07$  48
48          1*(R(J,I)**2-R(J,L1)**2)/(2.0*G*EJ)                                $07$  49
49           IF(X1.LT.HMIN)X1=HMIN                                             $07$  50
50           X2=H(J,L2)-(VM(J,L2)**2+VW(J,L2)**2)/(2.0*G*EJ)                   $07$  51
51           X3=H(J,L2)-(VW(J,L2)**2-(VW(J,L2)-XN*R(J,L2))**2)/(2.0*G*EJ)      $07$  52
52           IF(X2.LT.HMIN)X2=HMIN                                             $07$  53
53           IF(X3.LT.HMIN)X3=HMIN                                             $07$  54
54           X4=1.0-LOSS(J)/UDG4(X1,S(J,L1))*(UDG4(X3,S(J,L2))-UDG4(X2,S(J,L2))$07$  55
55          1)                                                                 $07$  56
56           GO TO 170                                                         $07$  57
```

```
 160   X4=1.0                                                              $075  58
 170   H(J,I)=UDG2(S(J,L1),WORK(J)/X4)                                     $075  59
 180   S(J,I)=UDG3(WORK(J),H(J,I))                                         $075  60
       GO TO 230                                                           $075  61
   190 KTEMP=0                                                             MOD.-JGW
       DO 192 J=1,NSTRMS                                                   MOD.-JGW
   192 IF(ABS(LOSS(J)).GT.1.0) KTEMP=KTEMP+1                               MOD.-JGW
       IF(KTEMP.EQ.NSTRMS)GO TO 205                                        MOD.-JGW
       DO 200 J=1,NSTRMS                                                   MOD.-JGW
       H(J,I)=H(J,L1)+(UDG2(S(J,L1),WORK(J))-H(J,L1))/LOSS(J)              $075  63
   200 S(J,I)=UDG3(WORK(J),H(J,I))                                         $075  64
       GO TO 230                                                           $075  65
   205 DO 207 J=1,NSTRMS                                                   MOD.-JGW
       H(J,I)=UDG6(WORK(J),LOSS(J))                                        MOD.-JGW
   207 S(J,I)=UDG3(WORK(J),H(J,I))                                         MOD.-JGW
       GO TO 230                                                           MOD.-JGW
 210   DO 220 J=1,NSTRMS                                                   $075  66
       S(J,I)=S(J,L1)+LOSS(J)                                              $075  67
 220   H(J,I)=UDG2(S(J,I),WORK(J))                                         $075  68
 230   DO 240 J=1,NSTRMS                                                   $075  69
 240   VW(J,I)=(XN*RIM1(J)*VW(J,I-1)+(H(J,I)-H(J,I-1))*G*EJ)/(XN*R(J,I)) $075  70
       GO TO 570                                                           $075  71
 [illegible] DO 260 J=1,NSTRMS                                             $075  72
       H(J,I)=WORK(J)                                                      $075  73
 260   VW(J,I)=(XN*RIM1(J)*VW(J,I-1)+(H(J,I)-H(J,I-1))*G*EJ)/(XN*R(J,I)) $075  74
       GO TO 330                                                           $075  75
 270   DO 280 J=1,NSTRMS                                                   $075  76
 280   VW(J,I)=WORK(J)/R(J,I)                                              $075  77
       GO TO 310                                                           $075  78
 290   DO 300 J=1,NSTRMS                                                   $075  79
 300   VW(J,I)=WORK(J)                                                     $075  80
 310   DO 320 J=1,NSTRMS                                                   $075  81
 320   H(J,I)=H(J,I-1)+XN*(R(J,I)*VW(J,I)-RIM1(J)*VW(J,I-1))/(G*EJ)       $075  82
 330   GO TO(340,400,420,340),IL                                           $075  83
 340   IF(L2.NE.I)GO TO 370                                                $075  84
       DO 360 J=1,NSTRMS                                                   $07[illegible]
       IF(IPASS.EQ.1.AND.ITER.EQ.0)GO TO 350                               $075  [illegible]
       IF(ITER.EQ.0)VV(J)=VM(J,I)                                          $075  87
       X1=H(J,I)-(VV(J)**2+VW(J,I)**2)/(2.0*G*EJ)                          $075  88
       X2=H(J,I)-(VW(J,I)**2-(VW(J,I)-XN*R(J,I))**2)/(2.0*G*EJ)            $075  89
       IF(X1.LT.HMIN)X1=HMIN                                               $075  90
       IF(X2.LT.HMIN)X2=HMIN                                               $075  91
       X3=1.0/(1.0+LOSS(J)*(1.0-UDG4(X1,S(J,I))/UDG4(X2,S(J,I))))          $075  92
       GO TO 360                                                           $075  93
 3[illegible]  X3=1.0                                                      $075  94
 3[illegible]  S(J,I)=UDG3(X3*UDG4(H(J,I),S(J,L1)),H(J,I))                 $075  95
       GO TO 570                                                           $075  96
 370   DO 390 J=1,NSTRMS                                                   $075  97
       IF(IPASS.EQ.1.AND.L2.GT.I)GO TO 380                                 $075  98
       X1=H(J,L1)-(VW(J,L1)**2-(VW(J,L1)-XN*R(J,L1))**2)/(2.0*G*EJ)+XN**2$075  99
      1*(R(J,I)**2-R(J,L1)**2)/(2.0*G*EJ)                                  $075 100
       IF(X1.LT.HMIN)X1=HMIN                                               $075 101
       X2=H(J,L2)-(VM(J,L2)**2+VW(J,L2)**2)/(2.0*G*EJ)                     $075 102
       X3=H(J,L2)-(VW(J,L2)**2-(VW(J,L2)-XN*R(J,L2))**2)/(2.0*G*EJ)        $075 103
       IF(X2.LT.HMIN)X2=HMIN                                               $075 104
       IF(X3.LT.HMIN)X3=HMIN                                               $075 105
       X4=1.0-LOSS(J)/UDG4(X1,S(J,L1))*(UDG4(X3,S(J,L2))-UDG4(X2,S(J,L2)))$075 106
```

```
          1)                                                                 $07$ 107
          GO TO 390                                                          $07$ 108
  380     X4=1.0                                                             $07$ 109
  390     S(J,I)=UDG3(X4*UDG4(H(J,I),S(J,L1)),H(J,I))                        $07$ 110
          GO TO 570                                                          $07$ 111
  400     DO 410 J=1,NSTRMS                                                  $07$ 112
  410     S(J,I)=UDG3(UDG4(H(J,L1)+LOSS(J)*(H(J,I)-H(J,L1)),S(J,L1)),H(J,I))$07$ 113
          GO TO 570                                                          $07$ 114
  420     DO 430 J=1,NSTRMS                                                  $07$ 115
  430     S(J,I)=S(J,L1)+LOSS(J)                                             $07$ 116
          GO TO 570                                                          $07$ 117
  440     DO 450 J=1,NSTRMS                                                  $07$ 118
  450     XI(J)=H(J,I-1)-XN*RIMI(J)*VW(J,I-1)/(G*EJ)                         $07$ 119
          GO TO(460,510,550,460),IL                                          $07$ 120
  460     IF(L2.NE.I)GO TO 490                                               $07$ 121
          DO 480 J=1,NSTRMS                                                  $07$ 122
          X2=XI(J)+(XN*R(J,I))**2/(2.0*G*EJ)                                 11/01/77
          IF(IPASS.EQ.1.AND.ITER.EQ.0)GO TO 470                              $07$ 123
          IF(ITER.EQ.0)VV(J)=VM(J,I)                                         $07$ 124
C ....DELETE THIS STATEMENT....UPDATE: 11/01/77                              $07$ 125
          X1=X2-VV(J)**2*(1.0+TBETA(J,I)**2)/(2.0*G*EJ)                      $07$ 126
          IF(X1.LT.HMIN)X1=HMIN                                              $07$ 127
          IF(X2.LT.HMIN)X2=HMIN                                              $07$ 128
          X3=1.0/(1.0+LOSS(J)*(1.0-UDG4(X1,S(J,I))/UDG4(X2,S(J,I))))         $07$ 129
          GO TO 480                                                          $07$ 130
  470     X3=1.0                                                             $07$ 131
  480     S(J,I)=UDG3(X3*UDG4(X2,S(J,L1)),X2)                                $07$ 132
          GO TO 570                                                          $07$ 133
    490 KPRES=0                                                              MOD.-JGW
          DO 492 J=1,NSTRMS                                                  MOD.-JGW
    492 IF(ABS(LOSS(J)).GT.2.0)KPRES=KPRES+1                                 MOD.-JGW
          IF(KPRES.EQ.NSTRMS)GO TO 505                                       MOD.-JGW
          DO 500 J=1,NSTRMS                                                  MOD.-JGW
          X4=XI(J)+(XN*R(J,I))**2/(2.0*G*EJ)                                 $07$ 135
          IF(X4.LT.HMIN)X4=HMIN                                              $07$ 136
          X1=UDG4(X4,S(J,L1))                                                $07$ [illegible]
          IF(IPASS.EQ.1.AND.L2.GT.I)GO TO 500                                $07$ [illegible]
          X2=XI(J)+(XN*R(J,L2))**2/(2.0*G*EJ)                                $07$ 139
          X3=H(J,L2)-(VM(J,L2)**2+VW(J,L2)**2)/(2.0*G*EJ)                    $07$ 140
          IF(X2.LT.HMIN)X2=HMIN                                              $07$ 141
          IF(X3.LT.HMIN)X3=HMIN                                              $07$ 142
          X1=X1-LOSS(J)*(UDG4(X2,S(J,L2))-UDG4(X3,S(J,L2)))                  $07$ 143
  500     S(J,I)=UDG3(X1,X4)                                                 $07$ 144
          GO TO 570                                                          $07$ 145
    505 DO 506 J=1,NSTRMS                                                    MOD.-JGW
    506 S(J,I)=UDG3(LOSS(J),XI(J))                                           MOD.-JGW
          GO TO 570                                                          MOD.-JGW
  510     IF(IPASS.EQ.1.AND.ITER.EQ.0)GO TO 530                              $07$ 146
          DO 520 J=1,NSTRMS                                                  $07$ 147
          IF(ITER.EQ.0)VV(J)=VM(J,I)                                         $07$ 148
          X1=H(J,I-1)+XN*(VV(J)*(TBETA(J,I)+XN*R(J,I)/VV(J))*R(J,I)-RIM1(J)*$07$ 149
         1VW(J,I-1))/(G*EJ)                                                  $07$ 150
          IF(X1.LT.HMIN)X1=HMIN                                              $07$ 151
          X2=UDG4(H(J,L1)+(X1-H(J,L1))*LOSS(J),S(J,L1))                      $07$ 152
  520     S(J,I)=UDG3(X2,X1)                                                 $07$ 153
          GO TO 570                                                          $07$ 154
  530     DO 540 J=1,NSTRMS                                                  $07$ 155
```

```
540   S(J,I)=S(J,L1)                                S07S 156
      GO TO 570                                     S07S 157
550   DO 560 J=1,NSTRMS                             S07S 158
560   S(J,I)=S(J,L1)+LOSS(J)                        S07S 159
570   RETURN                                        S07S 160
      END                                           S07S 161
```

Source listing of modified subroutine UD0302, the input routine, appears below. Modifications are on lines 190 - 198. Statement labels which indicate a date (line 177: 11/30/77) are program updates.

```
 1          SUBROUTINE UD0302                                                    $02$   2
 2          REAL LOSS,LAMI,LAMIP1,LAMIM1                                         $02$   3
 3          COMMON NSTNS,NSTRMS,NMAX,NFORCE,NBL,NCASE,NSPLIT,NREAD,NPUNCH,NPAG$02$   4
 4         1E,NSET1,NSET2,ISTAG,ICASE,IFAILO,IPASS,I,IVFAIL,IFFAIL,NMIX,NTRANS$02$   5
 5         2,NPLOT,ILOSS,LNCT,ITUB,IMID,IFAIL,ITER,LOG1,LOG2,LOG3,LOG4,LOG5,LO$02$   6
 6         3G6,IPRINT,NMANY,NSTPLT,NEQN                                        $02$   7
 7          COMMON NSPEC(30),NWORK(30),NLOSS(30),NDATA(30),NTERP(30),NMACH(30)$02$   8
 8         1,NL1(30),NL2(30),NDIMEN(30),IS1(30),IS2(30),IS3(30),NEVAL(30),NDIF$02$   9
 9         2F(4),NDEL(30),NLITER(30),NM(2),NRAD(2),NCURVE(30),NWHICH(30),NOUT1$02$  10
10         3(30),NOUT2(30),NOUT3(30),NBLADE(30)                                $02$  11
11          COMMON DM(11,5,2),WFRAC(11,5,2)                                      $02$  12
12          COMMON R(21,30),XL(21,30),X(21,30),H(21,30),S(21,30),VM(21,30),VW($02$  13
13         121,30),TBETA(21,30),DIFF(15,4),FDHUB(15,4),FDMID(15,4),FDTIP(15,4)$02$  14
14         2,TERAD(5,2)                                                        $02$  15
15          COMMON DATAC(100),DATA1(100),DATA2(100),DATA3(100),DATA4(100),DATA$02$  16
16         15(100),DATA6(100),DATA7(100),DATA8(100),DATA9(100),FLOW(10),SPEED($02$  17
17         230),SPDFAC(10),BBLOCK(30),BDIST(30),WBLOCK(30),WWBL(30),XSTN(150),$02$  18
18         3RSTN(150),DELF(30),DELC(100),DELTA(100),TITLE(18),DRDM2(30),RIM1(3$02$  19
19         40),XIM1(30),WORK(21),LOSS(21),TANEPS(21),XI(21),VV(21),DELW(21),LA$02$  20
20         5MI(21),LAMIM1(21),LAMIP1(21),PHI(21),CR(21),GAMA(21),SPPG(21),CPPG$02$  21
21         6(21),HKEEP(21),SKEEP(21),VWKEEP(21),DELH(30),DELT(30)              $02$  22
22          COMMON VISK,SHAPE,SCLFAC,EJ,G,TOLNCE,XSCALE,PSCALE,PLOW,RLOW,XMMAX$02$  23
23         I,RCONST,FM2,HMIN,C1,PI,CONTR,CONMX                                 $02$  24
24          DIMENSION II(21,30),JJ(21,30)                                        $02$  25
25          EQUIVALENCE (H,II),(S,JJ)                                            $02$  26
26          COMMON/PAGE/LIMIT,LQ                                                 $02$  27
27          NEVAL(1)=0                                                           $02$  28
28          READ(LOG1,100)TITLE                                                  $02$  29
29     100  FORMAT(18A4)                                                         $02$  30
30          WRITE(LOG2,110)TITLE                                                 $02$  31
31     110  FORMAT(10X,10HINPUT DATA,/,10X,10(1H*),//,10X,5HTITLE,34X,2H= ,18A$02$  32
32         14)                                                                 $02$  33
33          LNCT=LNCT+4                                                          $02$  34
34          CALL UDG1(LNCT)                                                      $02$  35
35          READ(LOG1,120)NSTNS,NSTRMS,NMAX,NFORCE,NBL,NCASE,NSPLIT,NSET1,NSET$02$  36
36         12,NREAD,NPUNCH,NPLOT,NPAGE,NTRANS,NMIX,NMANY,NSTPLT,NEQN           $02$  37
37     120  FORMAT(18I3)                                                         $02$  38
38          IF(NSTRMS.EQ.0)NSTRMS=11                                             $02$  39
39          IF(NMAX.EQ.0)NMAX=40                                                 $02$  40
40          IF(NFORCE.EQ.0)NFORCE=10                                             $02$  41
41          IF(NCASE.EQ.0)NCASE=1                                                $02$  42
42          IF(NPAGE.EQ.0)NPAGE=80                                               $02$  43
43          LQ=LOG2                                                              $02$  44
44          LIMIT=NPAGE                                                          $02$  45
45          CALL UD0303(LNCT,19)                                                 $02$  46
46          WRITE(LOG2,130)NSTNS,NSTRMS,NMAX,NFORCE,NBL,NCASE,NSPLIT,NSET1,NSE$02$  47
47         1T2,NREAD,NPUNCH,NPLOT,NPAGE,NTRANS,NMIX,NMANY,NSTPLT,NEQN          $02$  48
48     130  FORMAT(2X,/,10X,18HNUMBER OF STATIONS,21X,1H=,I3,/,10X,21HNUMBER O$02$  49
49         1F STREAMLINES,18X,1H=,I3,/,10X,20HMAX NUMBER OF PASSES,19X,1H=,I3,$02$  50
50         2/,10X,30HMAX NUMBER OF ARBITRARY PASSES,9X,1H=,I3,/,10X,29HBOUNDAR$02$  51
51         3Y LAYER CALC INDICATOR,10X,1H=,I3,/,10X,24HNUMBER OF RUNNING POINT$02$  52
52         4S,15X,1H=,I3,/,10X,33HSTREAMLINE DISTRIBUTION INDICATOR,6X,1H=,I3,$02$  53
53         5/,10X,34HNUMBER OF LOSS/D-FACTOR CURVE SETS,5X,1H=,I3,/,10X,34HNUM$02$  54
54         6BER OF LOSS/T.E.LOSS CURVE SETS,5X,1H=,I3,/,10X,26HSTREAMLINE INPU$02$  55
55         7T INDICATOR,13X,1H=,I3,/,10X,27HSTREAMLINE OUTPUT INDICATOR,12X,1H$02$  56
56         8=,I3,/,10X,24HPRECISION PLOT INDICATOR,15X,1H=,I3,/,10X,24HMAX NUM$02$  57
```

```
          9BER OF LINES/PAGE,15X,1H=,I3,/,10X,29HWAKE TRANSPORT CALC INDICATO$02$   58
          AR,10X,1H=,I3,/,10X,32HMAINSTREAM MIXING CALC INDICATOR,7X,1H=,I3,/$02$   59
          B,10X,33HNO OF STATIONS FROM ANALYTIC SECN,6X,1H=,I3,/,10X,27HLINE-$02$   60
          CPRINTER PLOT INDICATOR,12X,1H=,I3,/,10X,32HMOMENTUM EQUATION FORM $02$   61
          DINDICATOR,7X,1H=,I3)                                               $02$   62
           ITUB=NSTRMS-1                                                      $02$   63
           IMID=NSTRMS/2+1                                                    $02$   64
           IF(NMANY.EQ.0)GO TO 136                                            $02$   65
           READ(LOG1,132)(NWHICH(I),I=1,NMANY)                                $02$   66
    132    FORMAT(24I3)                                                       $02$   67
           CALL UD0303(LNCT,2)                                                $02$   68
           WRITE(LOG2,134)(NWHICH(I),I=1,NMANY)                               $02$   69
    134    FORMAT(2X,/,10X,51HGEOMETRY COMES FROM ANALYTIC SECTION FOR STATIO$02$   70
          1NS   ,23I3)                                                        $02$   71
    136    CALL UD0303(LNCT,7)                                                $02$   72
           READ(LOG1,140)G,EJ,SCLFAC,TOLNCE,VISK,SHAPE                        $02$   73
    140    FORMAT(6F12.0)                                                     $02$   74
           IF(G.EQ.0.0)G=32.174                                               $02$   75
           IF(EJ.EQ.0.0)EJ=778.16                                             $02$   76
           IF(SCLFAC.EQ.0.0)SCLFAC=12.0                                       $02$   77
           IF(TOLNCE.EQ.0.0)TOLNCE=0.001                                      $02$   78
           IF(VISK.EQ.0.0)VISK=0.00018                                        $02$   79
           IF(SHAPE.EQ.0.0)SHAPE=0.7                                          $02$   80
           WRITE(LOG2,150)G,EJ,SCLFAC,TOLNCE,VISK,SHAPE                       $02$   81
    150    FORMAT(2X,/,10X,22HGRAVITATIONAL CONSTANT,17X,1H=,F8.4,/,10X,17HJO$02$   82
          1ULES EQUIVALENT,22X,1H=,F8.3,/,10X,29HLINEAR DIMENSION SCALE FACTO$02$   83
          2R,10X,1H=,F8.4,/,10X,15HBASIC TOLERANCE,24X,1H=,F8.5,/,10X,19HKINE$02$   84
          3MATIC VISCOSITY,20X,1H=,F8.5,/,10X,17HB.L. SHAPE FACTOR,22X,1H=,F8$02$   85
          4.5)                                                                $02$   86
           CALL UD0303(LNCT,7)                                                $02$   87
           READ(LOG1,140)XSCALE,PSCALE,RLOW,PLOW,XMMAX,RCONST                 $02$   88
           IF(XMMAX.EQ.0.0)XMMAX=0.6                                          $02$   89
           IF(RCONST.EQ.0.0)RCONST=6.0                                        $02$   90
           WRITE(LOG2,160)XSCALE,PSCALE,RLOW,PLOW,XMMAX,RCONST                $02$   91
    160    FORMAT(2X,/,10X,29HPLOTTING SCALE FOR DIMENSIONS,10X,1H=,F7.3,/,10$02$   92
          1X,28HPLOTTING SCALE FOR PRESSURES,11X,1H=,F7.3,/,10X,22HMINIMUM RA$02$   93
          2DIUS ON PLOT,17X,1H=,F7.3,/,10X,24HMINIMUM PRESSURE ON PLOT,15X,1H$02$   94
          3=,F7.3,/,10X,40HMAXIMUM M-SQUARED IN RELAXATION FACTOR =,F8.4,/,10$02$   95
          4X,29HCONSTANT IN RELAXATION FACTOR,10X,1H=,F8.4)                   $02$   96
           CALL UD0303(LNCT,3)                                                $02$   97
           READ(LOG1,140)CONTR,CONMX                                          $02$   98
           WRITE(LOG2,164)CONTR,CONMX                                         $02$   99
    164    FORMAT(2X,/,10X,22HWAKE TRANSFER CONSTANT,17X,1H=,F8.5,/,10X,25HTU$02$  100
          1RBULENT MIXING CONSTANT,14X,1H=,F8.5)                              $02$  101
           CALL UD0303(LNCT,5+NCASE)                                          $02$  102
           READ(LOG1,170)(FLOW(K),SPDFAC(K),K=1,NCASE)                        $02$  103
    170    FORMAT(2F12.0)                                                     $02$  104
           WRITE(LOG2,180)(K,FLOW(K),SPDFAC(K),K=1,NCASE)                     $02$  105
    180    FORMAT(2X,/,10X,21HPOINTS TO BE COMPUTED,//,10X,2HNO,6X,8HFLOWRATE$02$  106
          1,4X,12HSPEED FACTOR,//,(10X,I2,F13.3,F14.3))                       $02$  107
           READ(LOG1,190)L1,(XSTN(K),RSTN(K),K=1,L1)                          $02$  108
    190    FORMAT(I3,/,(2F12.0))                                              $02$  109
           ISTAG=0                                                            $02$  110
           IF(RSTN(1).EQ.0.0)ISTAG=1                                          $02$  111
           NSPEC(1)=L1                                                        $02$  112
           CALL UD0303(LNCT,7+L1)                                             $02$  113
           WRITE(LOG2,200)L1,(XSTN(K),RSTN(K),K=1,L1)                         $02$  114
```

```
  200 FORMAT(2X,/,10X,36HANNULUS / COMPUTING STATION GEOMETRY,//,10X,24H$02$ 115
     1STATION  1  SPECIFIED BY,I3,7H POINTS,//,17X,4HXSTN,8X,4HRSTN,//,($02$ 116
     2F22.4,F12.4))                                                     $02$ 117
      IS1(1)=1                                                          $02$ 118
      LAST=L1                                                           $02$ 119
      DO 220 I=2,NSTNS                                                  $02$ 120
      READ(LOG1,210)L1                                                  $02$ 121
  210 FORMAT(I3)                                                        $02$ 122
      NEXT=LAST+1                                                       $02$ 123
      LAST=LAST+L1                                                      $02$ 124
      IF(LAST.GT.150)GO TO 550                                          $02$ 125
      READ(LOG1,170)(XSTN(K),RSTN(K),K=NEXT,LAST)                       $02$ 126
      IF(RSTN(NEXT).EQ.0.0)ISTAG=I                                      $02$ 127
      CALL UD0303(LNCT,5+L1)                                            $02$ 128
      IS1(I)=NEXT                                                       $02$ 129
      NSPEC(I)=L1                                                       $02$ 130
  220 WRITE(LOG2,230)I,L1,(XSTN(K),RSTN(K),K=NEXT,LAST)                 $02$ 131
  230 FORMAT(2X,/,10X,7HSTATION,I3,14H  SPECIFIED BY,I3,7H POINTS,//,17X$02$ 132
     1,4HXSTN,8X,4HRSTN,//,(F22.4,F12.4))                               $02$ 133
      SPEED(1)=0.0                                                      $02$ 134
      READ(LOG1,240)L1,NTERP(1),NDIMEN(1),NMACH(1),(DATAC(K),DATA1(K),DAS02$ 135
     1TA2(K),DATA3(K),K=1,L1)                                           $02$ 136
  240 FORMAT(4I3,/,(4F12.0))                                            $02$ 137
      CALL UD0303(LNCT,7+L1)                                            $02$ 138
      IS2(1)=1                                                          $02$ 139
      NDATA(1)=L1                                                       $02$ 140
      LAST=L1                                                           $02$ 141
      WRITE(LOG2,250)L1,NTERP(1),NDIMEN(1),NMACH(1),(DATAC(K),DATA1(K),DS02$ 142
     1ATA2(K),DATA3(K),K=1,L1)                                          $02$ 143
  250 FORMAT(2X,/,10X,24HSTATION CALCULATION DATA,//,7X,18HSTATION  1  NS02$ 144
     1DATA=,I3,7H NTERP=,I2,8H NDIMEN=,I2,7H NMACH=,I2,//,11X,5HDATAC,6X$02$ 145
     2,14HTOTAL PRESSURE,4X,17HTOTAL TEMPERATURE,4X,11HWHIRL ANGLE,//,(5$02$ 146
     3X,F12.4,F15.4,F19.3,F18.3))                                       $02$ 147
      DO 252 K=1,L1                                                     $02$ 148
  252 DATA1(K)=DATA1(K)*SCLFAC**2                                       $02$ 149
      LASTD=0                                                           $02$ 150
      NOUT1(1)=0                                                        $02$ 151
      NOUT2(1)=0                                                        $02$ 152
      DO 320 I=2,NSTNS                                                  $02$ 153
      LOGN=LOG1                                                         $02$ 154
      IF(NMANY.EQ.0)GO TO 258                                           $02$ 155
      DO 254 L1=1,NMANY                                                 $02$ 156
      IF(NWHICH(L1).EQ.I)GO TO 256                                      $02$ 157
  254 CONTINUE                                                          $02$ 158
      GO TO 258                                                         $02$ 159
  256 LOGN=LOG5                                                         $02$ 160
  258 READ(LOGN,260)NDATA(I),NTERP(I),NDIMEN(I),NMACH(I),NWORK(I),NLOSS($02$ 161
     1I),NL1(I),NL2(I),NEVAL(I),NCURVE(I),NLITER(I),NDEL(I),NOUT1(I),NOU$02$ 162
     2T2(I),NOUT3(I),NBLADE(I)                                          $02$ 163
  260 FORMAT(16I3)                                                      $02$ 164
      L1=3                                                              $02$ 165
      IF(NDATA(I).NE.0)L1=L1+5+NDATA(I)                                 $02$ 166
      IF(NDEL(I).NE.0)L1=L1+3+NDEL(I)                                   $02$ 167
      CALL UD0303(LNCT,L1)                                              $02$ 168
      WRITE(LOG2,270)I,NDATA(I),NTERP(I),NDIMEN(I),NMACH(I),NWORK(I),NLO$02$ 169
     1SS(I),NL1(I),NL2(I),NEVAL(I),NCURVE(I),NLITER(I),NDEL(I),NOUT1(I),$02$ 170
     2NOUT2(I),NOUT3(I),NBLADE(I)                                       $02$ 171
```

```
 270    FORMAT(2X,/, 7X,7HSTATION,I3,8H  NDATA=,I3,7H NTERP=,I2,8H NDIMEN=$02$ 172
       1,I2,7H NMACH=,I2,7H NWORK=,I2,7H NLOSS=,I2,5H NL1=,I3,5H NL2=,I3,7$02$ 173
       2H NEVAL=,I2,8H NCURVE=,I2,8H NLITER=,I3,6H NDEL=,I3,/,19X,6HNOUT1=$02$ 174
       3,I2,7H NOUT2=,I2,7H NOUT3=,I2,8H NBLADE=,I3)                      $02$ 175
        SPEED(I)=0.0                                                       $02$ 176
        IF(NDATA(I).EQ.0)GO TO 320                                         $02$ 177
        IF(NWORK(I).EQ.7)NLOSS(I)=1                                        11/30/77
        NEXT=LAST+1                                                        $02$ 178
        LAST=LAST+NDATA(I)                                                 $02$ 179
        IS2(I)=NEXT                                                        $02$ 180
        IF(LAST.GT.100)GO TO 550                                           $02$ 181
        READ(LOGN,280)SPEED(I),(DATAC(K),DATA1(K),DATA2(K),DATA3(K),DATA4($02$ 182
       1K),DATA5(K),DATA6(K),DATA7(K),DATA8(K),DATA9(K),K=NEXT,LAST)      $02$ 183
 280    FORMAT(F12.0,/,(6F12.0,/,4F12.0))                                  $02$ 184
        WRITE(LOG2,290)SPEED(I),(DATAC(K),DATA1(K),DATA2(K),DATA3(K),DATA4$02$ 185
       1(K),DATA5(K),DATA6(K),DATA7(K),DATA8(K),DATA9(K),K=NEXT,LAST)     $02$ 186
 290    FORMAT(2X,/,10X,7HSPEED =,F9.2,//,13X,5HDATAC,7X,5HDATA1,7X,5HDATA$02$ 187
       12,7X,5HDATA3,7X,5HDATA4,7X,5HDATA5,7X,5HDATA6,7X,5HDATA7,7X,5HDATA$02$ 188
       28,7X,5HDATA9,//,(10X,F9.4,F12.3,F13.6,F11.4,F12.5,F12.5,4F12.4)) $02$ 189
        IF(NLOSS(I).NE.1)GO TO 293                                         MOD.-JGW
        KPRESS=0                                                           MOD.-JGW
        DO 291 K=NEXT,LAST                                                 MOD.-JGW
  291   IF(ABS(DATA2(K)).GT.2.0)KPRESS=KPRESS+1                            MOD.-JGW
        KCHECK=LAST-NEXT+1                                                 MOD.-JGW
        IF(KPRESS.NE.KCHECK)GO TO 293                                      MOD.-JGW
        DO 292 K=NEXT,LAST                                                 MOD.-JGW
  292   DATA2(K)=DATA2(K)*SCLFAC**2                                        MOD.-JGW
  293   IF(NWORK(I).NE.1)GO TO 296                                         MOD.-JGW
        DO 294 K=NEXT,LAST                                                 $02$ 191
 294    DATA1(K)=DATA1(K)*SCLFAC**2                                        $02$ 192
 296    IF(NEVAL(I).GT.0.AND.NSTRMS.GT.NDATA(I))LAST=LAST+NSTRMS-NDATA(I) $02$ 193
        IF(NDEL(I).EQ.0)GO TO 320                                          $02$ 194
        NEXT=LASTD+1                                                       $02$ 195
        LASTD=LASTD+NDEL(I)                                                $02$ 196
        IS3(I)=NEXT                                                        $02$ 197
        IF(LASTD.GT.100)GO TO 550                                          $02$ 198
        READ(LOG1,300)(DELC(K),DELTA(K),K=NEXT,LASTD)                      $02$ 199
 300    FORMAT(2F12.0)                                                     $02$ 200
        WRITE(LOG2,310)(DELC(K),DELTA(K),K=NEXT,LASTD)                     $02$ 201
 310    FORMAT(2X,/,13X,4HDELC,8X,5HDELTA,//,(10X,F9.4,F12.4))             $02$ 202
 320    CONTINUE                                                           $02$ 203
        CALL UD0303(LNCT,5+NSTNS)                                          $02$ 204
        READ(LOG1,330)(WBLOCK(I),BBLOCK(I),BDIST(I),I=1,NSTNS)             $02$ 205
 330    FORMAT(3F12.0)                                                     $02$ 206
        WRITE(LOG2,340)(I,WBLOCK(I),BBLOCK(I),BDIST(I),I=1,NSTNS)          $02$ 207
 340    FORMAT(2X,/,10X,30HBLOCKAGE FACTOR SPECIFICATIONS,//,10X,66HSTATIO$02$ 208
       1N   WALL BLOCKAGE   WAKE BLOCKAGE   WAKE DISTRIBUTION FACTOR,//,(1$02$ 209
       20X,I4,F16.5,F16.5,F19.3))                                          $02$ 210
        IF(NSET1.EQ.0)GO TO 380                                            $02$ 211
        DO 370 K=1,NSET1                                                   $02$ 212
        READ(LOG1,350)L1,(DIFF(J,K),FDHUB(J,K),FDMID(J,K),FDTIP(J,K),J=1,L$02$ 213
       11)                                                                 $02$ 214
 350    FORMAT(I3,/,(4F12.0))                                              $02$ 215
        CALL UD0303(LNCT,6+L1)                                             $02$ 216
        WRITE(LOG2,360)K,L1,(DIFF(J,K),FDHUB(J,K),FDMID(J,K),FDTIP(J,K),J=$02$ 217
       11,L1)                                                              $02$ 218
 360    FORMAT(2X,/,10X,55HLOSS PARAMETER / DIFFUSION FACTOR CURVES FOR BL$02$ 219
```

```
      1ADE TYPE,I2,I5,16H D-FACTORS GIVEN,//,15X,9HDIFFUSION,5X,30HL O S $02$ 220
      2S   P A R A M E T E R S,/,16X,7HFACTORS,8X,3HHUB,9X,3HMID,8X,3HTIP$02$ 221
      3,//,(15X,F6.3,F13.5,F12.5,F11.5))                                $02$ 222
 370  NDIFF(K)=L1                                                       $02$ 223
 380  IF(NSET2.EQ.0)GO TO 450                                           $02$ 224
      DO 440 K=1,NSET2                                                  $02$ 225
      READ(LOG1,390)L1,L2                                               $02$ 226
 390  FORMAT(2I3)                                                       $02$ 227
      CALL UD0303(LNCT,7+L1)                                            $02$ 228
      NM(K)=L1                                                          $02$ 229
      NRAD(K)=L2                                                        $02$ 230
      READ(LOG1,400)TERAD(1,K),(DM(J,1,K),WFRAC(J,1,K),J=1,L1)          $02$ 231
 400  FORMAT(F12.0,/,(2F12.0))                                          $02$ 232
      WRITE(LOG2,410)K,L1,L2,TERAD(1,K),(DM(J,1,K),WFRAC(J,1,K),J=1,L1) $02$ 233
 410  FORMAT(2X,/,10X,51HFRACTIONAL LOSS DISTRIBUTION CURVES FOR BLADE C$02$ 234
      1LASS,I2,I5,16H POINTS GIVEN AT,I3,17H RADIAL LOCATIONS,//,10X,52HF$02$ 235
      2RACTION OF COMPUTING STATION LENGTH AT BLADE EXIT =,F7.4,//,10X,28$02$ 236
      3HFRACTION OF MERIDIONAL CHORD,4X,26HLOSS/LOSS AT TRAILING EDGE,//,$02$ 237
      4(15X,F11.4,20X,F11.4))                                           $02$ 238
      IF(L2.EQ.1)GO TO 440                                              $02$ 239
      DO 420 L=2,L2                                                     $02$ 240
      CALL UD0303(LNCT,5+L1)                                            $02$ 241
      READ(LOG1,400)TERAD(L,K),(DM(J,L,K),WFRAC(J,L,K),J=1,L1)          $02$ 242
 420  WRITE(LOG2,430)TERAD(L,K),(DM(J,L,K),WFRAC(J,L,K),J=1,L1)         $02$ 243
 430  FORMAT(2X,/,10X,52HFRACTION OF COMPUTING STATION LENGTH AT BLADE E$02$ 244
      1XIT =,F7.4,//,10X,28HFRACTION OF MERIDIONAL CHORD,4X,26HLOSS/LOSS $02$ 245
      2AT TRAILING EDGE,//,(15X,F11.4,20X,F11.4))                       $02$ 246
 440  CONTINUE                                                          $02$ 247
 450  IF(NSPLIT.EQ.0.AND.NREAD.EQ.0)GO TO 570                           $02$ 248
      READ(LOG1,460)(DELF(J),J=1,NSTRMS)                                $02$ 249
 460  FORMAT(6F12.0)                                                    $02$ 250
      L1=5                                                              $02$ 251
      IF(NSTRMS.GE.16)L1=8                                              $02$ 252
      CALL UD0303(LNCT,L1)                                              $02$ 253
      WRITE(LOG2,470)                                                   $02$ 254
      L1=NSTRMS                                                         $02$ 255
      IF(NSTRMS.GT.15)L1=15                                             $02$ 256
      WRITE(LOG2,480)(J,J=1,L1)                                         $02$ 257
 480  FORMAT(2X,/,10X,10HSTREAMLINE,I5,14I7)                            $02$ 258
 470  FORMAT(2X,/,10X,78HPROPORTIONS OF TOTAL FLOW BETWEEN HUB AND EACH $02$ 259
      1STREAMLINE ARE TO BE AS FOLLOWS)                                 $02$ 260
      WRITE(LOG2,490)(DELF(J),J=1,L1)                                   $02$ 261
 490  FORMAT(10X,4HFLOW,7X,15F7.4)                                      $02$ 262
      IF(NSTRMS.LE.15)GO TO 500                                         $02$ 263
      L1=L1+1                                                           $02$ 264
      WRITE(LOG2,480)(J,J=L1,NSTRMS)                                    $02$ 265
      WRITE(LOG2,490)(DELF(J),J=L1,NSTRMS)                              $02$ 266
 500  IF(NREAD.EQ.0)GO TO 570                                           $02$ 267
      READ(LOG1,510)((R(J,I),X(J,I),XL(J,I),II(J,I),JJ(J,I),J=1,NSTRMS),$02$ 268
      1I=1,NSTNS)                                                       $02$ 269
 510  FORMAT(3F12.0,2I3)                                                $02$ 270
      CALL UD0303(LNCT,5+NSTRMS)                                        $02$ 271
      WRITE(LOG2,520)                                                   $02$ 272
 520  FORMAT(2X,/,10X,32HESTIMATED STREAMLINE COORDINATES)              $02$ 273
      DO 530 I=1,NSTNS                                                  $02$ 274
      IF(I.GT.1)CALL UD0303(LNCT,3+NSTRMS)                              $02$ 275
 530  WRITE(LOG2,540)(I,J,R(J,I),X(J,I),XL(J,I),II(J,I),JJ(J,I),J=1,NSTR$02$ 276
```

```
                1MS)                                                                 $02$ 277
        540      FORMAT(2X,/,10X,79HSTATION  STREAMLINE   RADIUS  AXIAL COORDINATE  $02$ 278
                1 L-COORDINATE     CHECKS-  I     J,//,(3X,2I11,F14.4,F12.4,F16.4,I17$02$ 279
                2,I5))                                                               $02$ 280
                 GO TO 570                                                           $02$ 281
        550      WRITE(LOG2,560)                                                     $02$ 282
        560      FORMAT(1H1,10X,33HJOB STOPPED - TOO MUCH INPUT DATA)                $02$ 283
                 STOP                                                                $02$ 284
        570      RETURN                                                              $02$ 285
                 END                                                                 $02$ 286
```

Source listing of the 1964 - 65 Version 5 CALCOMP subroutine, UD0312, appears below.

```
      SUBROUTINE UD0312                                                 $12$   2
      REAL LOSS,LAMI,LAMIP1,LAMIM1                                      $12$   3
      COMMON NSTNS,NSTRMS,NMAX,NFORCE,NBL,NCASE,NSPLIT,NREAD,NPUNCH,NPAG$12$   4
     1E,NSET1,NSET2,ISTAG,ICASE,IFAILO,IPASS,I,IVFAIL,IFFAIL,NMIX,NTRANS$12$   5
     2,NPLOT,ILOSS,LNCT,ITUB,IMID,IFAIL,ITER,LOG1,LOG2,LOG3,LOG4,LOG5,LO$12$   6
     3G6,IPRINT,NMANY,NSTPLT,NEQN                                       $12$   7
      COMMON NSPEC(30),NWORK(30),NLOSS(30),NDATA(30),NTERP(30),NMACH(30)$12$   8
     1,NL1(30),NL2(30),NDIMEN(30),IS1(30),IS2(30),IS3(30),NEVAL(30),NDIF$12$   9
     2F(4),NDEL(30),NLITER(30),NM(2),NRAD(2),NCURVE(30),NWHICH(30),NOUT1$12$  10
     3(30),NOUT2(30),NOUT3(30),NBLADE(30)                               $12$  11
      COMMON DM(11,5,2),WFRAC(11,5,2)                                   $12$  12
      COMMON R(21,30),XL(21,30),X(21,30),H(21,30),S(21,30),VM(21,30),VW($12$  13
     121,30),TBETA(21,30),DIFF(15,4),FDHUB(15,4),FDMID(15,4),FDTIP(15,4)$12$  14
     2,TERAD(5,2)                                                       $12$  15
      COMMON DATAC(100),DATA1(100),DATA2(100),DATA3(100),DATA4(100),DATA$12$  16
     15(100),DATA6(100),DATA7(100),DATA8(100),DATA9(100),FLOW(10),SPEED($12$  17
     230),SPDFAC(10),RBLOCK(30),BDIST(30),WBLOCK(30),WWBL(30),XSTN(150),$12$  18
     3RSTN(150),DELF(30),DELC(100),DELTA(100),TITLE(18),DRDM2(30),RIM1(3$12$  19
     40),XIM1(30),WORK(21),LOSS(21),TANEPS(21),XI(21),VV(21),DELW(21),LA$12$  20
     5MI(21),LAMIM1(21),LAMIP1(21),PHI(21),CR(21),GAMA(21),SPPG(21),CPPG$12$  21
     6(21),HKEEP(21),SKEEP(21),VWKEEP(21),DELH(30),DELT(30)             $12$  22
      COMMON VISK,SHAPE,SCLFAC,EJ,G,TOLNCE,XSCALE,PSCALE,PLOW,RLOW,XMMAX$12$  23
     1,RCONST,FM2,HMIN,C1,PI,CONTR,CONMX                                $12$  24
      DIMENSION PSTAT(32),XX(32)                                        $12$  25
      XMAX=X(1,NSTNS)                                                   $12$  26
      XMIN=X(1,1)                                                       $12$  27
      DO 100 J=2,NSTRMS                                                 $12$  28
      IF(X(J,1).LT.XMIN)XMIN=X(J,1)                                     $12$  29
      IF(X(J,NSTNS).GT.XMAX)XMAX=X(J,NSTNS)                             $12$  30
  100 CONTINUE                                                          $12$  31
      XMIN=FLOAT(IFIX(XMIN))                                            MOD.-JGW
      IF((XMIN.GE.0.0).AND.(XMIN.LE.5.0))XMIN=0.0                       MOD.-JGW
      ALEN=(XMAX-XMIN)/XSCALE                                           MOD.-JGW
      XLEN=20.0                                                         MOD.-JGW
      IF(ALEN.LE.10.0)XLEN=10.0                                         MOD.-JGW
      IF(NPLOT.EQ.2)GO TO 134                                           $12$  42
      CALL PLOTID                                                       MOD.-JGW
      CALL PLOT(0.0,-12.0,-3)                                           MOD.-JGW
      CALL PLOT(0.0,0.5,-3)                                             MOD.-JGW
      CALL AXIS(0.0,0.0,16HAXIAL COORDINATE,-16,XLEN,0.0,XMIN,XSCALE,10.MOD.-JGW
     10)                                                                MOD.-JGW
      CALL AXIS(0.0,0.0,15HSTATIC PRESSURE,15,10.0,90.0,PLOW,PSCALE,10.0MOD.-JGW
     1)                                                                 MOD.-JGW
      J=1                                                               $12$  47
      K=1                                                               $12$  48
      YPEN=3.4                                                          MOD.-JGW
      XPEN=(XMAX/XSCALE)-3.0                                            MOD.-JGW
      SIZE=0.14                                                         MOD.-JGW
  110 DO 120 I=1,NSTNS                                                  $12$  49
      HS=H(J,I)-(VW(J,I)**2+VM(J,I)**2)/(2.0*G*EJ)                      $12$  50
      IF(HS.LT.HMIN)HS=HMIN                                             $12$  51
      PSTAT(I)=UDG4(HS,S(J,I))/SCLFAC**2                                $12$  52
  120 XX(I)=X(J,I)                                                      $12$  53
      CALL LINE(XX,PSTAT,NSTNS,1,1,K,XMIN,XSCALE,PLOW,PSCALE)           MOD.-JGW
      CALL SYMBOL(XPEN,YPEN+0.07,SIZE,K,0.0,-1)                         MOD.-JGW
      IF(J.EQ.NSTRMS)CALL SYMBOL(XPEN+0.14,YPEN,SIZE,17H=> TIP STREAMLINMOD.-JGW
```

```
          1E,0.0,17)                                                    MOD.-JGW
          IF(J.EQ.NSTRMS)GO TO 130                                      $12$  55
          K=K+1                                                         $12$  56
          IF(J.EQ.IMID)J=NSTRMS                                         $12$  57
          IF(J.EQ.NSTRMS)CALL SYMBOL(XPEN+0.14,YPEN,SIZE,17H=> MID STREAMLINMOD.-JGW
          1E,0.0,17)                                                    MOD.-JGW
          IF(J.EQ.1)J=IMID                                              $12$  58
          IF(J.EQ.IMID)CALL SYMBOL(XPEN+0.14,YPEN,SIZE,17H=> HUB STREAMLINE,MOD.-JGW
          10.0,17)                                                      MOD.-JGW
          YPEN=YPEN-0.25                                                MOD.-JGW
          GO TO 110                                                     $12$  59
      130 CALL PLOT(25.0,-12.0,-3)                                      MOD.-JGW
          IF(NPLOT.EQ.1)GO TO 180                                       $12$  62
      134 CALL PLOTID                                                   MOD.-JGW
          CALL PLOT(0.0,-12.0,-3)                                       MOD.-JGW
          CALL PLOT(0.0,0.5,-3)                                         MOD.-JGW
          CALL AXIS(0.0,0.0,16HAXIAL COORDINATE,-16,XLEN,0.0,XMIN,XSCALE,10.MOD.-JGW
          10)                                                           MOD.-JGW
          CALL AXIS(0.0,0.0,6HRADIUS,6,10.0,90.0,RLOW,XSCALE,10.0)      MOD.-JGW
          DO 150 J=1,NSTRMS                                             $12$  67
          DO 140 I=1,NSTNS                                              $12$  68
          XX(I)=X(J,I)                                                  $12$  69
      140 PSTAT(I)=R(J,I)                                               $12$  70
      150 CALL LINE(XX,PSTAT,NSTNS,1,0,11,XMIN,XSCALE,RLOW,XSCALE)      MOD.-JGW
          DO 170 I=1,NSTNS                                              $12$  76
          DO 160 J=1,NSTRMS                                             $12$  77
          PSTAT(J)=R(J,I)                                               $12$  78
      160 XX(J)=X(J,I)                                                  $12$  79
      170 CALL LINE(XX,PSTAT,NSTRMS,1,0,11,XMIN,XSCALE,RLOW,XSCALE)     MOD.-JGW
          CALL PLOT(25.0,-12.0,-3)                                      MOD.-JGW
      180 CALL ENDCC                                                    MOD.-JGW
          RETURN                                                        MOD.-JGW
          END                                                           $12$  83
```

REFERENCES

1. Smith, L. H., Jr.: "The Radial-Equilibrium Equation of Tur- bomachinery", J. Eng. Power, vol. 88, no. 1, Jan. 1966, pp. 1-12.

2. Dixon, S. L.: Fluid Mechanics, Thermodynamics of Turbomachinery, Pergamon Press, Inc., 1975, pp. 171-181.

3. Novak, R. A.: Streamline Curvature Computing Procedures f r Fluid Flow Problems. J. Eng. Power, vol. 89, no. 4, Oct. 1967, pp. 478-490.

4. Hearsey, Richard M.: A Revised Computer Program for Axial Compressor Design. Vol. I. Theory, Descriptions, and User's Instructions. UDRI-TR-47-Vol-1, Dayton Univ., 1975. (ARL-75-0001-Vol-1, AD-A009273)

5. Hearsey, Richard M.: A Revised Computer Program for Axial Compressor Design. Vol. II. Program Listing and Program Use Example. UDRI-TR-47-Vol-2, Dayton Univ., 1975. (ARL-75-0001-Vol-2, AD-A009157)

6. Pourier, Philip: Private Communication, March 19, 1979.

7. Dudzinski, Thomas J.; Krause, Lloyd N.: Flow Direction Measurement with Fixed-Position Probes, NASA TM X-1904, 1969.

8. Abbott, Ira H.; von Doenhoff, Albert E.; Stivers, Louis S., Jr.: Summary of Airfoil Data. NACA TR-824, 1945.

9. Jeffers, J. D.; May, A.; Deskin, W. .: Evaluation of a Technique for Predicting Stall Flutter in Turbine Engines, (PWA-FR-9620, Pratt & Whitney Aircraft Group; NASA Contract NAS3-19437.) NASA CR-135423, 1978.

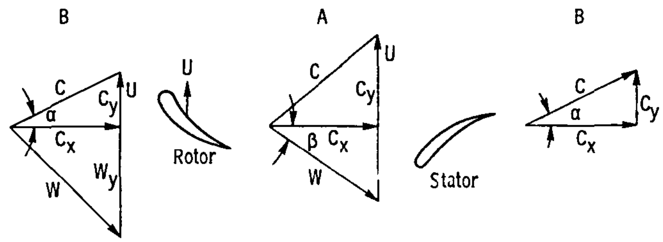


Figure 1. - Two-dimensional velocity triangles for an axial compressor.

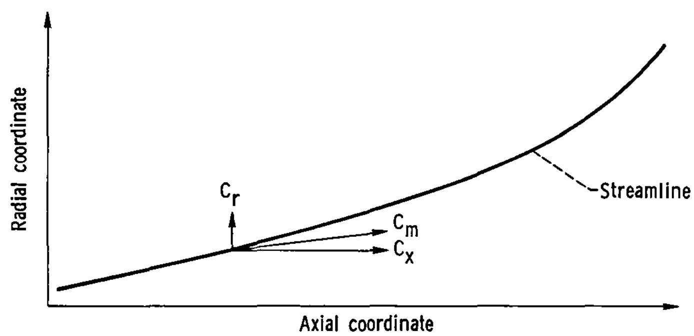


Figure 2. - Definition of meridional direction.

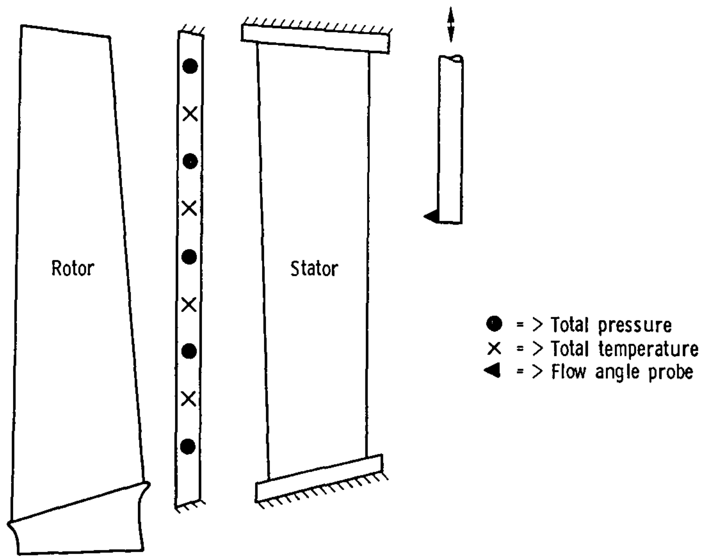


Figure 3. - Typical cascade instrumentation. The flow angle probe is capable of traversing radially. Total pressure may be obtained from flow angle probe.

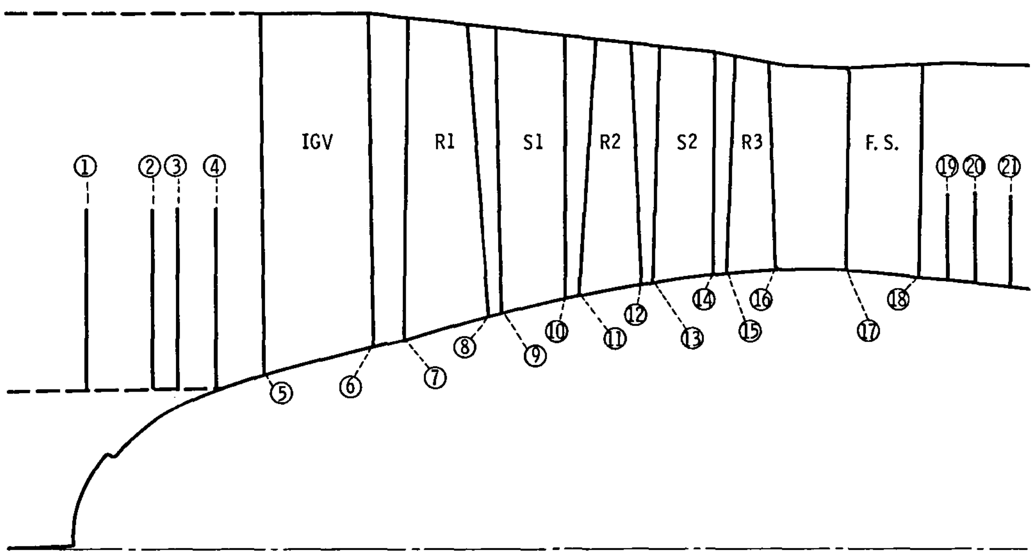


Figure 4. - Fan module axial station locations.

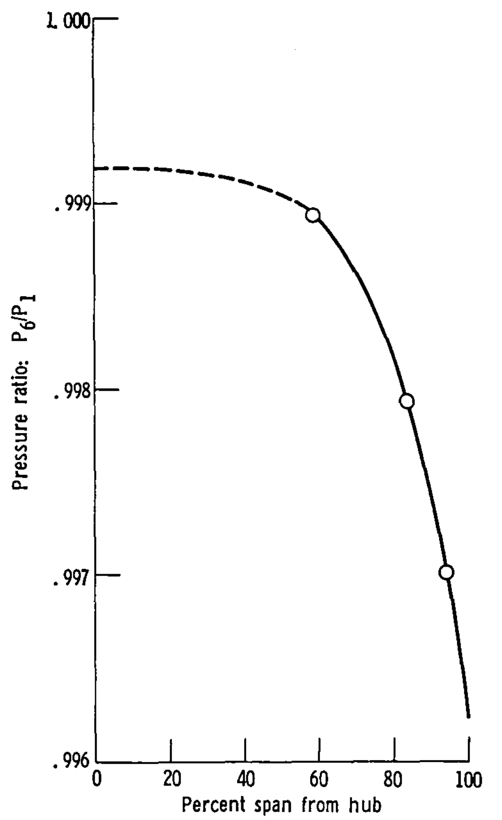


Figure 5. - Total pressure profile behind Inlet Guide Vane (Station 6). Data comes from flow angle probe. P_1 = 136.907 kPa (19.856 psia).

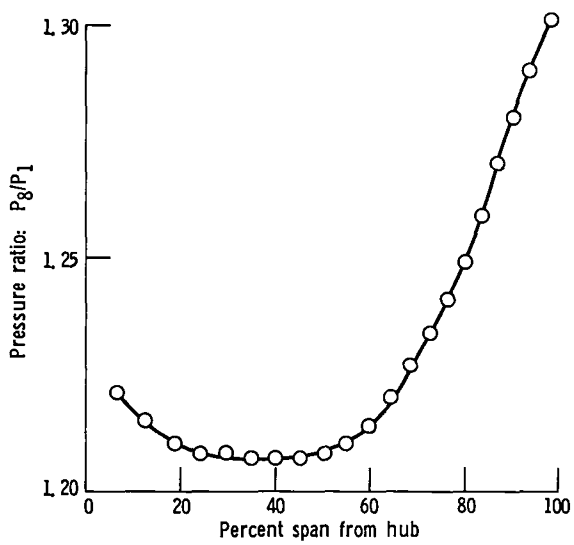


Figure 6. - Total pressure profile behind first rotor (Station 8). P_1 = 136.907 kPa (19.856 psia).

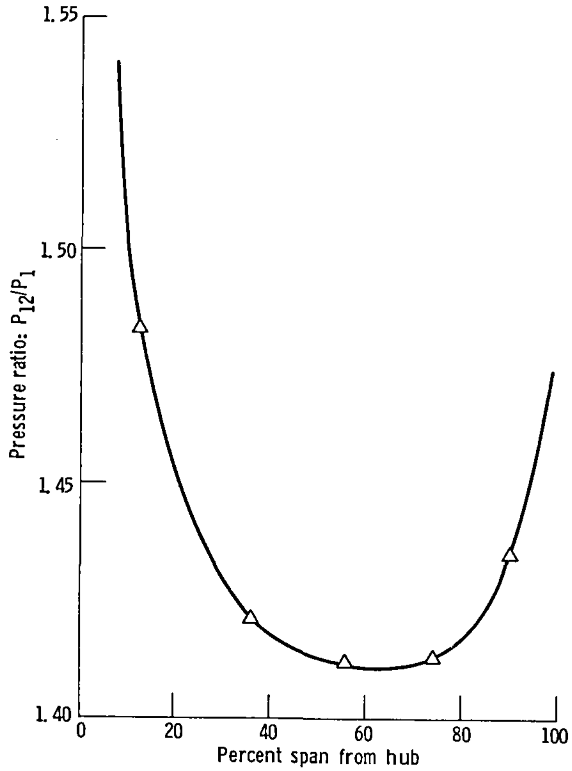


Figure 7. - Total pressure profile behind second rotor (Station 12). P_1 = 136.907 kPa (19.856 psia).

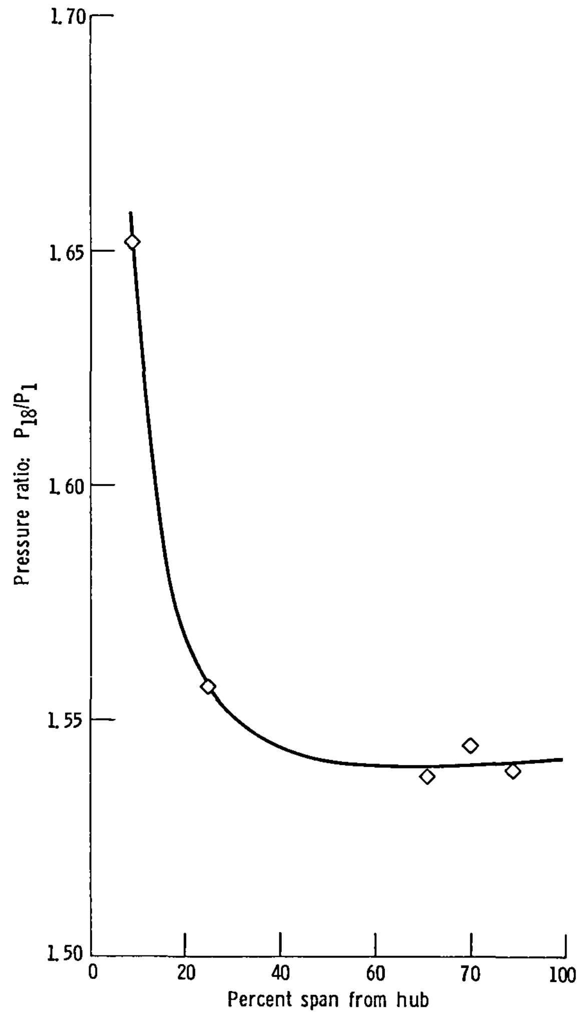


Figure 8. - Total pressure profile behind flow straightener (Station 18). P_1 = 136.907 kPa (19.856 psia).

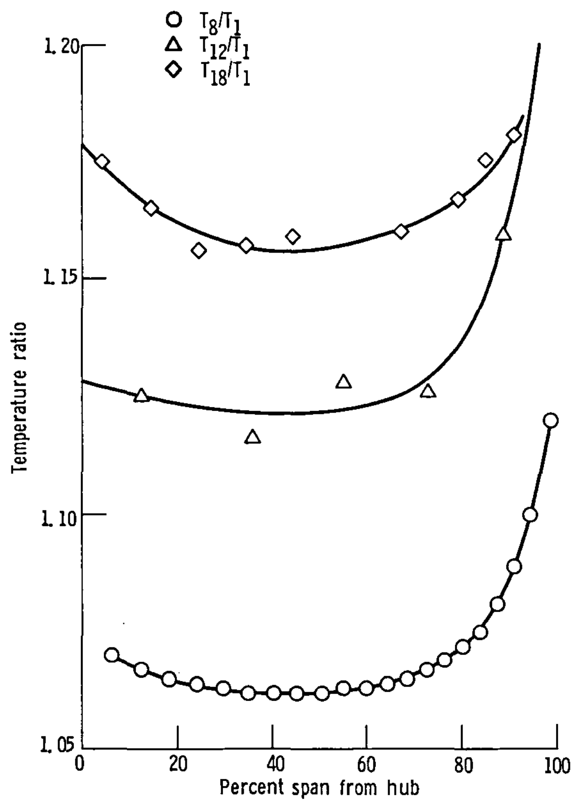
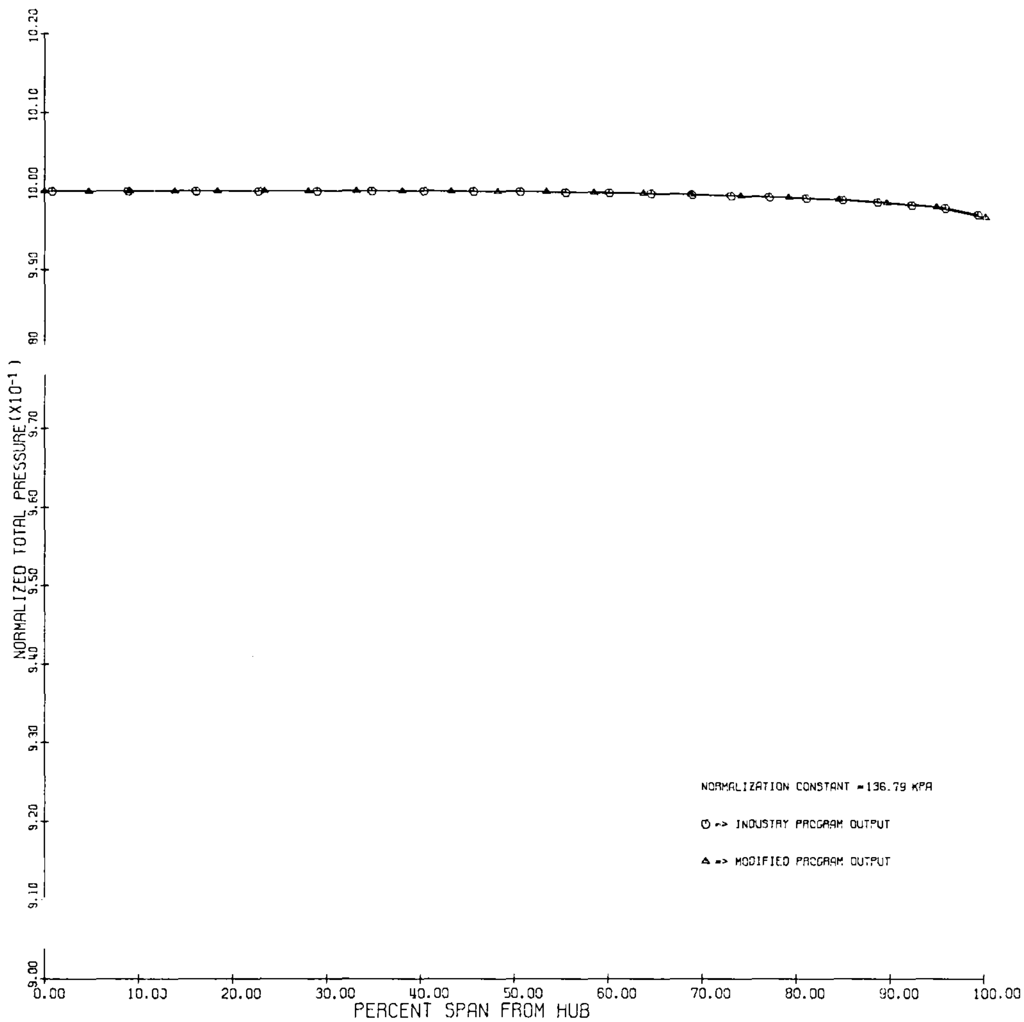
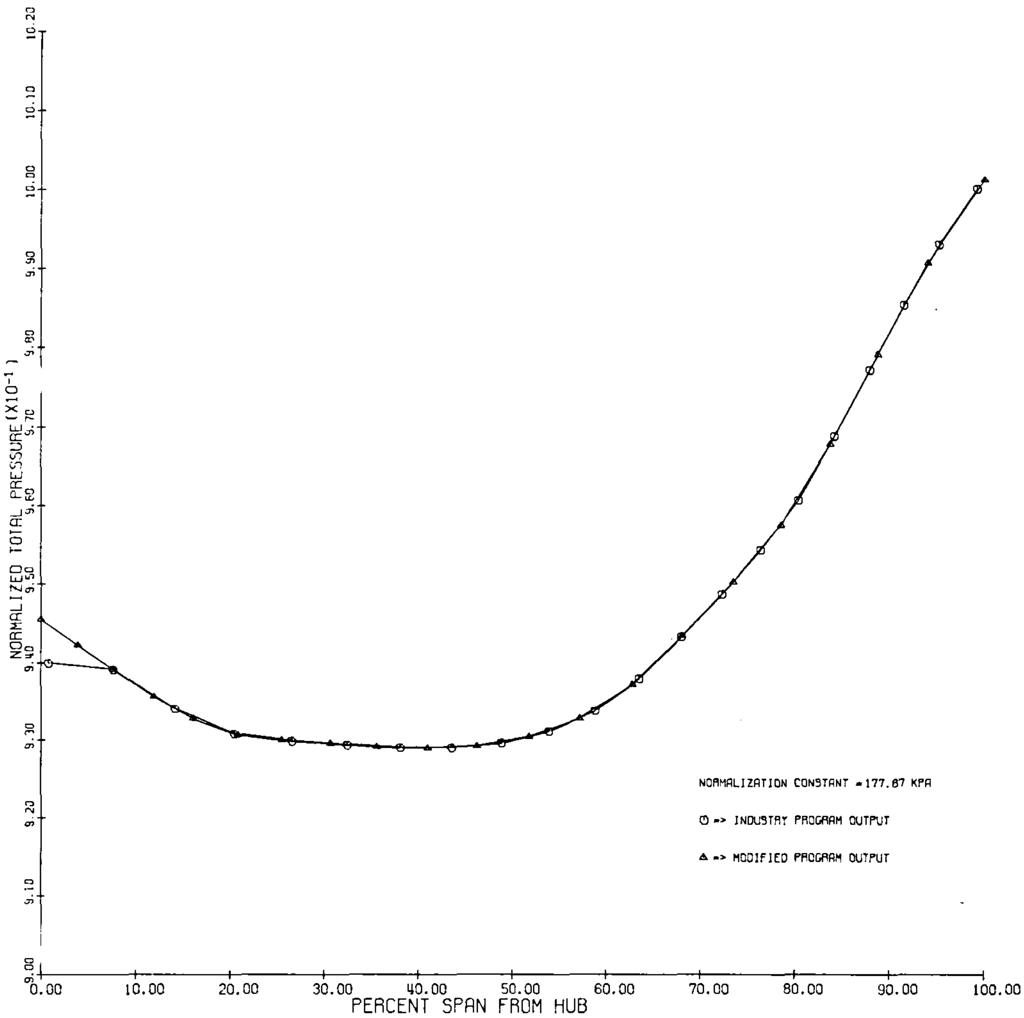


Figure 9. - Temperature profiles at axial stations. T_1 = 442.82K (797.08R).



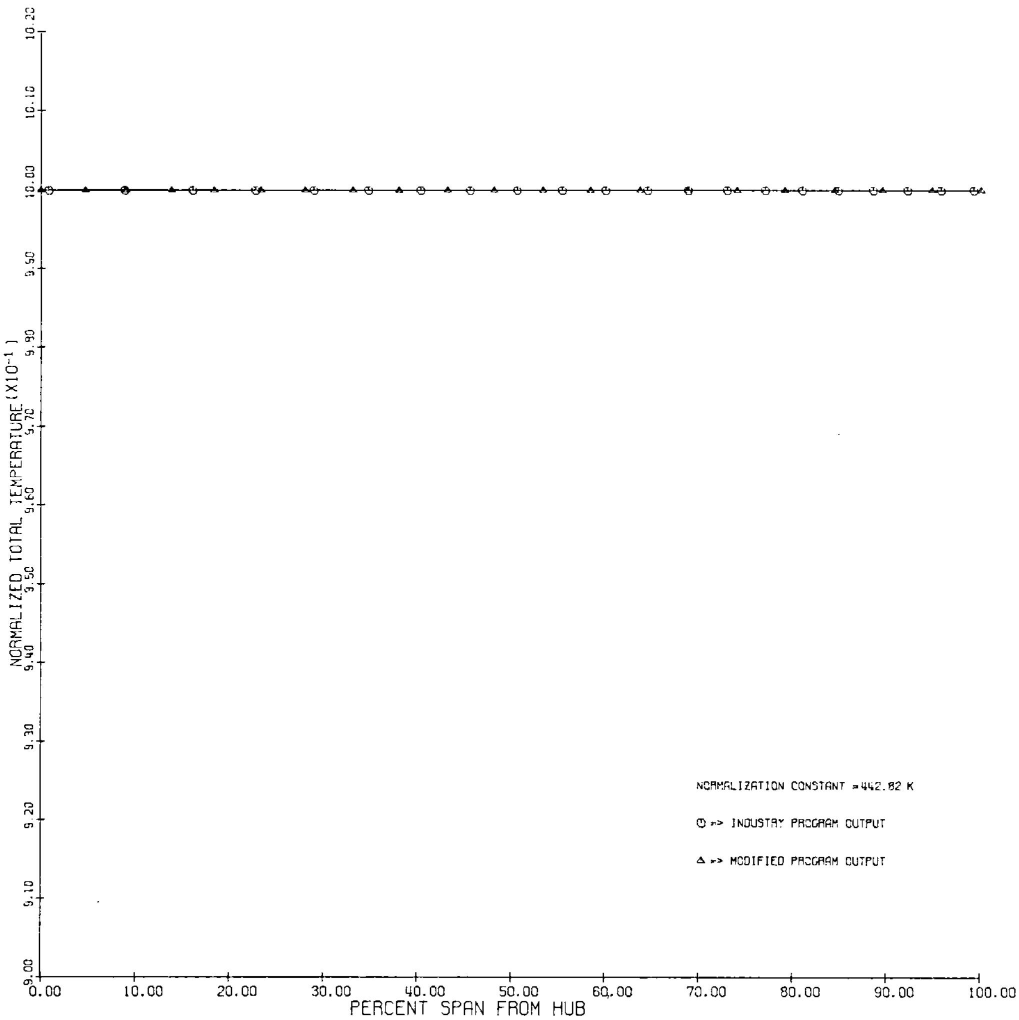
(a) Station 7.

Figure 10. - Comparison of program total pressure output.



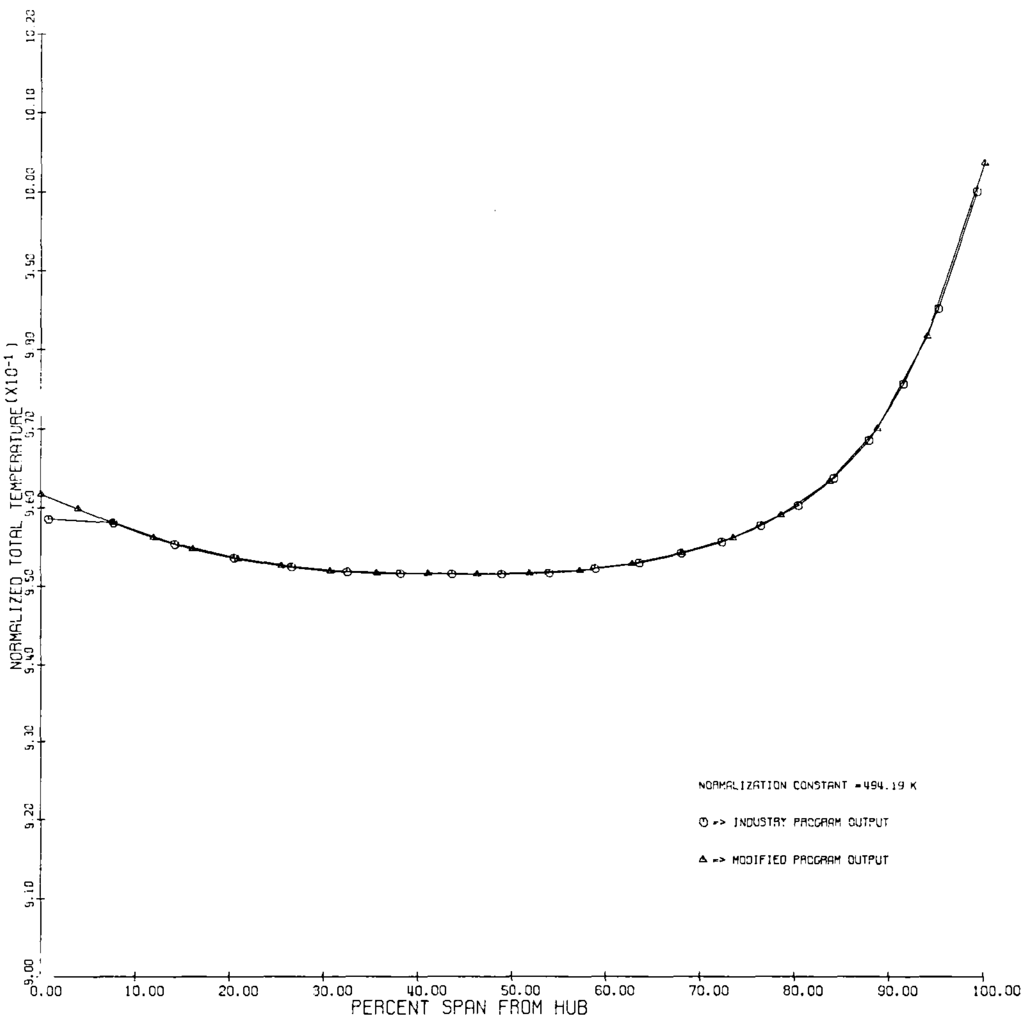
(b) Station 8.

Figure 10. - Concluded.



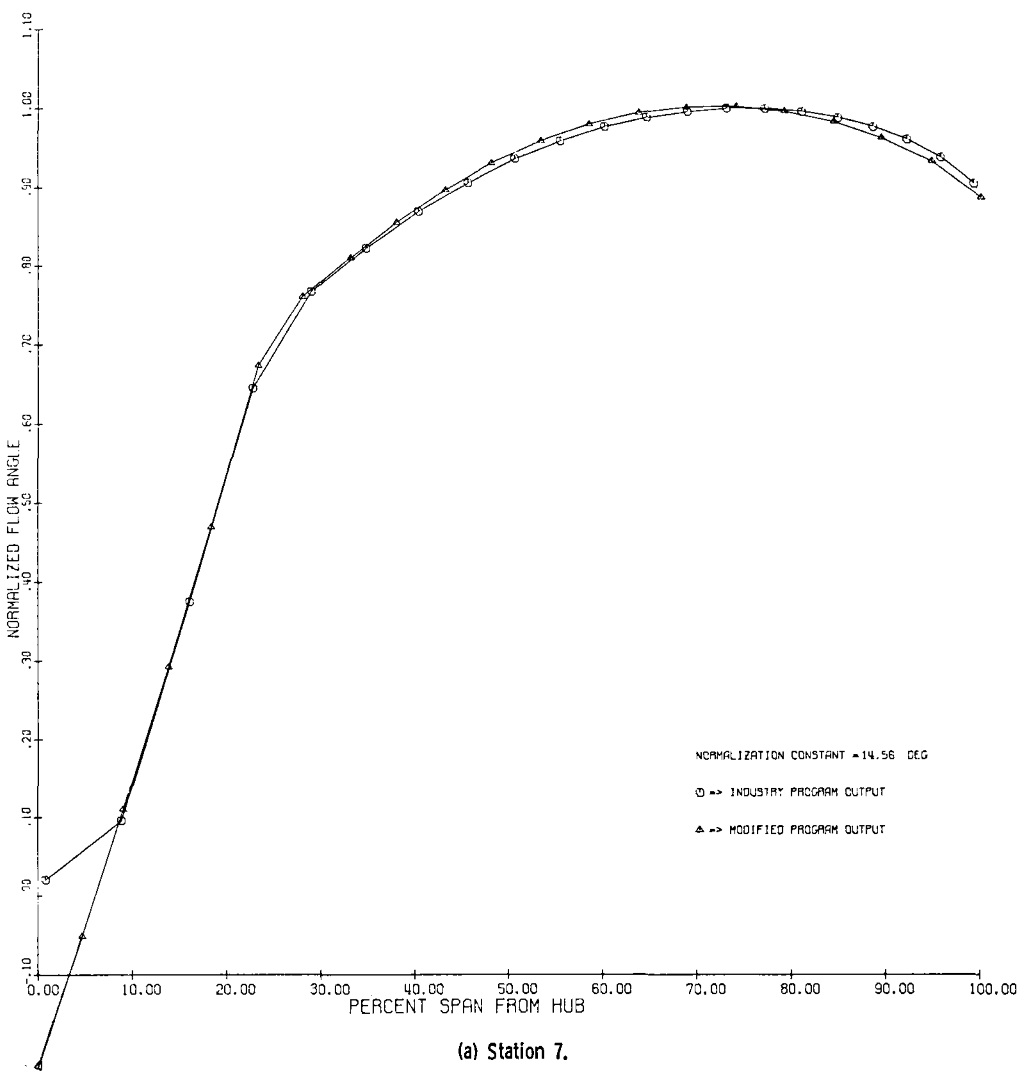
(a) Station 7.

Figure 11. - Comparison of program total temperature output.



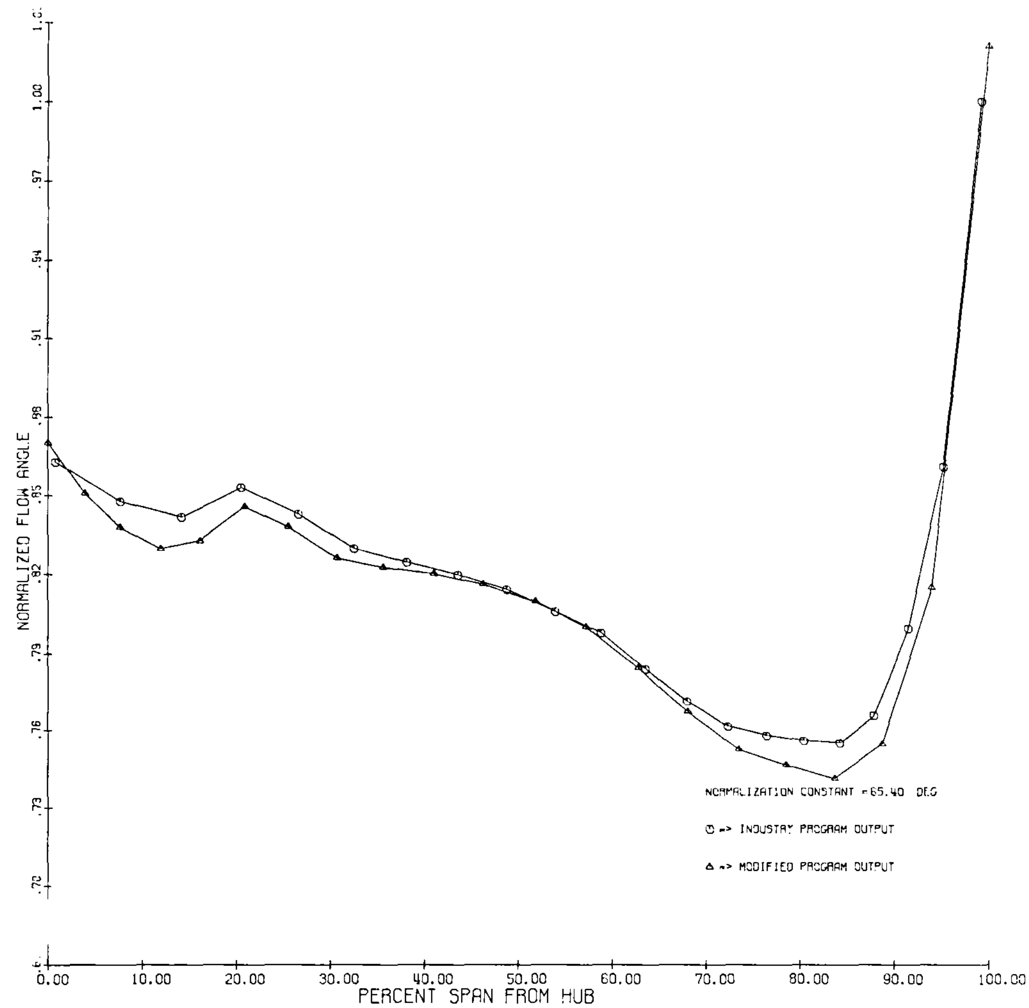
(b) Station 8.

Figure 11. - Concluded.



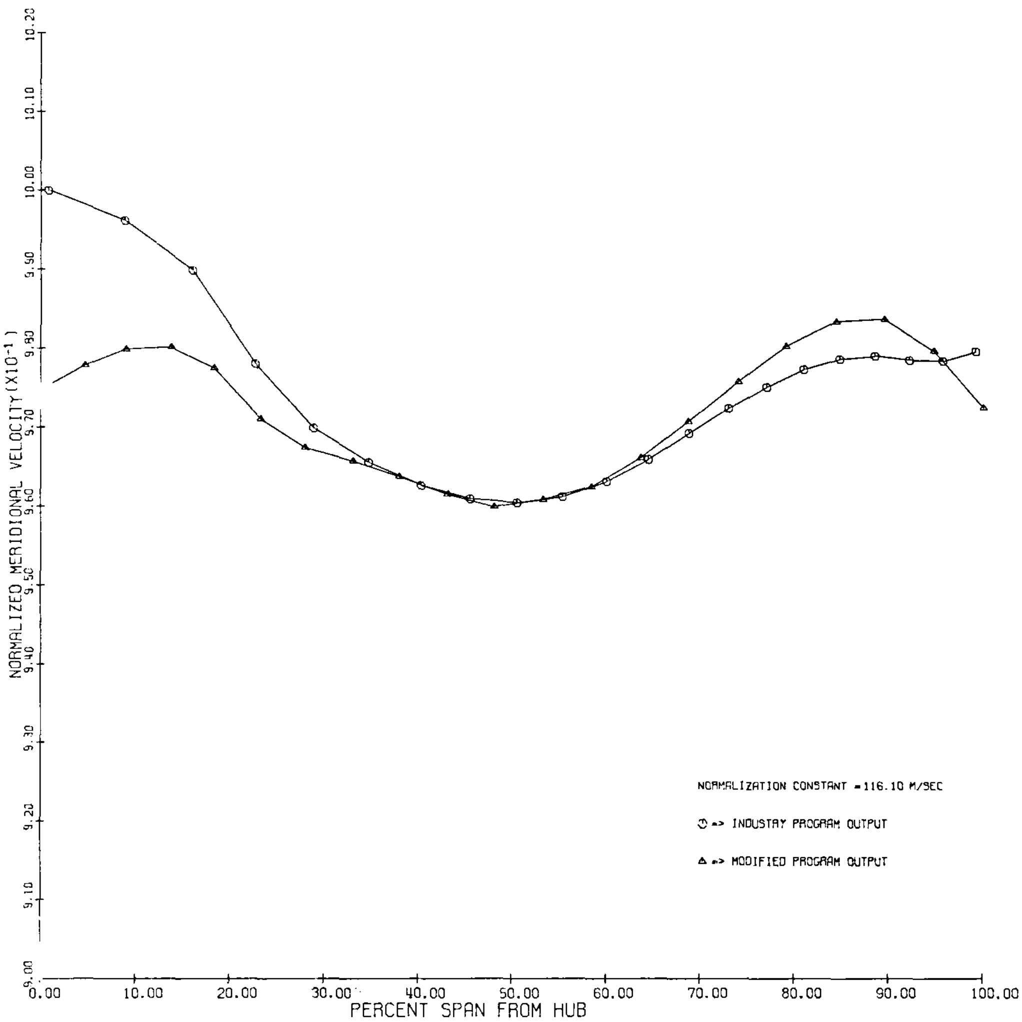
(a) Station 7.

Figure 12. - Comparison of program flow angle output.



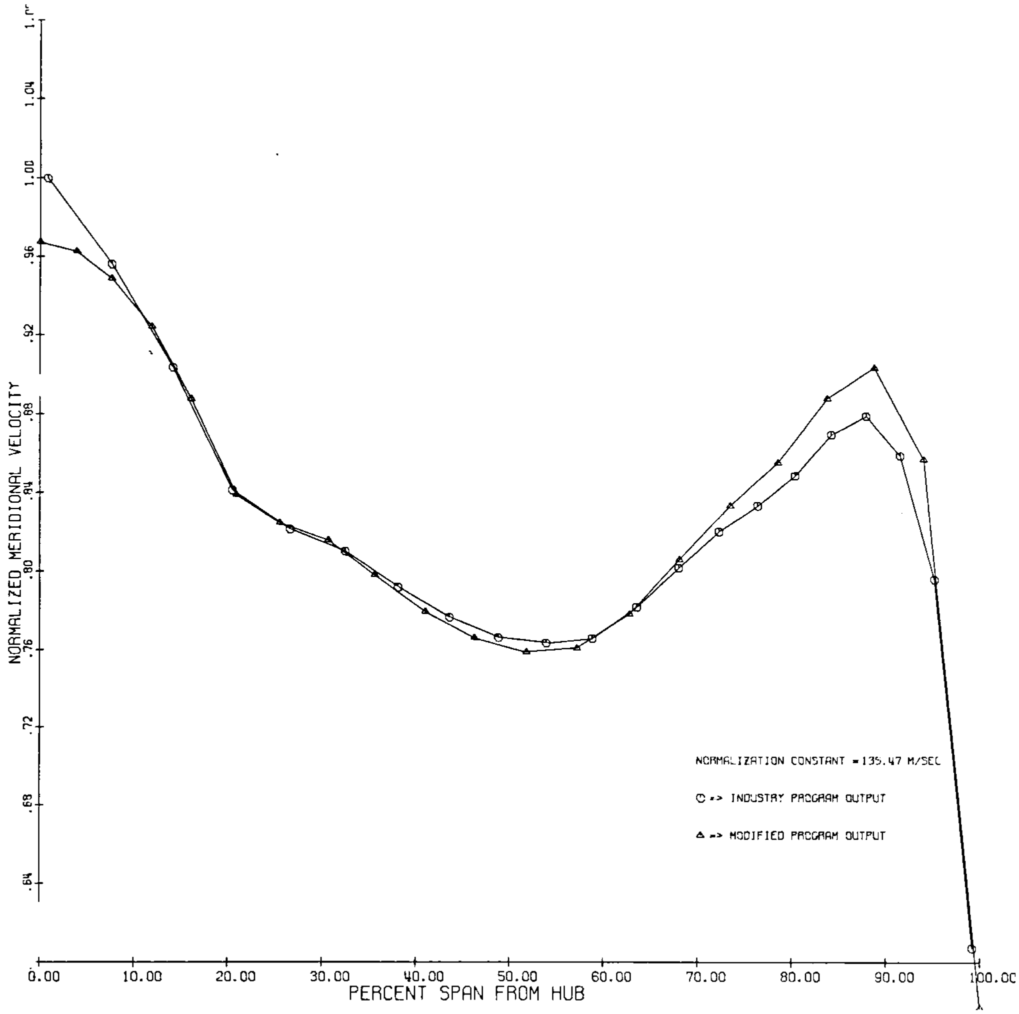
(b) Station 8.

Figure 12. - Concluded.



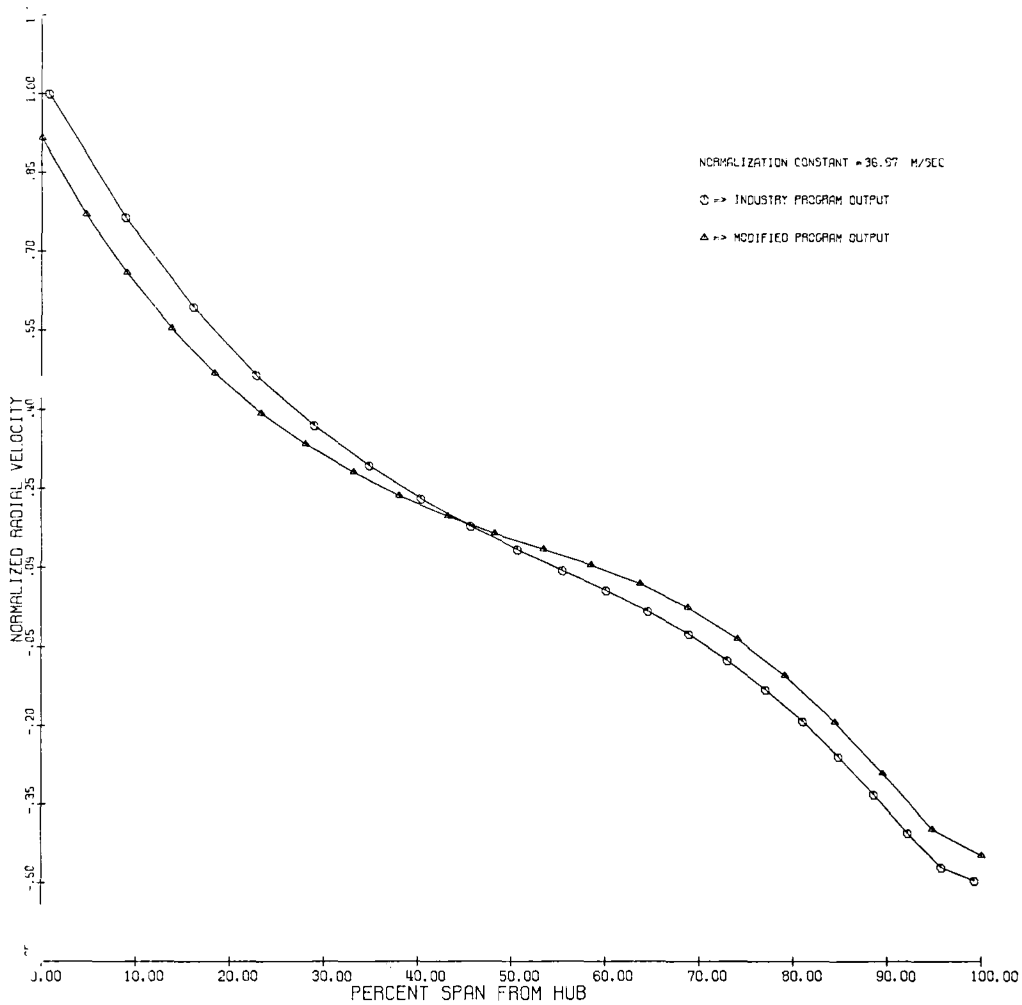
(a) Station 7.

Figure 13. - Comparison of program meridional velocity output.



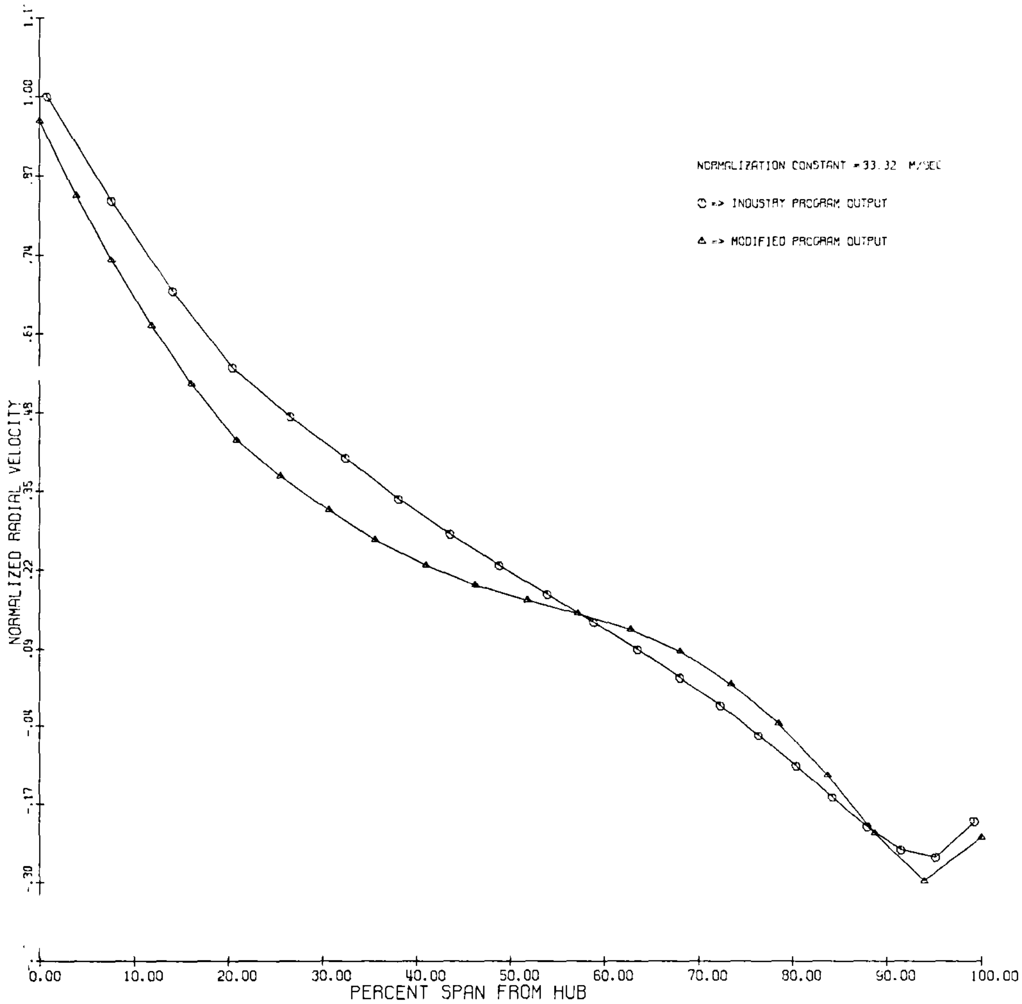
(b) Station 8.

Figure 13. - Concluded.



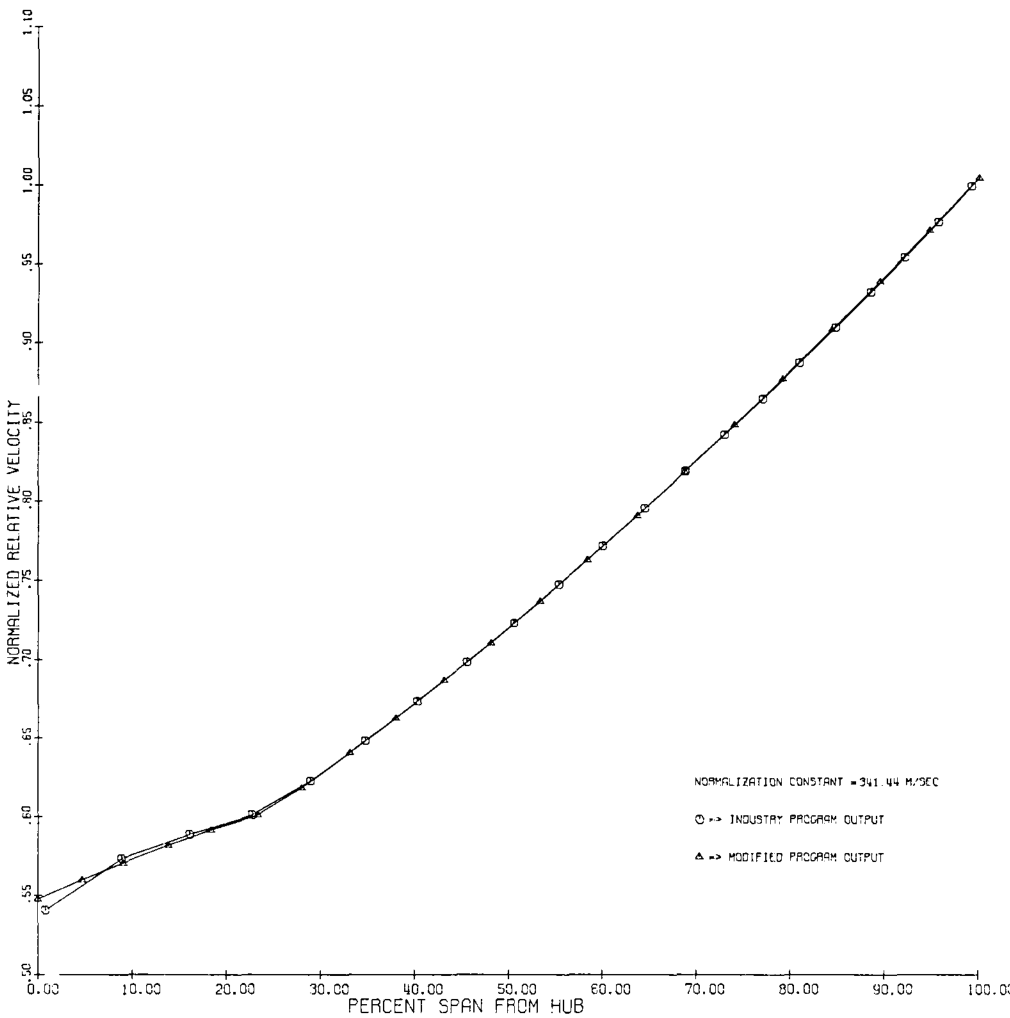
(a) Station 7.

Figure 14. - Comparison of program radial velocity output.



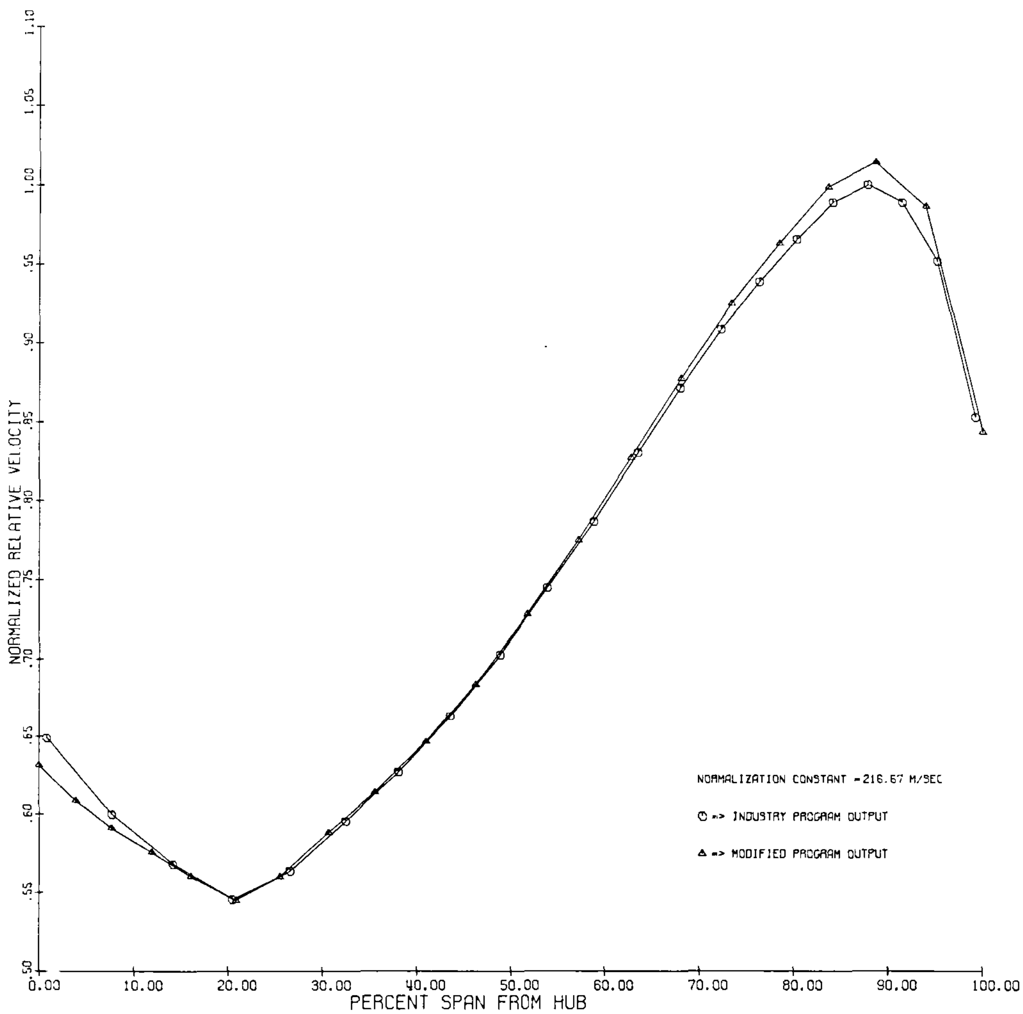
(b) Station 8.

Figure 14. - Concluded.



(a) Station 7.

Figure 15. - Comparison of program relative velocity output.



(b) Station 8.

Figure 15. - Concluded.

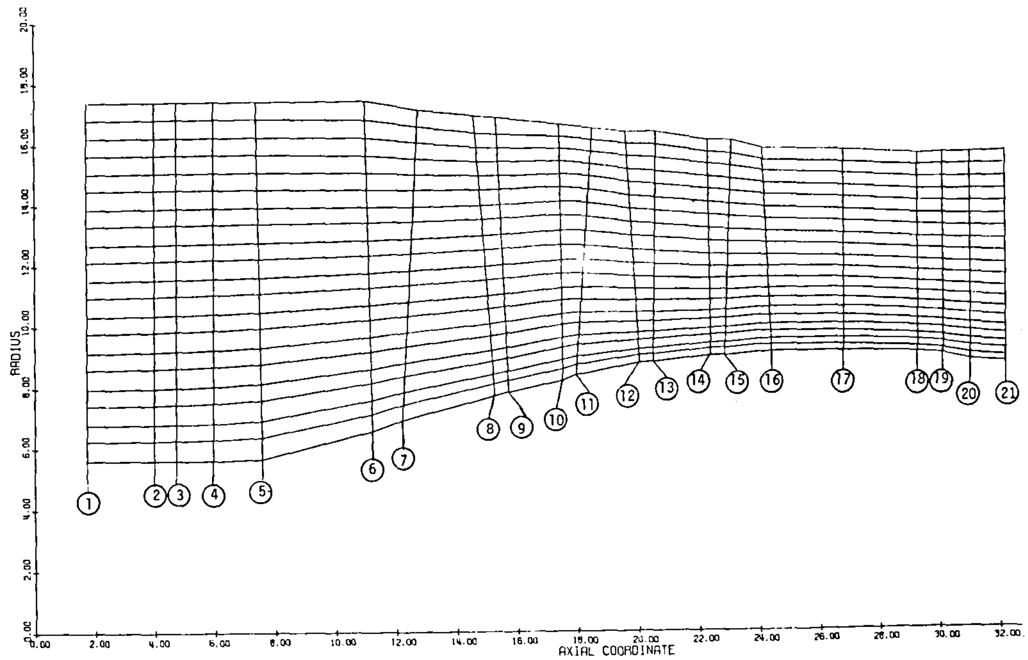


Figure 16. - Streamline contraction through fan module (program generated CALCOMP plot). One unit of length on the axes equals 2.54 cm. (1.0 inch).

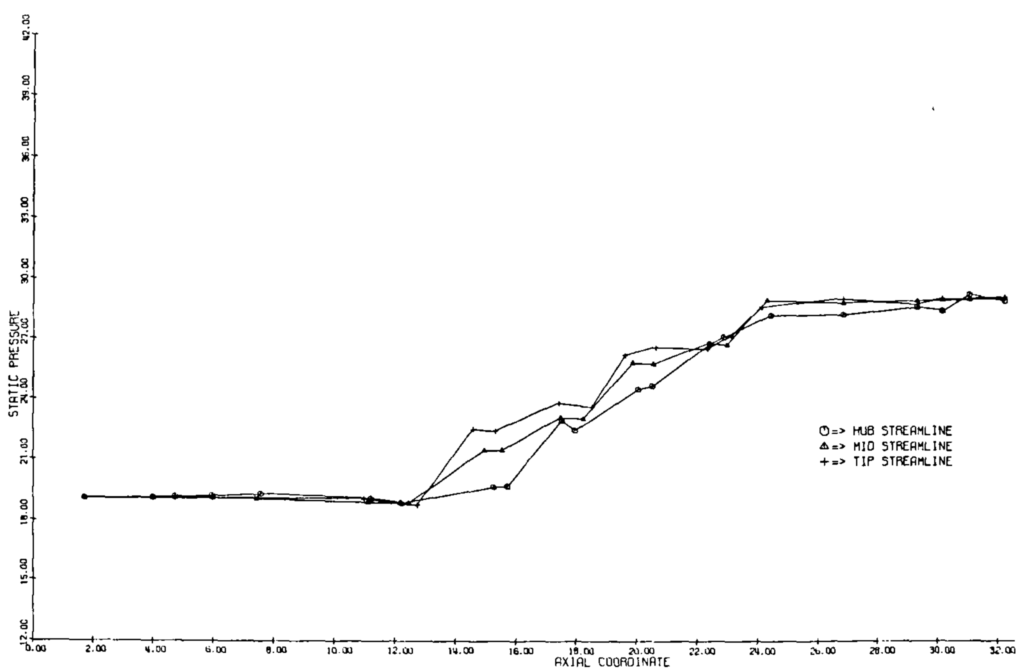


Figure 17. - Static pressure distribution in fan module (program generated CALCOMP plot). One unit of pressure equals 6.895 kPa (1.0 psia). One unit of length equals 2.54 cm. (1.0 inch).

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-79312		
4. Title and Subtitle MODIFICATION OF AXIAL COMPRESSOR STREAMLINE PRO-GRAM FOR ANALYSIS OF ENGINE TEST DATA		5. Report Date November 1979 6. Performing Organization Code
7. Author(s) Jeffrey G. Williams		8. Performing Organization Report No. E-268 10. Work Unit No.
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135		11. Contract or Grant No. 13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

This report describes modifications of an existing axial compressor streamline analysis computer program to allow input of measured radial pressure and temperature profiles obtained from engine or cascade data. The proposed modifications increases the input flexibility and are accomplished without changing the computer program's input format. The computer program was written by Richard M. Hearsey under a grant from the Aerospace Research Laboratory at Wright-Patterson Air Force Base. Since this report is intended to supplement the above computer program, the reader is referred to Hearsey's reports for theory, complete program listings, and detailed user's instructions.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Streamline analysis Axial compressor computer program	Unclassified - unlimited STAR Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified		

* For sale by the National Technical Information Service, Springfield, Virginia 22161

End of Document