N 73-20305

NASA CONTRACTOR REPORT

NASA CR-2209

NASA CR-2209

CASE FILE COPY

DESIGN AND EXPERIMENTAL PERFORMANCE OF SHORT CURVED WALL DIFFUSERS WITH AXIAL SYMMETRY UTILIZING SLOT SUCTION

by Tah-teh Yang, William G. Hudson, and Carl D. Nelson

Prepared by
CLEMSON UNIVERSITY
Clemson, S.C.
for Lewis Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • MARCH 1973

1. Report No. NASA CR-2209	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle DESIGN AND EXPERIMENTAL PERFORMANCE OF SHORT CURVED WALL DIFFUSERS WITH AXIAL SYMMETRY UTILIZING SLOT SUCTION	5. Report Date March 1973 6. Performing Organization Code
7. Author(s) Tah-teh Yang, William G. Hudson, and Carl D. Nelson	8. Performing Organization Report No. None 10. Work Unit No.
9. Performing Organization Name and Address Clemson University Clemson, South Carolina	11. Contract or Grant No. NGL 41-001-031 13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

Project Manager, Albert J. Juhasz, Airbreathing Engines Division, NASA Lewis Research Center, Cleveland, Ohio

16. Abstract

The feasibility of designing short curved wall axially symmetrical subsonic diffusers utilizing suction through slots in the diffuser walls to prevent flow separation was investigated. A potential flow analysis was made, and a digital computer program was written for determining the diffuser wall contour for prescribed boundary conditions. The flow field included branch flow so that the suction slot geometry could be a part of the diffuser design. One bell shaped diffuser and three annular diffusers with area ratios of either 2.5:1 or 3:1 were designed, fabricated, and tested. Minimum suction requirements for metastable operation ranged from 6.3% to 12% when operating with inlet air velocities in the 100 to 250 ft/sec (30 to 76 m/sec) range. For stable operation suction rates from 10% to 22% were required. In all cases the diffuser effectiveness was above 95% based on the conventional definition, and from 81% to 94% when the suction loss was accounted for. The exit velocity profiles were virtually flat with no more than ±9% variation over 95% of the exit area when operated with sufficient suction to prevent flow separation.

17. Key Words (Suggested by Author(s)) Diffuser bleed Wall suction	18. Distribution Statement Unclassified - unlimited

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 113	22. Price* $3.00

* For sale by the National Technical Information Service, Springfield, Virginia 22151

Page Intentionally Left Blank

TABLE OF CONTENTS

SUMMARY

This report presents results of an investigation concerning the analysis, design, and testing of short curved wall axially symmetrical subsonic diffusers utilizing suction through slots in the diffuser walls to prevent flow separation.

A potential flow analysis was made and a digital computer program was written for determining the diffuser wall geometry for prescribed boundary conditions. The prescribed boundary conditions included the following for the fluid velocity at the diffuser walls (both inner and outer walls for the annular diffusers): a region of nearly constant high velocity from the diffuser entrance to the slot; a region of rapid deceleration at the slot, and a region of nearly constant low velocity downstream from the slot to the diffuser exit. Branch flow was included in the flow field and this allowed the suction slot geometry to be a part of the design.

Four diffuser geometries were designed, fabricated and tested:

(1) a 7-inch (18 cm) exit diameter bell shaped diffuser with an area ratio of 3:1 and a length to inlet radius ratio of 3.5:1.

(2) a 9-inch (23 cm) exit diameter annular diffuser with an area ratio of 3:1 and a length to inlet width ratio of 5.0:1.

(3) a 9-inch (23 cm) exit diameter annular diffuser with an area ratio of 2.5:1 and a length to inlet width ratio of 3.9:1.

(4) an 18-inch (46 cm) exit diameter annular diffuser with an area ratio of 3:1 and a length to inlet width ratio of 3.6:1.

For inlet air velocities in the 200-250 ft/sec (61-76 m/sec) range, metastable operation required 6.5% suction for the bell shaped diffuser while

stable operation required 12% suction. Corresponding suction rates for the 9-inch (23 cm) annular diffuser (3:1 area ratio) were 8.5% and 16.0% for metastable and stable operation respectively. For the 9-inch (23 cm) annular diffuser with 2.5:1 area ratio, 6.3% and 12% suction rates were required. For the 18-inch (46 cm) annular diffuser, only 7% suction was required for metastable operation and 10% for stable operation.

Suction rate requirements were found to increase as inlet velocity was decreased. Thus for stable operation of the 18-inch (46 cm) diffuser the required suction rate was about 20% at 100 ft/sec (30 m/sec) inlet velocity.

In all cases the diffuser effectiveness was above 95% based on the conventional definition, and from 81% to 94% when the suction loss was accounted for. The exit velocity profiles were virtually flat with no more than ± 9% variation over 95% of the exit area when operated with sufficient suction to prevent separation.

Non-uniform inlet tests and tests with periodic shedding of wakes upstream of the diffuser inlet were conducted on the 9-inch (23 cm) annular diffuser with AR=3:1. Non-uniform inlet conditions caused no significant increase in the required suction rates. However, periodic shedding of wakes upstream of the diffuser inlet caused the suction requirement for stable operation to be increased from 16.0% to 18.0%.

A downstream obstruction in the form of a perforated metal sheet at the diffuser exit slightly increased the minimum suction rate requirement at low inlet velocities and had no significant effect upon the suction rate requirement at inlet velocities above 150 ft/sec (46 m/sec).

LIST OF SYMBOLS

A	Area
AR	Area ratio, exit to inlet
C_p	Pressure recovery coefficient, $\frac{P_{s,e} - P_{s,i}}{P_{d,i}}$
D	Diameter
FS	Fraction of inlet flow removed by suction
KE	Kinetic energy per unit mass flowing
$\dot{m}$	Mass flow rate
n	Natural coordinate normal to streamlines
P	Pressure (either static, dynamic, or total depending upon subscript)
Q	Local to upstream velocity vector magnitude ratio
R	Radial coordinate of axially symmetrical system
Re No	Reynold's number, $\left[\frac{(4\ A_{cs}/\text{perimeter})(\text{velocity})}{\text{fluid kinematic viscosity}}\right]_{\text{inlet}}$
s	Natural coordinate along a streamline
U	Fluid velocity outside the boundary layer
$\overline{U}$	Mean velocity at a section
W	Width of diffuser inlet passage
X	Axial coordinate of axially symmetrical system
α	Velocity vector angle measured from horizontal
δ	Boundary layer displacement thickness
ε	Diffuser effectiveness, $\frac{\dot{m}_e\ (P_{s,e} - P_{s,i})}{\dot{m}_i\ (P_{s,e} - P_{s,i})_{\text{ideal}}}$
η	Diffuser effectiveness, $\frac{(P_{s,e} - P_{s,i})}{(P_{s,e} - P_{s,i})_{\text{ideal}}}$

κ	Kinetic energy coefficient, $\frac{1}{A}\int_A \left(\frac{U}{\bar{U}}\right)^3 dA$
Φ	Maximum value of the velocity potential function
ϕ	Velocity potential function
Ψ	Maximum value of the stream function
ψ	Stream function
ρ	Fluid density

Subscripts

cs	Cross sectional
e	Diffuser exit
d,i	Dynamic inlet
i	Diffuser inlet
max	Maximum
s,e	Static exit
s,i	Static inlet
t,e	Total exit
t,i	Total inlet
1	Upstream
2	Downstream

SECTION I
INTRODUCTION

1.1 Background

In an internal flow system, a diffuser is a transitional section which connects a flow passage having a smaller cross-sectional area to a flow passage of larger area. Two desirable characteristics of a diffuser are that it will provide a nearly uniform velocity distribution at its exit plane and that it will transform most of the kinetic energy of the fluid at entrance into potential energy in the form of a higher static pressure at the exit plane.

A previous report by the authors [1][1] concerned the feasibility of designing short two-dimensional diffusers utilizing suction through the diffuser wall slots. The design philosophy was to prescribe the diffuser wall velocity so as to have essentially constant high velocity upstream of a slot, constant low velocity downstream of the slot, and thus have all the deceleration occur across the slot. By removing through the slot those fluid particles having insufficient kinetic energy to get across the adverse pressure gradient in the region of the slot, flow separation could be avoided. The concept of having deceleration occur abruptly and applying slot suction to this narrow region was originally suggested by A. A. Griffith as reported by Lachmann [2] and is henceforth referred to as the Griffith diffuser.

The two-dimensional Griffith diffusers designed and tested were only partially successful. When sufficient suction was applied to prevent

---

[1] Numbers in brackets refer to the references at the end of this report.

separation, the diffuser exit plane velocity distribution was nearly uniform and very low total pressure losses were achieved. However, the suction rates required for stable operation varied from 16% for 2:1 area ratio to 44% for 4:1 area ratio. Problems were encountered in designing the optimum slot geometry and also in controlling the sidewall boundary layer.

In 1952, L. R. Manoni [3] designed several bell channel diffusers based upon the Griffith concept. He used an electrical analogy tank to determine the wall geometry required for the specified wall velocity distribution. While his design method was only approximate and somewhat cumbersome, he was able to demonstrate that such a concept would enable the design and operation of a relatively short diffuser without separation provided that sufficient slot suction could be applied.

In order to design planar flow channel geometries, J. D. Stanitz [4] in 1953 performed a double transformation of the describing equations into a coordinate system where the geometry did not need to be known for the equations to be solved. After the equations were solved on the transformed plane, the inverse transformation yielded the desired geometry.

In 1971, C. D. Nelson [5] extended the work of Stanitz by developing a method for transforming the flow equations of an axially symmetrical coordinate system. Nelson also devised an approximation to account for the branch flow due to the slot by solving the channel design problem with a more complicated set of boundary conditions.

1.2 Objective

The objective of the research effort described in this report was to examine the concept and the practicality of a short axially symmetrical subsonic curved wall diffuser using suction through slots in the diffuser walls to prevent flow separation.

1.3 Scope

In order to achieve the objective a research program was carried out to

(1) Develop a digital computer program which will determine the wall shape of an axially symmetrical or a two-dimensional diffuser for prescribed boundary conditions.

(2) Utilize the program to design one bell-shaped diffuser and three annular diffusers with area ratios of about 3:1.

(3) Fabricate and test each of the above diffusers to determine

- (a) The minimum suction requirement for unseparated flow as a function of inlet air velocity;
- (b) Performance characteristics such as diffuser effectiveness, total pressure loss, and exit plane velocity distribution;
- (c) Effect of non-uniform inlet velocity distributions;
- (d) Effect of periodic shedding of wakes upstream of the diffuser inlet;
- (e) Effect of imbalance between inner and outer wall suction rates per unit slot length;
- (f) Effect of downstream blockage in the form of perforated steel plates.

SECTION 2
ANALYSIS AND DESIGN

2.1 Analysis for Determination of Diffuser Geometry

The describing equations of incompressible irrotational motion and continuity in the axially symmetrical coordinate system are

$$\frac{\partial^2 \phi}{\partial R\,\partial X} - \frac{\partial^2 \phi}{\partial X\,\partial R} = 0 \qquad (1)$$

and

$$\frac{\partial^2 \phi}{\partial X^2} + \frac{\partial^2 \phi}{\partial R^2} + \frac{1}{R}\frac{\partial \phi}{\partial R} = 0 \qquad (2)$$

The geometry of the channel being designed might appear as shown in Figure 1.

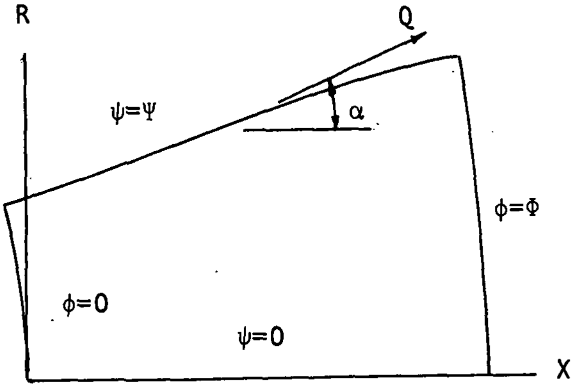


Figure 1. Possible flow channel geometry on the physical plane.

Transforming equations (1) and (2) into natural coordinates by using

$$ds = \cos\alpha\,dX + \sin\alpha\,dR \qquad (3)$$
$$dn = -\sin\alpha\,dX + \cos\alpha\,dR \qquad (4)$$

and introducing the stream function and velocity potential in natural coordinates

$$d\psi = R\,Q\,dn \qquad (5)$$
$$d\phi = Q\,ds \qquad (6)$$

results in the equations

$$\frac{\partial \ln Q}{\partial n} - \frac{\partial \alpha}{\partial s} = 0 \tag{7}$$

and

$$\frac{\partial \ln Q}{\partial s} + \frac{1}{R}\frac{\partial R}{\partial s} + \frac{\partial \alpha}{\partial n} = 0 \tag{8}$$

The channel geometry on the natural coordinate plane might then appear as shown in Figure 2.

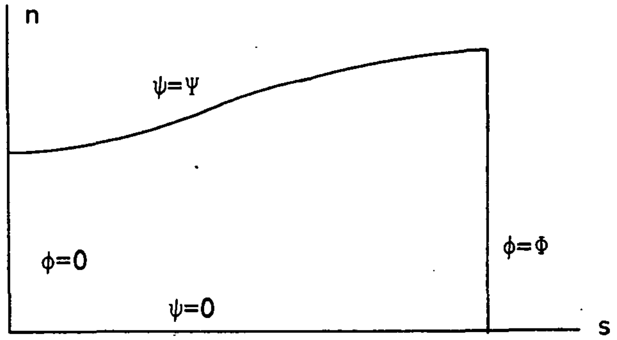


Figure 2. Possible flow channel geometry on natural coordinate system.

Now using equations (5) and (6) to transform from natural coordinates to a ϕ-Ψ coordinate system results in the following equations of irrotational motion and continuity.

$$R\frac{\partial \ln Q}{\partial \psi} - \frac{\partial \alpha}{\partial \phi} = 0 \tag{9}$$

and

$$\frac{1}{R}\frac{\partial \ln Q}{\partial \phi} + \frac{1}{R^2}\frac{\partial R}{\partial \phi} + \frac{\partial \alpha}{\partial \psi} = 0 \tag{10}$$

The flow field on the ϕ-Ψ plane might appear as shown in Figure 3.

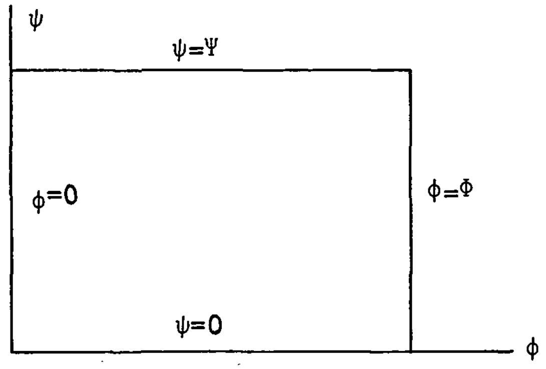


Figure 3. Possible flow field on stream function-velocity potential plane.

Cross differentiation of equations (9) and (10) to eliminate α yields

$$R^4 \frac{\partial^2 \ln Q}{\partial \psi^2} + R^3 \frac{\partial R}{\partial \psi} \frac{\partial \ln Q}{\partial \psi} + R^2 \frac{\partial^2 \ln Q}{\partial \phi^2}$$

$$-R \frac{\partial R}{\partial \phi} \frac{\partial \ln Q}{\partial \phi} + R \frac{\partial^2 R}{\partial \phi^2} - 2 \left(\frac{\partial R}{\partial \phi}\right)^2 = 0 \tag{11}$$

The inverse transformations for α, X, and R are

$$d\alpha = R \frac{\partial \ln Q}{\partial \psi} d\phi - \left(\frac{1}{R} \frac{\partial \ln Q}{\partial \phi} + \frac{1}{R^2} \frac{\partial R}{\partial \phi}\right) d\psi \tag{12}$$

$$dX = \frac{\cos \alpha}{Q} d\phi - \frac{\sin \alpha}{RQ} d\psi \tag{13}$$

$$dR = \frac{\sin \alpha}{Q} d\phi + \frac{\cos \alpha}{R Q} d\psi \tag{14}$$

Since in eqn. (11) the radial coordinate R remains as a dependent variable, one method of solution consists of assuming values for the radial coordinates and then alternately solving for ln Q and R in an iterative manner.

As a first trial, assume R=1 which reduces equation (11) to

$$\frac{\partial^2 \ln Q}{\partial \psi^2} + \frac{\partial^2 \ln Q}{\partial \phi^2} = 0 \tag{15}$$

which is the same as the corresponding plane flow equation of Stanitz [4]. The term ln Q may be found for a given set of boundary conditions by utilizing some numerical scheme such as the Gauss-Siedel method. Boundary conditions are prescribed along the wall streamlines and along the inlet and outlet equipotential lines.

The first approximation to the velocity vector angles α are then computed from

$$\alpha_2 - \alpha_1 = \int_{\psi} \frac{\partial \ln Q}{\partial \psi} \, d\phi - \int_{\phi} \frac{\partial \ln Q}{\partial \phi} \, d\psi \tag{16}$$

where the path of integration is chosen to be along equipotential lines and streamlines.

The R coordinates are then computed using the line integrals

$$R_2 - R_1 = \int_{\psi} \frac{\sin \alpha}{Q} \, d\phi \tag{17}$$

and $$R_2^2 - R_1^2 = 2 \int_{\phi} \frac{\cos \alpha}{Q} \, d\psi \tag{18}$$

These R coordinates are treated as a second approximation to solve equation (11) again for new velocities. New flow angles are computed using the line integrals

$$\alpha_2 - \alpha_1 = \int_{\psi} R \frac{\partial \ln Q}{\partial \psi} \, d\phi \tag{19}$$

and $$\alpha_2 - \alpha_1 = -\int_{\phi} \left(\frac{1}{R} \frac{\partial \ln Q}{\partial \phi} + \frac{1}{R^2} \frac{\partial R}{\partial \phi} \right) d\psi \tag{20}$$

Corrected R coordinates are again computed from equations (17) and (18) and the entire process is reiterated until the R coordinates have converged sufficiently.

The axial coordinates are computed next by using the line integrals

$$X_2 - X_1 = \int_{\psi} \frac{\cos\ \alpha}{Q}\ d\phi \tag{21}$$

and $$X_2 - X_1 = -\int_{\phi} \frac{\sin\ \alpha}{R\ Q}\ d\psi \tag{22}$$

In order to account for the branch flow due to slot suction, a more complicated set of boundary conditions are necessary. The physical flow system may now appear as in Figure 4. The formerly simple channel wall now involves curves B, C, D, and E with some of the flow leaving across C. This introduces a stagnation point at A.

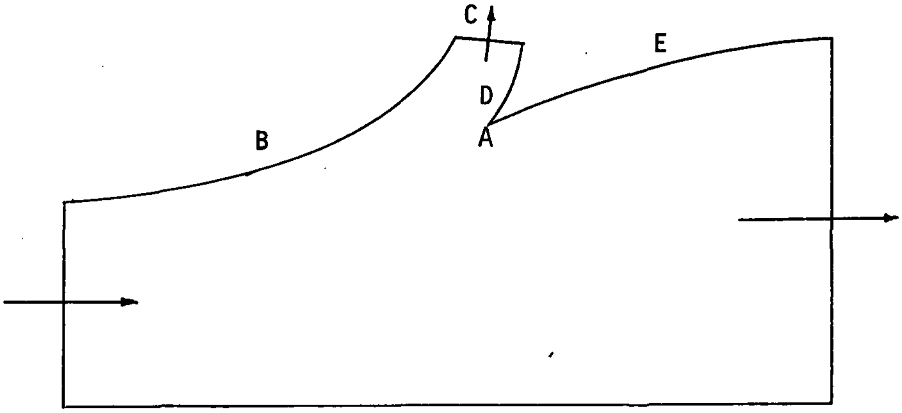


Figure 4. Possible geometry with branch flow.

On streamlines B, D, and E, the value of the desired velocity is prescribed as usual, but on line C (which is chosen to be a constant potential line),

$$\frac{\partial\ \ln Q}{\partial\phi} = 0 \quad \text{and} \quad \frac{\partial R}{\partial\phi} = 0$$

are specified which in essence prescribes a parallel flow across C.

The stagnation point A introduces a singularity into equation (11) since ln Q is not defined when Q = 0. In order to circumvent the difficulty introduced by this singularity, a portion of the potential flow solution for two-dimensional flow around a wedge was patched into a small region around the stagnation point. The introduction of this two-dimensional wedge flow patch yields only an approximate solution which becomes more accurate as the radial dimension of the patch becomes small relative to the distance of the stagnation point from the axis of symmetry.

2.2 Digital Computer Design Program

The digital computer program which was used to design the wall contours of the Griffith diffusers described in this report has the following features.

(1) It may be used to design either planar (two-dimensional) or axially symmetrical (bell shaped or annular) flow channels.

(2) Branch flow due to slot suction is incorporated into the design program.

(3) The boundary conditions are flexible; i.e., at either entrance to or exit from the flow channel, the velocities may be prescribed (Dirichlet boundary conditions) or the velocity gradients may be set to zero (Neumann boundary conditions) so as to result in unidirectional flow at either entrance or exit. The wall velocities are prescribed at up to 148 points along each channel wall (both inner and outer walls if annular).

(4) Up to 148 mesh points may be used in the phi direction and up to 48 points may be used in the psi direction.

(5) Printed output includes the coordinates (R and X), velocities (Q), and flow angles (α) for each mesh point and a summary of the wall coordinates of the channel design.

(6) Graphical output includes all the streamlines and potential flow

lines on the X-R (or X-Y) plane. Also, the inlet and outlet velocity profiles are plotted as a function of the R (or Y) coordinate.

Using a 48 x 135 grid, a typical axially symmetrical design requires about 45 minutes on an IBM System 360 Model 50 computer. A typical two-dimensional design of comparable grid points requires about 20 minutes.

The program listing is given in Appendix A. Appendix B describes briefly the procedure used in obtaining the 18-inch (46 cm) annular diffuser design.

2.3 Griffith Diffuser Geometry

Figures 5, 6, and 7 show the geometries of the bell shaped diffuser, the 9-inch (23 cm) annular diffuser, and the 18-inch (46 cm) annular diffuser respectively. The diffusers were designed to have an area ratio of about 3:1 since this represents a typical diffuser application.

The diffuser geometries were analyzed using the Douglas Neumann technique [6] to predict the velocity distributions along the diffuser walls and at the diffuser inlet and exit planes. A comparison was made between the velocities predicted by the analysis program and those prescribed to the design program. In each case the agreement was good although slight disagreements occurred in the vicinity of the slot.

The bell shaped diffuser had the simplest geometry and was designed and tested first. Only one curved wall had to be considered and its slot was relatively far away from the axis of symmetry.

The second geometry designed and tested was the 9-inch (23 cm) annular diffuser. Since two slotted walls were involved with the inner wall slot being closer to the axis of symmetry, this case provided a more severe test of the slot design procedure using the two-dimensional wedge flow patch.

The third diffuser used the same outer body as the second diffuser, but a smaller inner body was designed so as to change the area ratio from 3:1 to

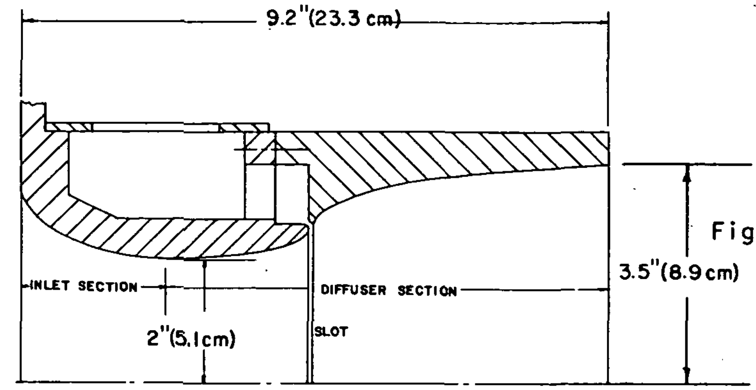


Figure 5. Bell shaped diffuser

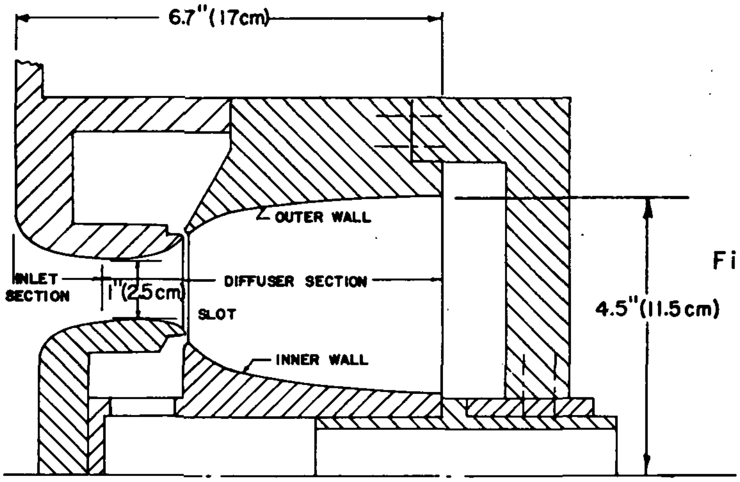


Figure 6. 9" (23 cm) annular diffuser

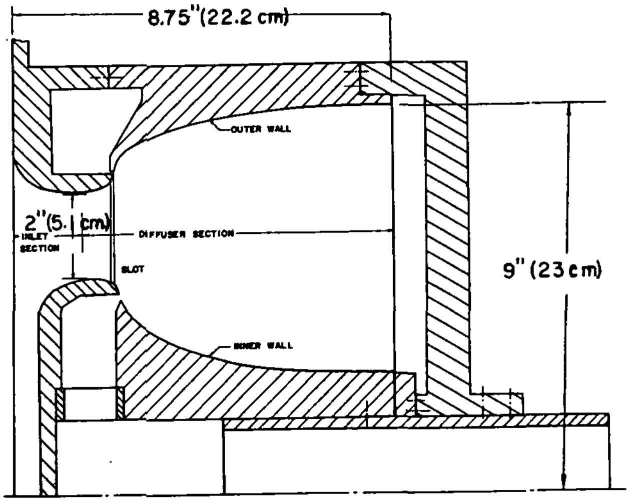


Figure 7. 18" (46 cm) annular diffuser

2.5:1. This diffuser was to demonstrate that a common outer body could be used with more than one inner body thus providing more than one area ratio without a completely new diffuser.

The last diffuser designed and tested was the 18-inch (46 cm) annular diffuser. This diffuser was to verify that a reduction of slot suction can be achieved by reducing the ratio of $\overline{\delta}/W$. $\overline{\delta}$ is the boundary layer displacement thickness just upstream of the slot, and W is the annular width of the diffuser at inlet.

SECTION 3
INSTRUMENTATION AND APPARATUS

3.1 Test Facility

Two primary air supplies were used in testing the diffusers. The 18" annular diffuser was tested using a 30,000 CFM (14 m^3/sec) industrial fan delivering air through a 42-inch (106 cm) by 48-inch (122 cm) duct 30 ft. (9.1 m) long.

All other diffusers were tested using a 10,000 CFM (4.72 m^3/sec) fan delivering air through a 20-inch (51 cm) circular duct 20 ft. (6.1 m) long. Flow straightener tubes and fine mesh screens were installed inside the duct to provide a uniform steady flow to the test section. A shutter type damper at the fan inlet was used to regulate the flow rate.

Two diesel superchargers were installed in parallel providing for up to 900 CFM (0.43 m^3/sec) suction air through the diffuser slots. Seven gate valves located in six suction lines and an air bleed line provided a flexible setup for varying both the total amount of slot suction and the distribution of suction between the inner wall and the outer wall of the annular diffusers. Plastic pipe lengths and fittings were attached between the diffuser and the suction control valves and all joints were sealed with tape to eliminate air leakage.

The diffusers were fabricated from aluminum stock. The wall contours were cut on a pattern lathe. Figure 8 shows the 18-inch (46 cm) annular diffuser attached to the end of the large air supply duct. Figure 9 shows the bell shaped diffuser attached to the small air supply duct.

3.2 Instrumentation and Measurements

The stagnation pressure at the end of each duct and the static pressure at the diffuser inlet were measured with a micromanometer having resolution of 0.001 inch of water. Wall static pressure taps were installed in the diffusers by glueing 1/16-inch o.d. brass tubes into holes drilled in the diffuser walls and smoothing the ends to conform to the diffuser contour. The pressure signals were monitored on an inductive type pressure transducer through a rapid scanning mechanism. Up to 48 pressure taps could be scanned in 30 seconds. The output of the pressure transducer was recorded on paper tape with a digital recorder.

Velocity profiles were measured with a constant temperature hot wire probe. The voltage output of the probe was displayed by a multirange digital voltmeter with a selective damping device. The probe was positioned in the diffuser with a traversing mechanism adapted from a milling machine bed.

Depending upon the size of diffuser being tested either five or six flow meters were used to measure the slot suction rates. A manometer bank with 20 legs each 60-inches high employing either water or carbon tetrachloride was used to obtain the flow meter pressure differentials. A copper-constantan thermocouple was used to measure the air temperature, and a well-type barometer was used to obtain atmospheric pressure.

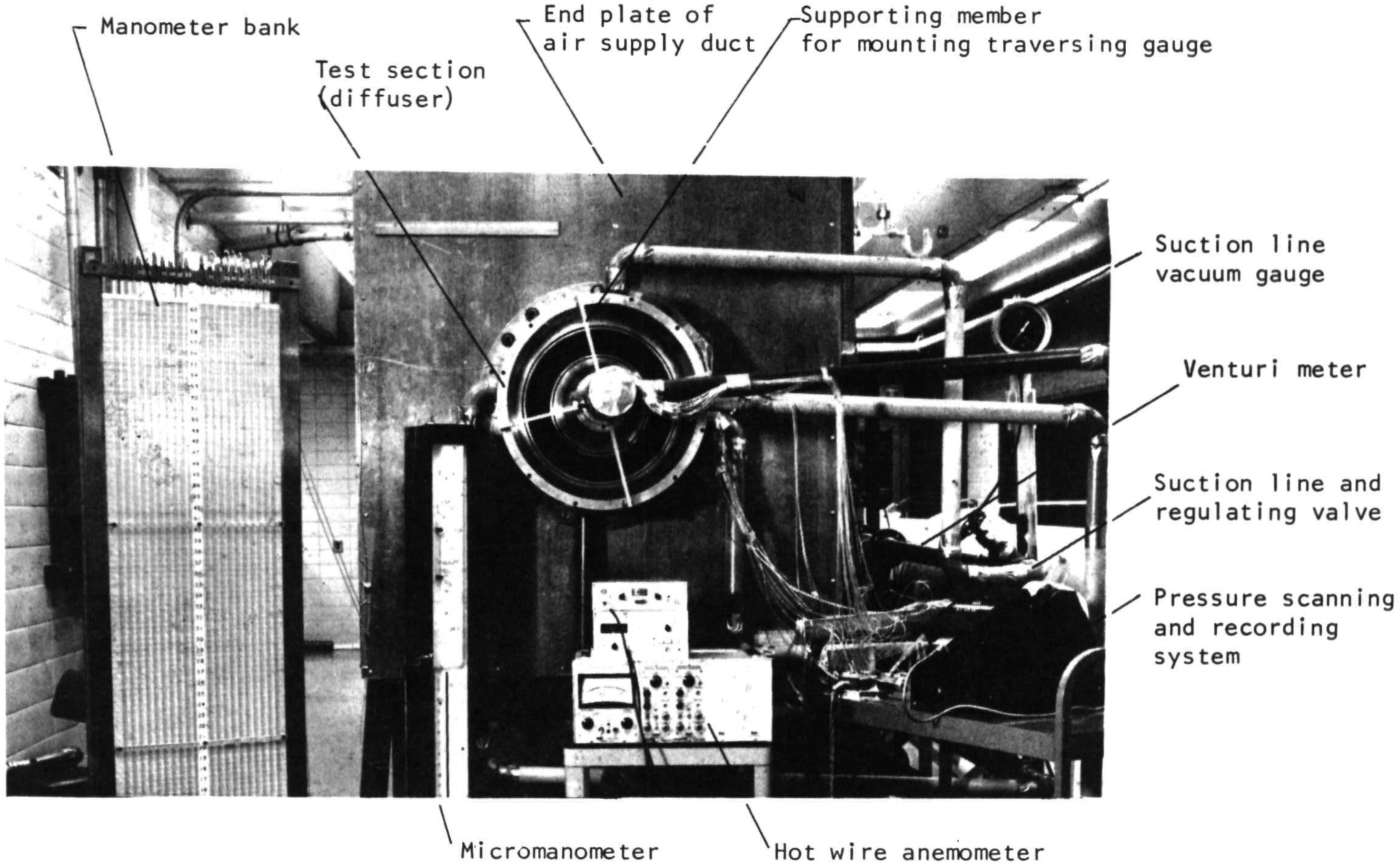


Figure 8. 18" (46 cm) annular diffuser attached to the large air supply duct

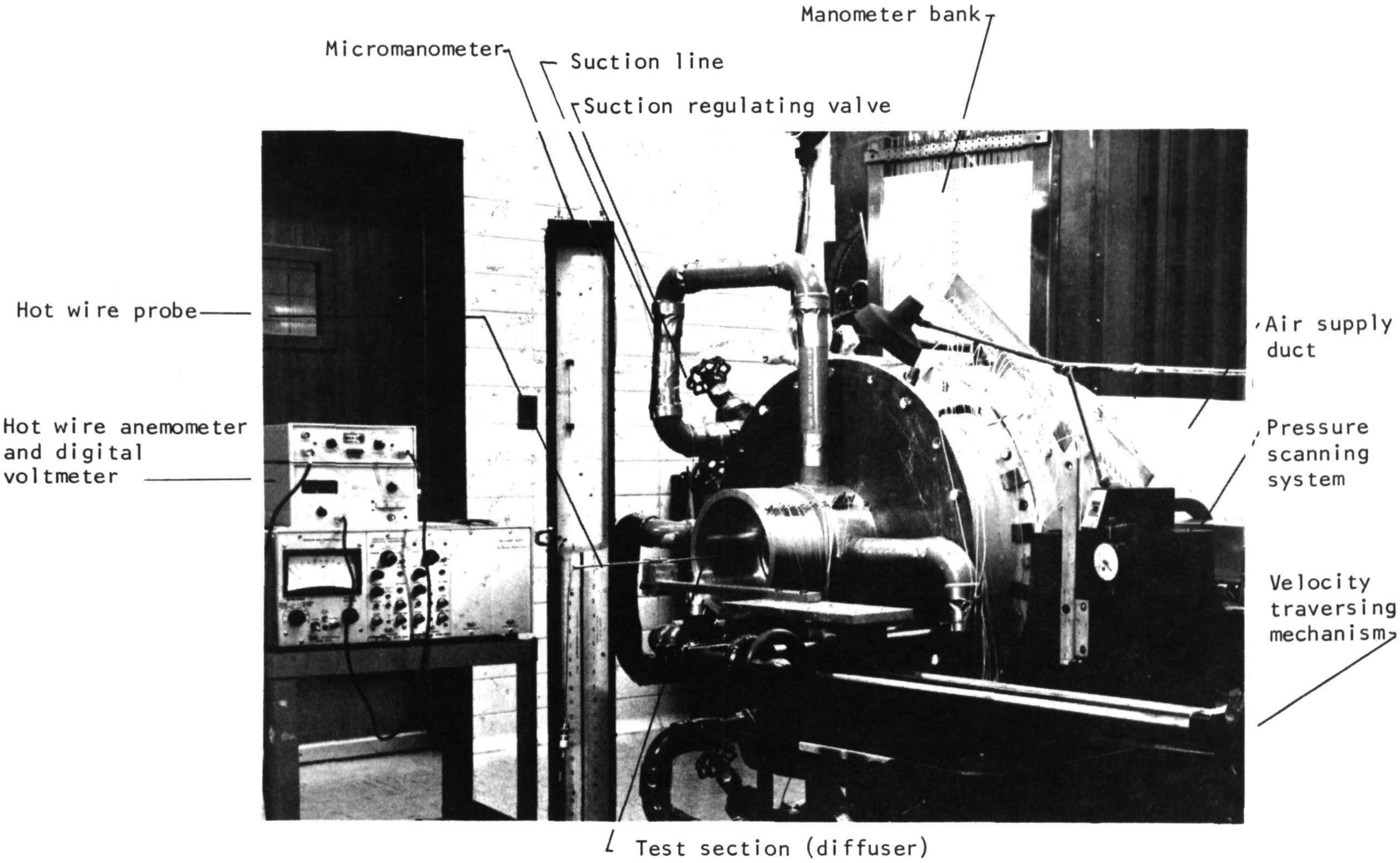


Figure 9. Bell shaped diffuser attached to the small air supply duct

SECTION 4

TEST CONDITIONS AND PROCEDURE

4.1 Test Conditions

As reported in reference [1], the Griffith diffuser operates with either very high effectiveness (when unseparated flow exists) or very low effectiveness (when separated flow exists). The minimum suction rate required for unseparated flow is thus a very important diffuser characteristic.

Experimental work showed that two modes of operation with unseparated flow can exist. In the stable mode a flow disturbance may cause temporary flow separation, but unseparated flow will return when the disturbance is removed. In the metastable mode, unseparated flow persists even if a disturbance sufficient to cause separation is removed. Such factors as inlet velocity, area ratio, suction slot width, and diffuser downstream conditions have an effect upon the minimum suction required for both metastable and for stable unseparated flow.

An extensive set of tests were made to determine the minimum suction requirements for unseparated flow for each diffuser. Inlet air velocities were varied from 50 ft/sec (15.2 m/sec) to 280 ft/sec (85 m/sec). For the 9-inch (23 cm) annular diffuser, slot widths of 1/16-inch (0.159 cm) to 1/4-inch (0.635 cm) were used. For the 18-inch (46 cm) annular diffuser, slot widths of 1/8-inch (0.317 cm) to 1/2-inch (1.27 cm) were used. Area ratios of 2.5:1 and 3:1 were used.

The effect of downstream conditions was investigated in two ways.

(1) A perforated sheet metal plate with 51% free area was attached to

the diffuser exit plane of the 9-inch (23 cm) and of the 18-inch (46 cm) annular diffuser. The 18-inch (46 cm) diffuser plate had 5/8-inch (1.59 cm) holes while the 9-inch (23 cm) diffuser plate had 5/16-inch (.79 cm) holes.

(2) The usual annular diffuser requires struts which may cause flow separation. Three struts located 120° apart were mounted in the downstream constant velocity zone of the 9-inch (23 cm) annular diffuser (AR = 2.5:1) to simulate this structural problem.

The effect of an unsteady inlet velocity was examined by installing a motor driven propeller just upstream of the inlet to the 9-inch (23 cm) diffuser. The effect upon suction requirements when driven at 500 RPM was examined.

In addition to tests made to determine minimum suction requirements, performance tests were made to determine diffuser effectiveness, total pressure loss, wall pressure distribution, exit plane velocity distribution, and center line velocity distribution for the four diffusers when operating with about 200 ft/sec (60 m/sec) inlet velocity and with a suction rate sufficient to maintain unseparated flow.

4.2 Test Procedure

In determining the minimum suction requirement for stable operation, the pressure differential between the end of the supply duct and the end of the diffuser inlet section was observed in order to adjust the damper so as to give the desired inlet velocity. The suction blowers were started with the air bleed valve wide open and the slot suction valves nearly closed, thus having a deficiency of suction. The suction valves were slowly opened while observing the venturi meter manometers so as to keep the proper ratio of outer wall to inner wall suction rate. This ratio was approximately the same as the slot length ratio. The air bleed valve was slowly closed so as to increase the suction capacity either simultaneously with or subsequent to the opening of the suction valves. Increasing the suction rate sufficiently would cause the flow to become unseparated, and this condition was considered to be the minimum for stable operation.

In determining the minimum suction requirement for metastable operation, the suction rate was decreased from the amount required for stable operation (or from any excess suction rate) until separation occurred. It was again necessary to observe the flow manometers as the suction rate was reduced so as to keep the suction rate per unit length of slot about the same on the inner and outer walls.

For tests involving performance characteristics such as effectiveness, velocity distribution, and wall pressure distribution, three static pressure measurements were made with the micromanometer: (1) at the end of the air supply duct (2) at the diffuser inlet (3) differential between (1) and (2). Flow meter manometers were read and recorded. Wall pressure readings were recorded on paper tape.

For velocity traverses, the reference position of the probe traversing mechanism was observed. Measurements were then made at the various desired stations by manipulating the traversing mechanism to the desired position and recording the hot-wire voltage.

SECTION 5

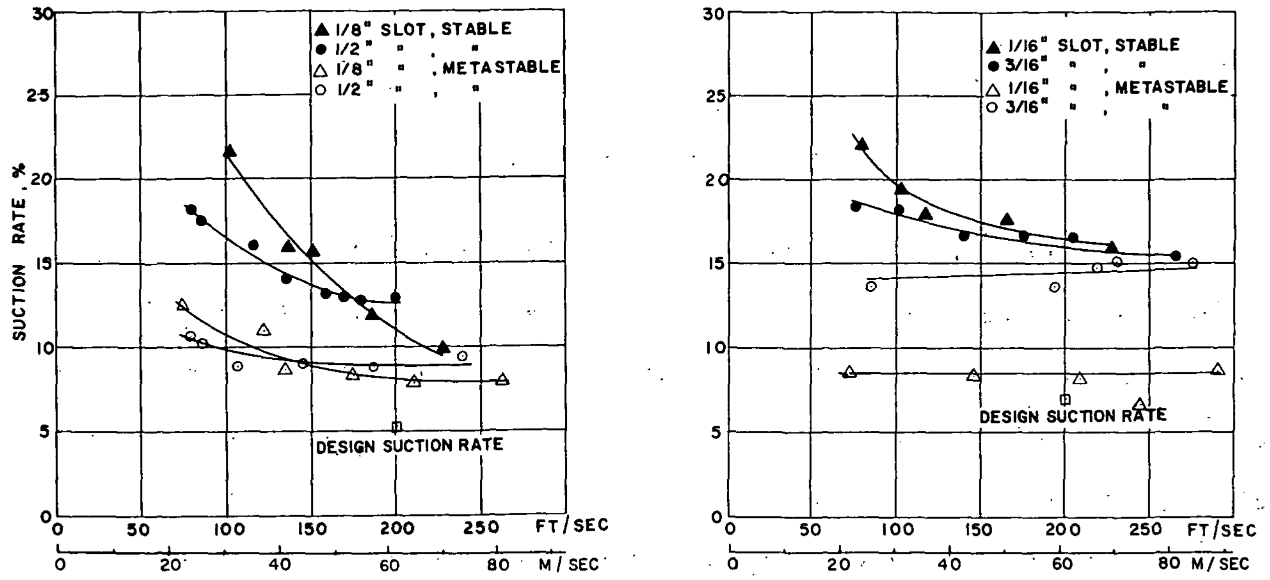
DISCUSSION OF RESULTS

The results will be presented in the following arrangement. The minimum suction requirements for unseparated flow will be shown first. Then measured and predicted velocity distributions at the exit planes, the walls, and the passage centerlines will be shown. Further the effects of downstream blockage, of non-uniform inlet velocities, and of unsteady inlet velocity will be discussed. Finally a table of diffuser performance characteristics will be presented.

5.1 Minimum Suction Requirement

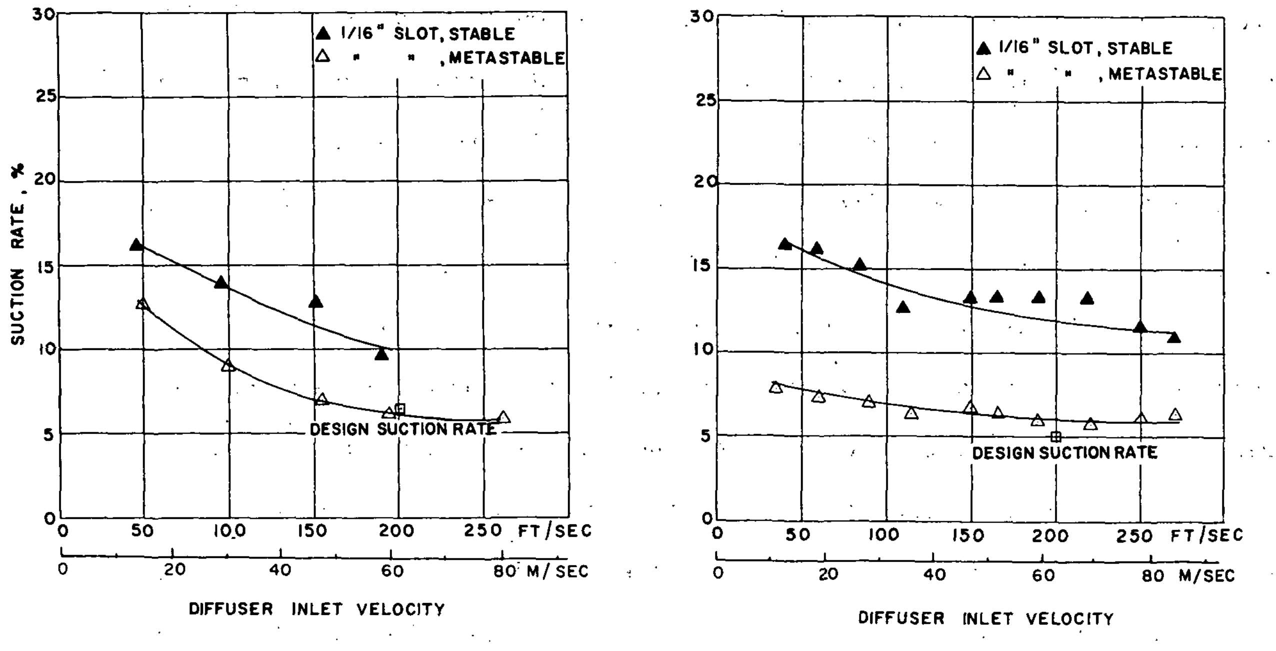
Figure 10 shows the minimum suction requirements for unseparated flow through the four diffusers when approximately uniform inlet velocities exist. Both stable and metastable requirements are shown when operated with the design slot width (1/8-inch (.32 cm) for 18-inch (46 cm) diffuser, and 1/16-inch (.16 cm) for the other diffusers) and, for two of the diffusers, when operated with a slot width several times wider than the design width. A wider slot enables a given suction rate to be removed with a smaller pressure drop through the slot. In each case the design suction requirement is shown for an inlet velocity of 200 ft/sec (61 m/sec). The design suction was determined by assuming the one-seventh power law for the boundary layer velocity profile and applying Taylor's criterion [7]. This criterion gives an estimate of the amount of fluid wich must be removed to prevent flow separation due to an unfavorable pressure gradient across the suction slot. The boundary layer fluid from the solid surface up to where the velocity ratio

$$\frac{u}{U_1} = \sqrt{1.0 - \left(\frac{U_2}{U_1}\right)^2} \qquad (23)$$



a.18-INCH (46 cm) ANNULAR DIFFUSER AR=3:1

b.9-INCH (23 cm) ANNULAR DIFFUSER AR=3:1



c.9-INCH (23 cm) ANNULAR DIFFUSER AR=2.5:1

d.BELL SHAPED DIFFUSER AR=3:1

Figure 10. Minimum suction requirement for unseparated flow when using uniform inlet velocity.

exists must be removed according to Taylor's criterion. U_1 and U_2 are the velocities just upstream and downstream of the slot at the outer edge of the boundary layer, and u is the velocity within the boundary layer upstream of the slot along that streamline which ends in a stagnation point just downstream of the slot.

The results indicate that the design suction rate was very close to that required for metastable operation for the 9-inch (23 cm) annular diffuser (AR = 2.5:1). For the 9-inch (23 cm) annular (AR = 3:1) and for the bell shaped diffusers the design suction rate was slightly below that required for metastable operation. For the 18-inch (46 cm) diffuser, the design rate of 5% was 40% lower than the 7% actually required for metastable operation.

The suction rates required tended to drop asymptotically as the inlet velocities were increased. Since the boundary layer thickness tends to decrease with increased inlet velocities, this trend was expected.

The bell shaped diffuser required slightly smaller suction rates than the annular diffusers having the same area ratio because only one wall requires suction. Dropping the area ratio from 3:1 to 2.5:1 also reduced the suction requirement as expected.

The suction requirement for stable operation was of course above the metastable operation requirement but the differences tended to become less at higher inlet velocities and with wider slots. The wider slots also tended to make the suction requirement curves flatter or less dependent upon inlet velocities.

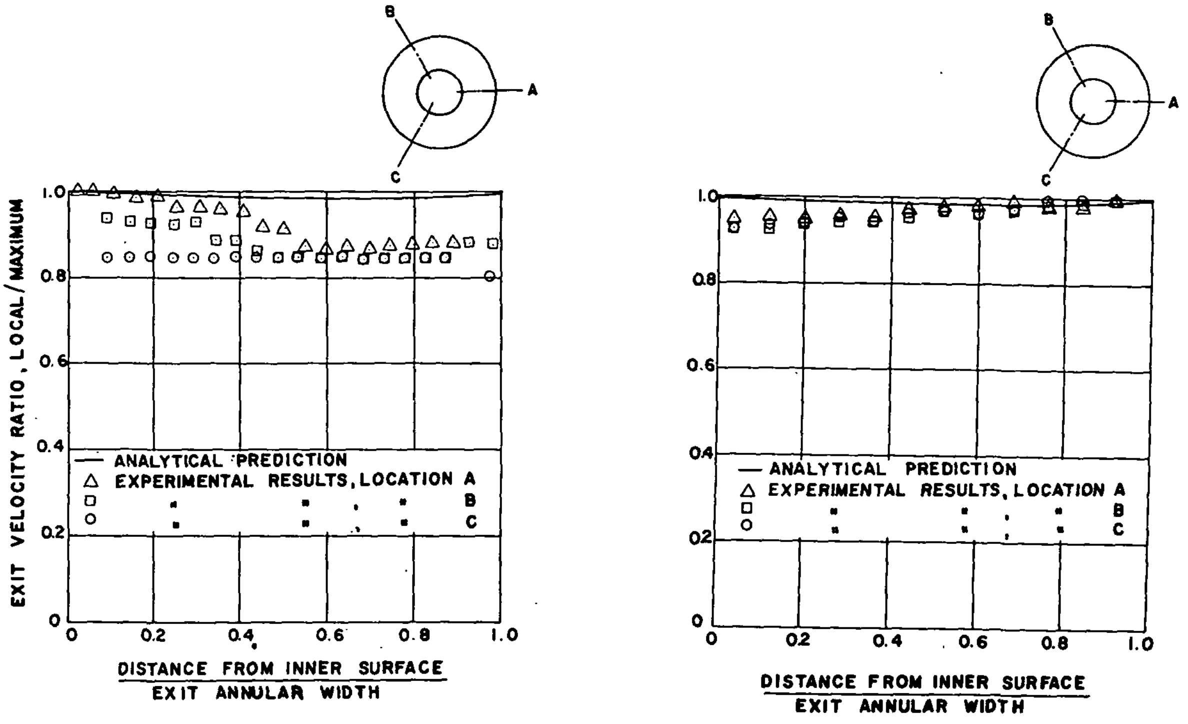
For inlet air velocities in the 200-250 ft/sec (61-76 m/sec) range, metastable operation required 6.5% suction for the bell shaped diffuser while stable operation required 12% suction. Corresponding suction rates for the 9-inch (23 cm) annular diffuser (3:1 AR) were 8.5% for metastable and 16.0% for stable operation. For the 9-inch (23 cm) annular diffuser with 2.5:1 area ratio, 6.3% and 12% suction rates were required respectively for metastable and stable operation. For the most recently designed diffuser, the 18-inch (46 cm) annular (AR = 3:1), 7% suction was required for metastable operation and 10% for stable operation at an inlet velocity of 230 ft/sec (70 m/sec). For lower inlet velocities the suction rate increased as shown in figure 10a.

5.2 Velocity Distributions

Figure 11 shows the exit plane velocity profiles for the four diffusers when an approximately uniform inlet velocity of 200 ft/sec (61 m/sec) is used and unseparated flow exists. Three sets of velocity measurements resulting from three circumferential locations 120° apart are shown for each diffuser. The velocity distribution predicted by the digital computer analysis program is also shown.

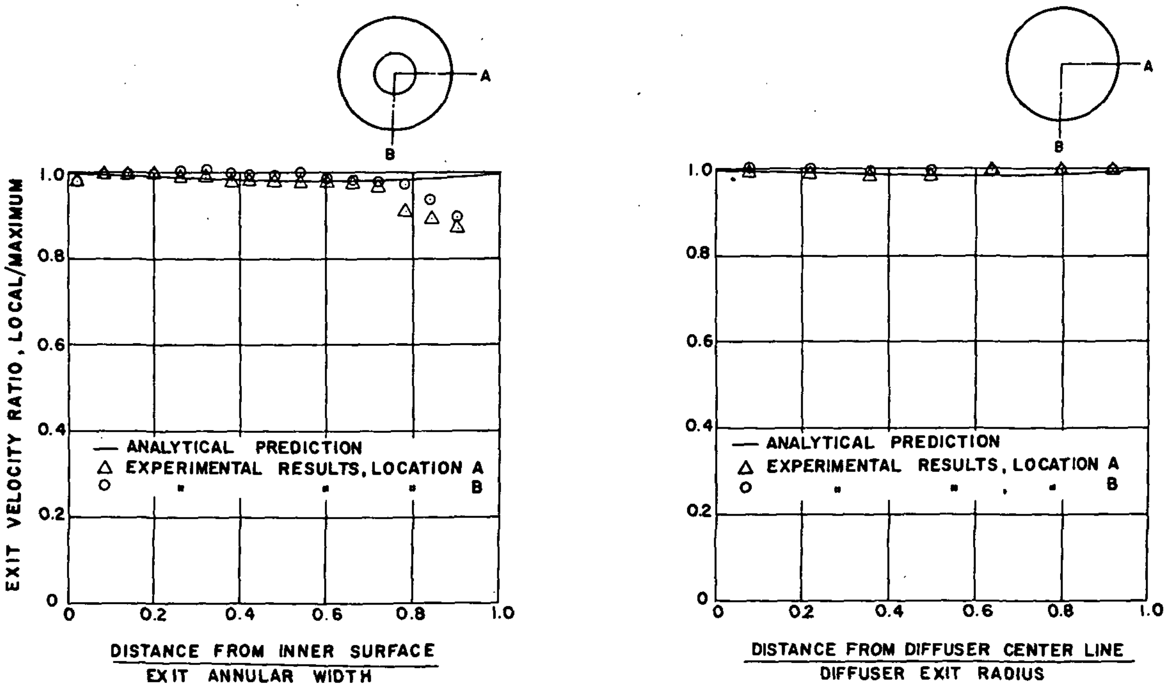
The dominant feature shown by this set of results is the virtually flat exit velocity profile of each of the four diffusers. For the bell shaped diffuser, the velocity ratio (local exit velocity/maximum exit velocity) was 0.98 ± 0.02 over 95% of the exit area. For the 9-inch (23 cm) annular (AR = 3:1) the exit velocity ratio was 0.96 ± 0.04 over 95% of the exit area. For the 9-inch (23 cm) annular (AR = 2.5:1) the exit velocity ratio was 0.93 ± 0.07 over 95% of the exit area. For the 18-inch (46 cm) annular diffuser the exit velocity ratio was 0.91 ± 0.09 over 95% of the exit area. The computer analysis program predicted a virtually flat exit velocity profile for each diffuser as shown.

Figure 12 shows the "wall velocity" distribution for unseparated flow through the four diffusers when an approximately uniform inlet velocity of



a. 18-INCH(46 cm) ANNULAR DIFFUSER AR=3:1

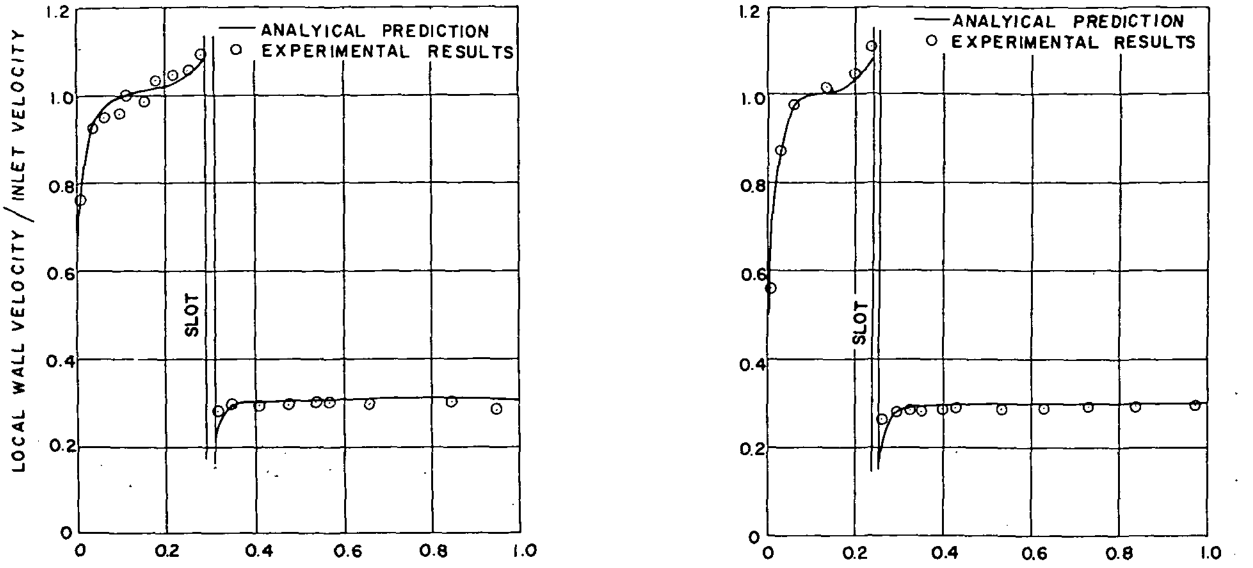
b. 9-INCH(23 cm) ANNULAR DIFFUSER AR=3:1



c. 9-INCH(23 cm) ANNULAR DIFFUSER AR=2.5:1

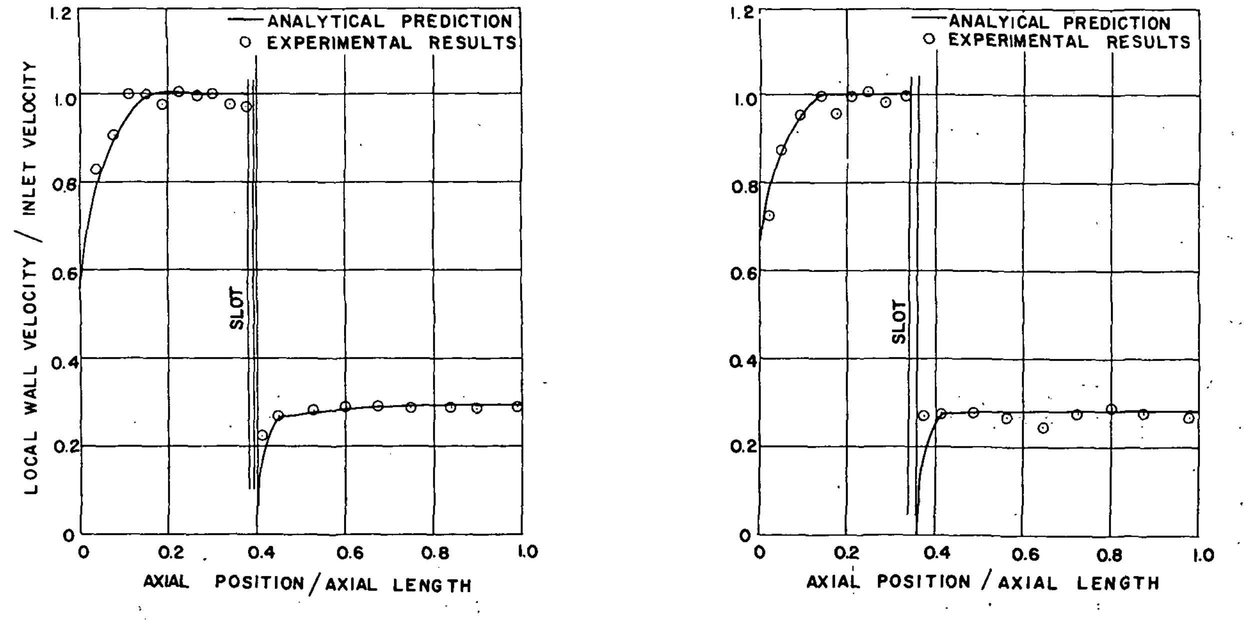
d. BELL SHAPED DIFFUSER AR=3:1

Figure 11. Exit plane velocity distribution for unseparated flow when using uniform inlet velocity.



a. 18-INCH(46cm) ANNULAR DIFFUSER UPPER WALL AR=3:1

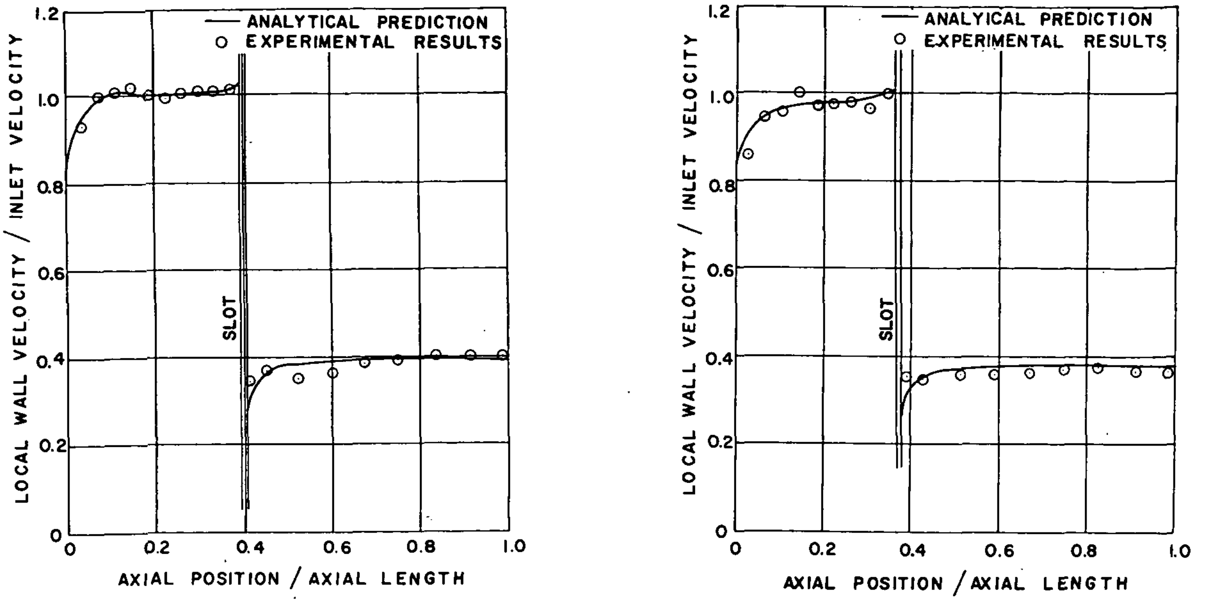
b. 18-INCH(46cm) ANNULAR DIFFUSER LOWER WALL AR=3:1



c. 9-INCH(23cm) ANNULAR DIFFUSER UPPER WALL AR=3:1

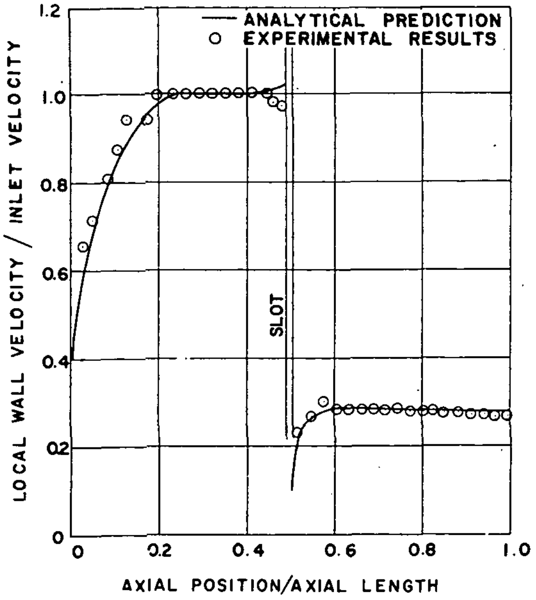
d. 9-INCH(23cm) ANNULAR DIFFUSER LOWER WALL AR=3:1

Figure 12. Wall velocity distribution for unseparated flow when using uniform inlet velocity.



e. 9-INCH (23 cm) ANNULAR DIFFUSER UPPER WALL AR=2.5 : 1

f. 9-INCH (23 cm) ANNULAR DIFFUSER LOWER WALL AR=2.5 : 1



g. BELL SHAPED DIFFUSER AR=3 : 1

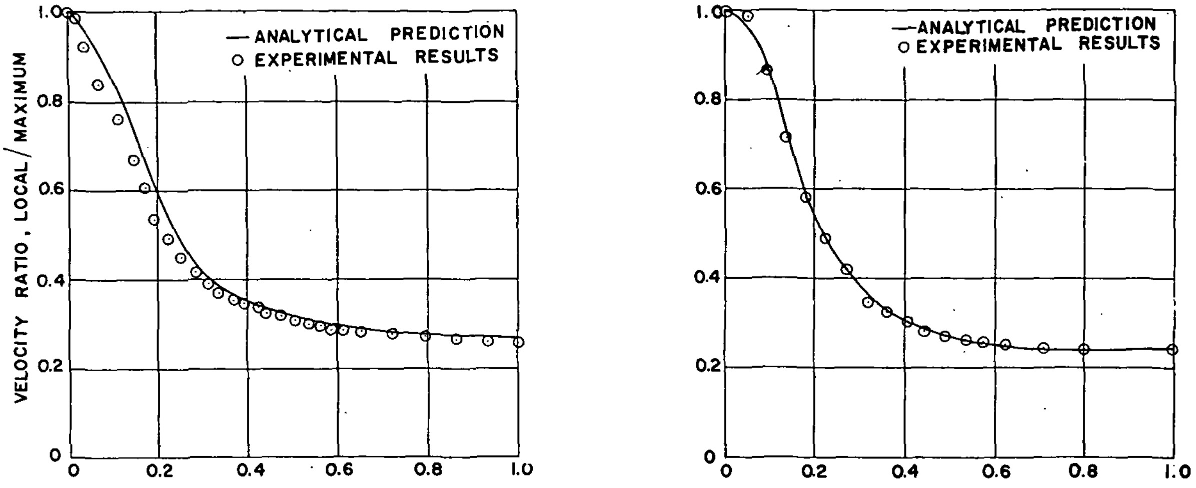
Figure 12 (continued). Wall velocity distribution for unseparated flow when using uniform inlet velocity.

200 ft/sec (61 m/sec) exists. The "wall velocities" were not measured directly but were computed from pressure coefficients obtained from wall pressure taps. Thus they are velocities near the wall just outside the boundary layer. For the three annular diffusers velocity distributions are shown for both the inner and the outer wall. The wall velocity distribution predicted by the computer analysis program is also shown for each diffuser. There is generally good agreement between the wall velocity distribution predicted and that determined from wall pressure measurements.

The "wall velocity" distributions show the design philosophy of the Griffith-type diffuser. There is virtually no deceleration along the diffuser walls and thus the "wall velocity" must change drastically at the slot.

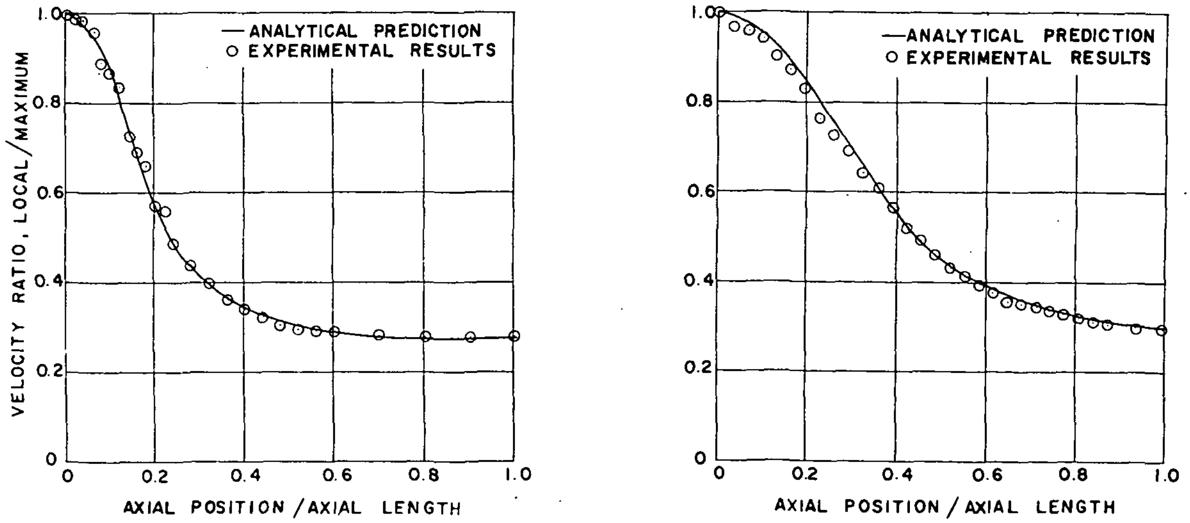
One may note the difference between the "wall velocity" distribution and the "centerline velocity" distribution shown by Figure 13. The centerline profiles were determined from velocity measurements along a constant radius near the middle of the flow passage. The deceleration at the center of the flow passage is seen to occur over a significant part of the diffuser length in contrast with the wall deceleration. For the centerline profile the axial position was taken from the diffuser inlet (smallest cross-sectional area), while for the wall velocity profile, the axial length included the nozzle or accelerating portion of the inlet section.

If the velocity distribution were precisely uniform at both the inlet and the exit planes, the velocity ratio along the passage centerline would drop from 1.0 at inlet to [(1.0 - FS/AR] at exit. For example, for the 9-inch (23 cm) annular diffuser (AR = 3:1) the velocity ratio would, with uniform inlet and exit velocity distribution, drop from 1.0 to (1.0 - 0.21)/3.0 or 0.263 since 21% suction was applied. The measured velocity ratio dropped from 1.0 to 0.24. A suction rate higher than the minimum required for stable



a. 18-INCH (46 cm) ANNULAR DIFFUSER AR = 3:1

b. 9-INCH (23 cm) ANNULAR DIFFUSER AR = 3:1



c. 9-INCH (23 cm) ANNULAR DIFFUSER AR = 2.5 : 1

d. BELL SHAPED DIFFUSER AR = 3:1

Figure 13. Axial velocity profile along a constant radius near the middle of the flow passage.

operation was used in order to minimize the fluctuation of the hot wire measurement caused by wakes shed from the stem of the probe.

5.3 Effect of Partial Blockage Downstream

Figure 14 shows the minimum suction requirement for unseparated flow through the 18-inch (46 cm) annular diffuser when a perforated sheet metal plate was attached to the diffuser exit plane. The plate had 5/16-inch (0.825 cm) diameter holes resulting in approximately 51% free area and simulated a partial blockage or obstruction downstream of the diffuser. There was a slight increase in the required suction rate at low inlet velocities when compared with no downstream obstruction. At inlet velocities above 150 ft/sec (46 m/sec) no significant effect was observed. The velocity distribution just upstream of the perforated plate was determined to be very similar to the exit plane velocity distribution when no downstream obstruction existed.

Three struts located 120° apart were mounted in the downstream section of the 9-inch (23 cm) (AR = 2.5:1) annular diffuser to simulate flow interference due to structural members. Figure 15 shows the exit plane circumferential velocity profile when this diffuser was operated with an approximately uniform inlet velocity of 200 ft/sec (61 m/sec). Approximately 10% suction was used to prevent flow separation. The velocity ratio drops to about 0.92 as the strut location is approached. This amount of variation was no more than the radial velocity variation when no struts were present. By taking advantage of the nearly constant velocity zone for the location of the struts, the problem of flow separation was avoided.

5.4 Effect of Non-Uniform Inlet Velocities

A series of tests were made to determine the effect of non-uniform diffuser inlet velocities on the 9-inch (23 cm) annular diffuser (AR = 2.5:1). Figure 16 shows the non-uniform duct discharge velocity profiles when a plate in the

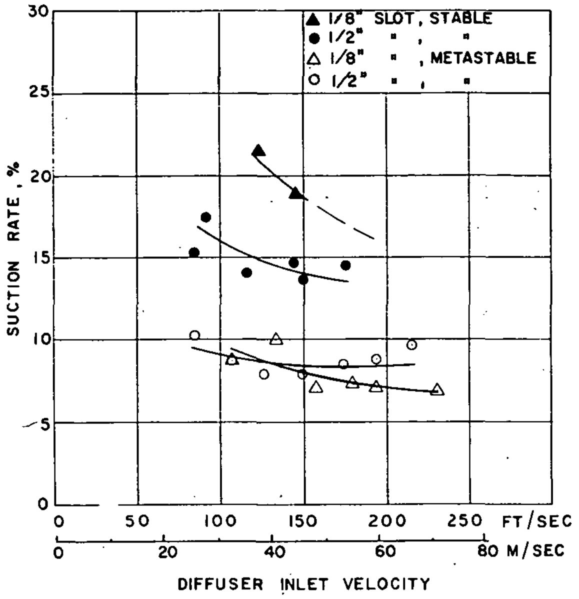


Figure 14. Minimum suction requirement for unseparated flow through 18-inch (46 cm) diffuser having a perforated metal plate at the exit plane.

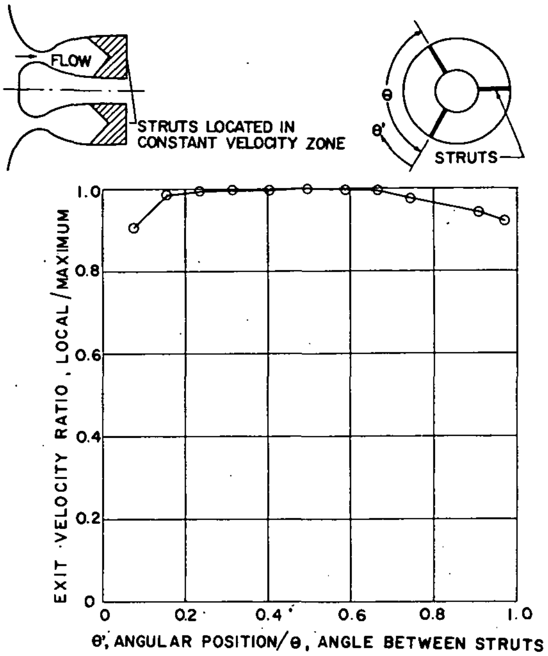


Figure 15. Exit plane circumferential velocity profile when three struts were mounted in the downstream section of the 9-inch (23 cm) (AR=2.5:1) diffuser.

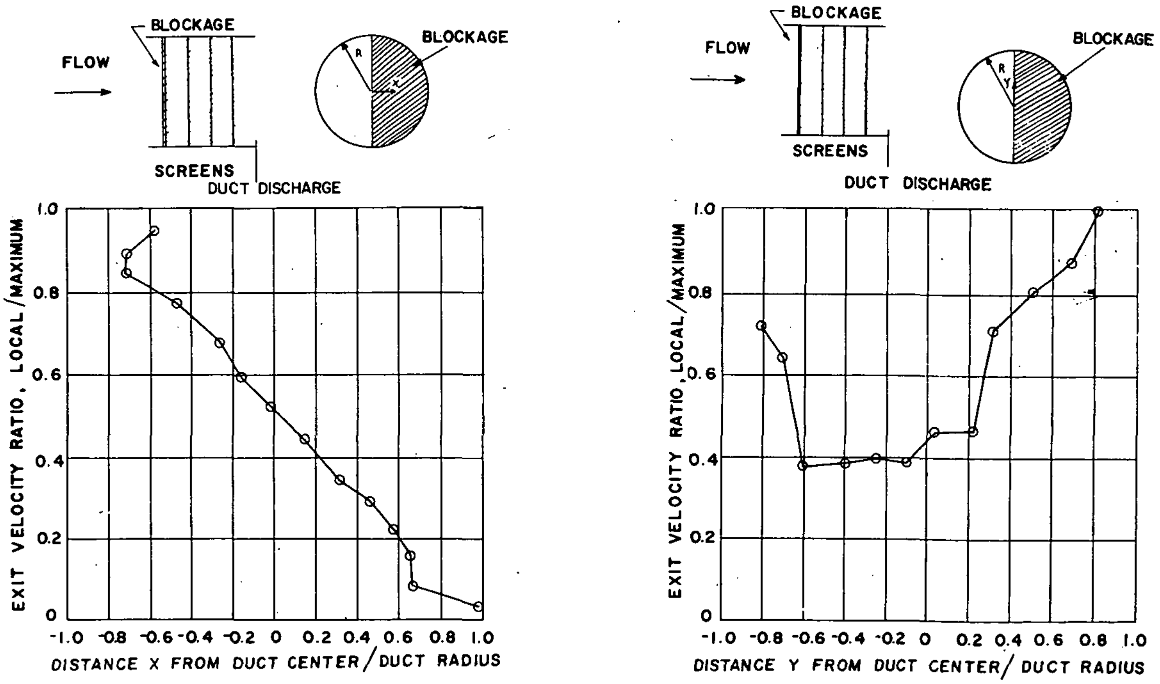


Figure 16. Duct discharge velocity profiles when partial blockage consisting of a semi-circular plate was installed.

shape of a half circle was mounted in the supply duct. These duct discharge velocity profiles were determined before the diffuser was attached to the end of the duct. Subsequent diffuser exit velocity measurements were made after installing the diffuser at the end of the partially blocked air duct. Figure 17 shows the resulting exit radial velocity distribution at two circumferential locations 90° apart. The velocity ratio variation was from a maximum near the inner wall to about 82% of maximum near the outer wall. This variation was only slightly more than when a uniform inlet velocity was used.

Figure 18 shows the non-uniform duct discharge velocity profile when a 14-inch (35 cm) circular plate was installed in the supply duct so as to yield a radially non-uniform duct discharge. Figure 19 shows the diffuser exit velocity profiles which were subsequently determined. As with the circumferential distortion, the diffuser exit radial velocity distribution was determined at two locations 90° apart. The velocity ratio variation was from a maximum near the inner wall to about 83% of maximum near the outer wall. The suction rates required for unseparated flow were approximately the same as with previous tests using uniform inlet velocities.

5.5 Effect of Unsteady Inlet Velocity

As previously mentioned, the effect of an unsteady inlet velocity was examined by installing a motor driven propeller which caused wakes to be shed upstream of the 9-inch (23 cm) annular diffuser inlet.

Figure 20 shows the minimum suction requirement for stable operation without and with the propeller. The shedding of wakes from the propeller caused an increase in the suction requirement, but the difference tends to become less as the inlet air velocity is increased. At 200 ft/sec (61 m/sec) the suction requirement for stable operation was increased from 16% to 18% by the propeller.

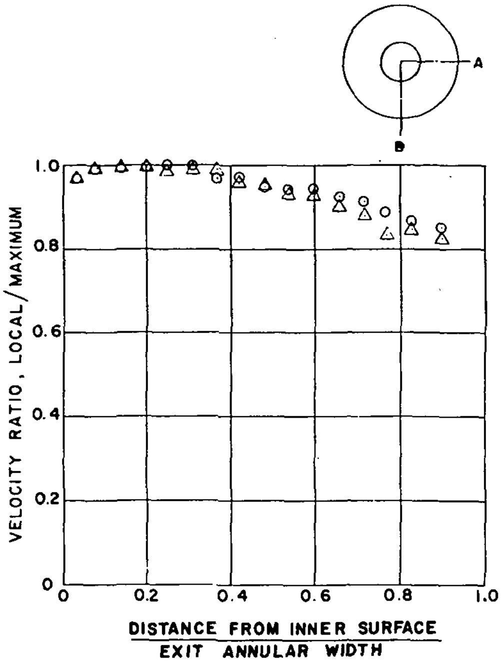


Figure 17. Exit radial velocity profiles of 9-inch (23 cm) diffuser (AR=2.5:1) when supply duct was partially blocked as shown by Fig. 16.

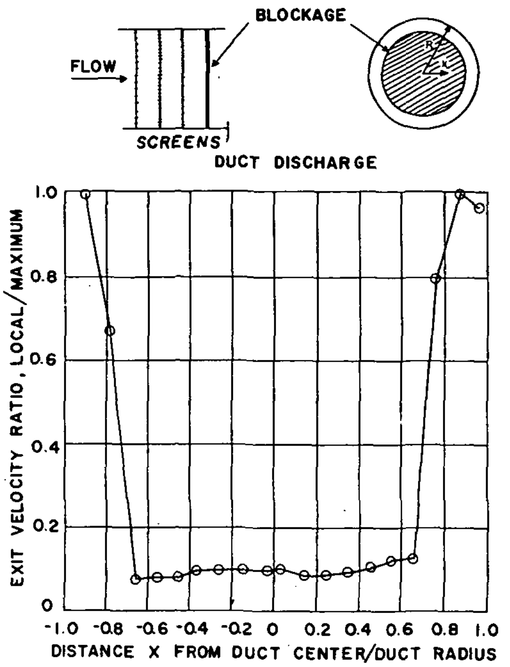


Figure 18. Duct discharge velocity profile when a circular plate was installed at center of duct passage.

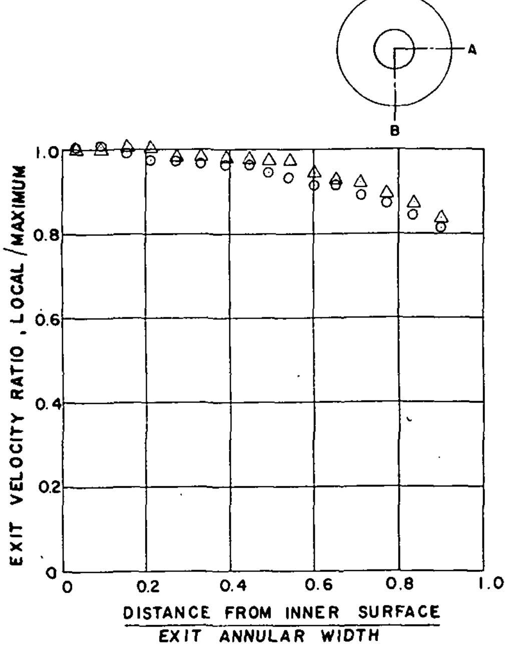


Figure 19. Exit radial velocity profile of 9-inch (23 cm) diffuser (AR=2.5:1) when supply duct was partially blocked as shown by Fig. 18.

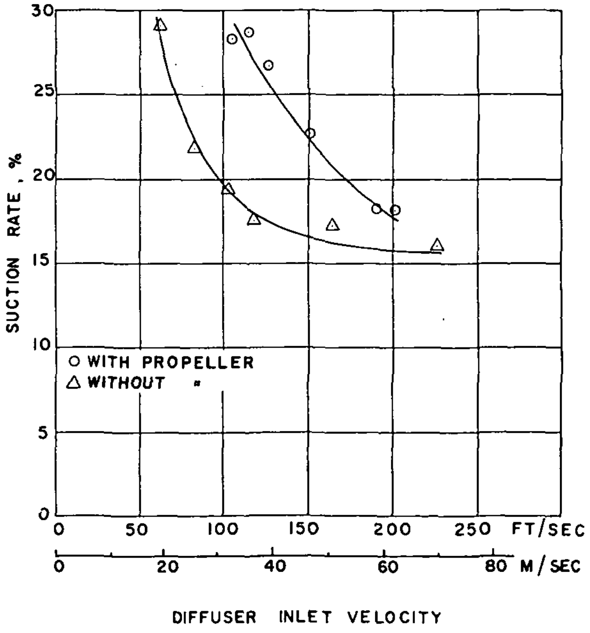


Figure 20. Minimum suction requirement for stable unseparated flow through 9-inch (23 cm) diffuser (AR=3:1) without and with a propeller.

5.6 Diffuser Performance

Tests were run to determine the effectiveness of the various Griffith diffusers. The effectiveness of the Griffith-type diffuser should be high when operated without flow separation because it is a very short diffuser and thus friction and turbulence losses should be very low. The price which must be paid is the suction loss, and for the effectiveness to have real meaning, this loss should be accounted for.

For this reason, this report will show both the conventional diffuser effectiveness, η (= actual pressure rise/ideal pressure rise) and a more meaningful effectiveness called ε. ε = (flow rate out)(actual pressure rise)/(flow rate in)(ideal pressure rise). Appendix C derives the following expressions used in computing η and ε.

$$\eta(\text{conventional}) = \frac{(P_{s,e} - P_{s,i})}{P_{d,i}\left[\kappa_i - \kappa_e\left(\frac{1.0 - FS}{AR}\right)^2\right]} \tag{24}$$

$$\varepsilon(\text{modified}) = \frac{(1.0 - FS)(P_{s,e} - P_{s,i})}{P_{d,i}\left[\kappa_i - \kappa_e\left(\frac{1.0 - FS}{AR}\right)^2\right]} \tag{25}$$

where FS is the fraction removed by suction and κ is the kinetic energy coefficient to account for the non-uniformity of the flow at the diffuser inlet and exit planes. Typical values of κ_i and κ_e were 1.025 and 1.011 respectively.

The logic of modifying the conventional effectiveness by the ratio of the exit flow rate to the inlet flow rate becomes evident when one considers that the conventional definition represents a ratio of energy terms per unit mass flowing. When no suction or injection occurs, flow rates in and out cancel.

The total pressure loss (see Appendix C) was computed from

$$P_{t,i} - P_{t,e} = (1-\eta)\, P_{d,i} \left[\kappa_i - \kappa_e \left(\frac{1.0 - FS}{AR} \right)^2 \right] \qquad (26)$$

and non-dimensionalized to the inlet dynamic pressure $P_{d,i}$.

Table I shows the results of the performance tests which were made on the four diffusers. In the 150-250 ft/sec (46-76 m/sec) range of inlet velocities, the diffuser effectiveness values based on the conventional definition were from 97% to 99%. When slot suction was properly charged to the diffusers, the effectiveness values were about 86% for the 9-inch (23 cm) annular (AR = 3:1), 88% for the 9-inch (23 cm) annular (AR = 2.5:1), 89% for the 18-inch (46 cm) annular, and 91% for the bell shaped diffuser. A typical total pressure loss was 2% of the inlet dynamic pressure. Fluctuations from this value were probably due more to slight pressure measurement errors than to actual changes in the total pressure loss.

TABLE I. GRIFFITH DIFFUSER PERFORMANCE

Test No.	Inlet Velocity Ft./Sec	Inlet Velocity M/Sec	Inlet Reynolds Number	Slot Suction %	Pressure Recovery Coefficient C_p	Effectiveness ε% (Eq. 25)	Effectiveness η(%) (Eq. 24)	Total Pressure Loss, % of Inlet Dynamic (Eq. 26)
A. Bell Shaped Diffuser; AR=3:1; Slot Width = 1/16 in. (.159 cm)								
1	82	25	4.2×10^4	7.3	0.92	91.5	98.8	1.1
2	145	44	7.4×10^4	6.2	0.92	93.6	99.0	0.9
3	235	72	12.0×10^4	6.3	0.91	92.0	98.2	1.7
B. 9 in. (23 cm) Annular Diffuser; AR=3:1; Slot Width = 1/16 in. (.159 cm)								
4	79	24	2.0×10^4	16.6	0.90	81.3	95.2	4.5
5	142	43	3.6×10^4	12.3	0.91	85.5	97.5	2.3
6	236	72	6.0×10^4	11.2	0.91	86.5	97.5	2.3
C. 9 in. (23 cm) Annular Diffuser; AR=2.5:1; Slot Width = 1/16 in. (.159 cm)								
7	82	25	2.8×10^4	12.8	0.89	86.1	98.8	1.1
8	147	45	5.0×10^4	11.0	0.87	86.6	97.4	2.2
9	237	72	8.0×10^4	9.5	0.88	89.5	99.0	0.9
D. 18 in. (46 cm) Annular Diffuser; AR=3:1; Slot Width = 1/2 in. (1.27 cm)								
10	84	26	4.3×10^4	10.3	0.90	86.7	96.7	3.1
11	144	44	7.3×10^4	8.3	0.92	90.9	99.0	0.9
12	229	70	11.7×10^4	10.8	0.93	88.5	99.0	0.9

SECTION 6
CONCLUSIONS

The following conclusions were reached based upon the results of the research program.

(1) The concept of the Griffith diffuser is a reasonable approach to designing short axially symmetrical diffusers for special applications.

(2) The potential flow design program utilizing the two dimensional wedge flow patch to approximate the branch flow from the suction slot yields a workable curved wall diffuser. This is not to rule out the possibility of improvement by using a more sophisticated design program such as an improved branch flow patch and the accounting for boundary layer displacement thickness.

(3) Reasonably good agreement was achieved among the velocity distributions prescribed to the design program, predicted by the analysis program, and determined experimentally.

(4) The slot suction rate required to prevent separation with stable flow was found to decrease asymptotically with diffuser inlet velocity and radial dimensions. Thus the minimum suction rate was about 10% at 230 fps (69 m/sec) for the largest diffuser tested.

(5) High performance can be expected from the Griffith diffuser when sufficient suction is used to prevent flow separation. An effectiveness of 88% is a typical value if slot suction is accounted for.

(6) A reasonably uniform exit velocity profile results from the Griffith type diffuser when sufficient suction to prevent flow separation is used.

REFERENCES

1. Yang, T, and Hudson, W. G., "Proposed Design and Experimental Performance of Short Two Dimensional Curved Wall Diffusers Utilizing Suction Slots," NASA Contractor Report CR-120783, September, 1971.

2. Lachmann, G. V., "Boundary Layer and Flow Control," 1st ed., Vol. 1, Pergamon Press, 1961, p. 119.

3. Manoni, L. R., "Wide Angle Diffusers Employing Boundary Layer Control," United Aircraft Corporation Research Department Report R-95-460-12, 1952.

4. Stanitz, J. D., "Design of Two-Dimensional Channels with Prescribed Velocity Distributions Along the Channel Walls," NACA Report 1115, 1953.

5. Nelson, C. D., "An Analytical Design Method and Experimental Results for Axially Symmetrical Diffusers Having Incompressible, Unseparated Flow and Employing Slot Suction," Ph. D. Dissertation, Mechanical Engineering Department, Clemson University, August, 1971.

6. Smith, A. M. O., and Pierce, J., "Exact Solution of Neumann Problem. Calculation of Non-Circulatory Plane and Axially Symmetric Flows About or Within Arbitrary Boundaries," Douglas Aircraft Company Report No. 26988, 1968.

7. Goldstein, Sydney, "Low Drag and Suction Airfoils," Journal of Aeronautical Sciences, Vol. 15, No. 4, April, 1948.

APPENDIX A - PROGRAM DESCRIPTION AND LISTING

The main program is the controller for the program. It calls each of the subroutines as needed, and it contains the main loop for the iteration of the radial coordinates.

SETUP is a subroutine which computes coefficients which are needed to perform Simpson's Rule integration.

SIMP is a subroutine for integration using Simpson's Rule.

PART I is a subroutine which reads in, checks, and outputs the input data. Obvious errors in input data terminates the program after error explanation is printed. Certain constants to control subsequent calculations are computed and initialization of arrays Q and R are also performed in PART I.

COEF is a subroutine which computes constants used in the solution for the velocity distribution.

RELAX is a subroutine which uses the Gauss-Siedel Technique with overrelaxation to solve for ln Q over the region. Where derivatives of the velocity are prescribed as zero, this subroutine corrects those boundary conditions accordingly.

PART II is a subroutine which performs the inverse transformation to get the flow angles, α, the R coordinates, and finally the X coordinates if the iteration on R is on its last pass. This subroutine also corrects the boundary conditions on R where required. The integration is carried out first along the middle streamline and then along each potential line individually.

OUTPUT is a subroutine which prints out the R array upon request during the iteration on R. At the completion of all computations, it outputs the Q, R, X, and α arrays and a summary of the wall coordinates of the finished channel design.

GRAPHIC is a subroutine which plots out the streamlines and equipotential lines for the final channel design. Inlet and outlet channel velocity profiles are optionally output.

INPUT DATA PREPARATION INSTRUCTIONS

The following is a description of the input data, their meanings, and how they must appear on the data cards.

Control Data

CARD 1

Column	Variable	Description
1	FLAG1	FLAG1 is the input flag for the plane flow case. FLAG1 = 0 for axially symmetric flow. FLAG1 = 1 for plane flow.
2	FLAG2	FLAG2 is the boundary condition flag. FLAG2 = 0: Dirichlet B.C. are used on the inlet and outlet potential lines and the velocities are set equal to the first and last wall velocities input respectively. FLAG2 = 1: Neumann B.C. are used on the inlet and outlet potential lines. FLAG2 = 2: Dirichlet B.C. are used on the inlet potential line with the velocities being input there and Neumann B.C. are used on the outlet potential line. FLAG2 = 3: Neumann B.C. are used on the inlet potential line and Dirichlet B.C. are used on the outlet potential line with the velocities being input there. FLAG2 = 4: Dirichlet B.C. are used on the inlet and outlet potential lines with the velocities being input on both.
3	FLAG3	FLAG3 is the bell diffuser flag. FLAG3 = 0 if the diffuser is plane or annular. FLAG3 = 1 if the diffuser is a bell diffuser.
4	FLAG4	FLAG4 is the graphical output flag. FLAG4 = 0 if no graphical output is desired. FLAG4 = 1 if graphical output is desired.

Column	Variable	Description
7-66	HEDR	HEDR is a 60 character comment that will be printed out at the top of each page of output.
69-70	DATE1	DATE1 is the two digit number giving the month of the date.
71-72	DATE2	DATE2 is the two digit number giving the day of the date.
73-74	DATE3	DATE3 is the two digit number giving the year of the date.
77-80	CASE	CASE is a four digit case number that will be printed out at the top of each page of output.

CARD 2

Column	Variable	Description
1-10	RMULT	RMULT is a constant by which all computed X and Y coordinates will be multiplied at the completion of the program. If RMULT = 0.0 no multiplication will take place.
11-20	W	W is the relaxation factor. Although the optimum value varies with the grid size, a value of 1.9 has been found to work well in planar cases, 1.9 in bell cases, and 1.7 in annular cases.
21-30	TOLSYS	TOLSYS is the accuracy to which the radial coordinates are to be computed. This accuracy is determined as the largest change of any radial coordinate from the previous iteration.
31-40	TOLDE	TOLDE is the accuracy to which the velocity is to be computed. This accuracy is determined as the largest change of any velocity from the previous iteration.
41-50	MSTAGU	This is the number of the nodal point in the ϕ direction at which the upper stagnation point (slot downstream lip) is to be located.
51-60	MSTAGL	This is the number of the nodal point in the ϕ direction at which the lower stagnation point is to be located. For plane and bell designs this item is not used.

CARD 3

Column	Variable	Description
1-10	M	The number of mesh points to be used in the ϕ direction (The maximum number 148.)

Column	Variable	Description
11-20	N	The number of mesh points to be used in the ψ direction. (The maximum number is 48.)
21-30	MXISYS	The maximum number of iterations allowed to find the radial coordinates. If FLAG1 = 1 then MXISYS = 1. MXISYS = 10 gives good results for annular and bell diffusers.
31-40	MXIDE	The maximum number of iterations allowed to find the velocities; 300-400 usually gives good results for the 2-D case and 50 gives good results for the axially symmetric case.
41-50	NSTAGU	This is the number of nodal points in the ψ direction at which the upper stagnation point (slot downstream lip) is located. N-NSTAGU cannot be less than 3.
51-60	NSTAGL	This is the number of nodal points in the ψ direction at which the lower stagnation point is located. For plane and bell designs this item is not used. NSTAGL cannot be less than 4.

CARD 4

Column	Variable	Description
1-10	RADIN	RADIN is the radial coordinate at the nodal point numbered (1,1) for annular design. If left blank its value is assumed to be 1.0. This item is not used in plane or bell designs.
11-20	MIDJ	The number of the streamline for an annular diffuser. Usually chosen as N/2. Care should be exercised in the choise of MIDJ since an improper choice can cause an annular design to cross the X axis. This results in a divide check, overflow or underflow and a consequent termination of the program. This item is only used for annular designs.
21-30	IPRSYS	The frequency at which intermediate values of the coordinates of the diffuser are printed out during iteration on R. If no intermediate values are desired, make IPRSYS greater than MXIDE.
31-40	IPRDE	The frequency at which intermediate values of the velocity are printed out.
41-50	MSLOTU	The number of the potential line at the bottom of the upper slot. This number is usually only slightly larger than MSTAGU.
51-60	MSLOTL	The number of the potential line at the bottom of the lower slot. This item is only used for annular designs.

Spacing and Boundary Conditions

The following variables are input six to a card in columns 1-10, 11-20, etc.

Variable	Description
PHI	The values of the velocity potential along a streamline at each nodal point. There should be M values.
Q (ϕ, Ψ_1) upstream	The values of the velocity along the upper wall upstream of and along the upstream wall inside of the slot. There should be MSLOTU of these values.
Q (ϕ, Ψ_2) downstream	The values of the velocity along the upper wall downstream of the slot. There should be M-MSTAGU of these values.
QSLOTU (ϕ)	The values of the velocity along the downstream wall inside of the slot. There should be MSLOTU-MSTAGU of these values.
Q $(\phi, 0)$ upstream	The values of the velocity along the lower wall upstream of and along the upstream wall inside of the slot. There should be MSLOTL of these values. Omit for plane and bell designs.
Q (ϕ, Ψ_3) downstream	The values of the velocity along the lower wall downstream of the slot. There should be M-MSTAGL of these values. Omit for plant and bell designs.
QSLOTL (ϕ)	The values of the velocity along the downstream wall inside of the slot. There should be MSLOTL-MSTAGL of these values. Omit for plane and bell designs.
PSI	The values of the stream function along a potential line at each nodal point. There should be N values.
Q $(0, \psi)$	The value of the velocity at nodal points along the inlet potential line. There should be N values. Values are input only if FLAG2 = 2 or 4.
Q (Φ, ψ)	The values of the velocity at nodal points along the outlet potential line. There should be NSTAGU values for plane and bell designs and NSTAGU-NSTAGL + 1 values for annular designs. Values are input only if FLAG2 = 3 or 4.

OUTPUT EXPLANATION

On the first page of output the control data is printed out and the options called for by the control data are listed. Following the printout of the control data comes a printout of the wall boundary conditions and spacing in the ϕ direction. The wall boundary conditions are given in three parts: the inlet wall data, the downstream wall data and data inside the slot. The first column of each set of data gives the arc length, S, computed along the wall streamline. The second column labeled Q gives the velocity that is input along the wall streamline. The third column labeled PHI gives the values of ϕ as input. The fourth column labeled DPHI gives $\Delta\phi$, the spacing as computed by subtracting adjacent values of ϕ. For annular designs three more sets of wall boundary conditions are output for the diffuser lower wall.

Following the wall streamline boundary conditions the stream function distribution is printed out along with the spacing computed from it. The numbers to the left of these outputs give the number of the node to which the information corresponds.

Following the input data comes a list of the convergence variables ITER, JTER, EPS and DEL for each iteration on R (these are printed out for the axially symmetric case only). ITER is the number of the iteration on R that was then in progress. JTER is the number of iterations on the velocity that have been completed. EPS is the accuracy to which R has been computed. DEL is the accuracy of solution of the lnQ equation.

The arrays of Q, R, X and ALPHA are output after the last iteration. The arrays QSLOTU, QSLOTL, RSLOTU, RSLOTL, XSLOTU, XSLOTL, ASLOTU, and ASLOTL, the values of the corresponding variables inside the slot on the downstream wall, are also output. The values of ITER, JTER, EPS and DEL are output in the heading of each array.

A summary of the wall coordinates is output last along with the velocity that was prescribed along the wall, the arc length computed along the wall and the tangent angle along the wall. This output is presented in three parts as was the input data along the walls. For annular channels another set of output is presented for a lower wall.

The following 45 pages is a program listing for the design program.

```
C          **********     MAIN  PROGRAM     **********
C
C          CLEMSON  UNIVERSITY  MECHANICAL  ENGINEERING  DEPARTMENT
C
C          PROGRAM  70-02
C
C          AXIALLY SYMMETRIC AND 2-D BRANCHED FLOW CHANNEL DESIGN
C
      DIMENSION DX(150),ESIMP(150),FSIMP(150),GSIMP(150),
     1       GRAND(150),GIFT(150),PHI(150),PSI(150),
     2       DPSI(150),DPHI(150),S(150),SBAR(150),S1(150),
     3       SBAR1(150),S2(150),SBAR2(150),HEDR(15),
     4       EPHI(150),FPHI(150),GPHI(150),EPSI(150),
     5       FPSI(150),GPSI(150),RSLOTU(150),RSLOTL(150),
     6       QSLOTU(150),QSLOTL(150),ASLOTU(150),
     7       ASLOTL(150),XSLOTU(150),XSLOTL(150)
      DOUBLE PRECISION P
      INTEGER FLAG1,FLAG2,FLAG3,FLAG4,CASE,DATE1,DATE2,DATE3
      COMMON / C1 / IMINSU,IMAXSU,IMINSI,IMAXSI,GRAND,GIFT,
     1              DX,ESIMP,FSIMP,GSIMP
      COMMON / C2 / FLAG1,FLAG2,FLAG3,FLAG4,HEDR,DATE1,DATE2,
     1              DATE3,CASE,MXIDE,ICKDE,IPRDE,IMIN,IMAX,
     2              JMIN,JMAX,NUT,M,N,TOLDE,DEL,EPS,JTER,
     3              ITER,DPHI,DPSI,S,SBAR,RMULT,S1,S2,SBAR1,
     4              SBAR2,MSTAGU,NSTAGU,MSLOTU,MSTAGL,NSTAGL,
     5              MSLOTL,RADIN,W,PHI,PSI,RSLOTU,RSLOTL,
     6              QSLOTU,QSLOTL,ASLOTU,ASLOTL,XSLOTU,
     7              XSLOTL,G,MIDJ,TOLSYS
      COMMON / C3 / EPHI,FPHI,GPHI,EPSI,FPSI,GPSI,MXISYS,
     1              IPRSYS
      CALL PART1
      II = 1
      DO 100 ITER=1,MXISYS
      CALL COEF
      CALL RELAX
      CALL PART2
      IF (EPS .LE. TOLSYS) GO TO 200
      IF (ITER .LT. II*IPRSYS) GO TO 100
      NUT = 1
      II = II + 1
      CALL OUTPUT
  100 CONTINUE
  200 NUT = 0
      CALL OUTPUT
      IF (FLAG4 .LE. 0) STOP
      CALL GRAPIC
      END
C          **********     SUBROUTINE  SETUP     **********
C
C          COMPUTES COEFFICIENTS FOR USE IN SIMPSON RULE
C          INTEGRATION
C
      SUBROUTINE SETUP
      DIMENSION DX(150),ESIMP(150),FSIMP(150),GSIMP(150),
     1       GRAND(150),GIFT(150)
      COMMON / C1 / IMINSU,IMAXSU,IMINSI,IMAXSI,GRAND,GIFT,
     1              DX,ESIMP,FSIMP,GSIMP
      DO 1000 I=IMINSU,IMAXSU
      DX1 = DX(I)
      DX2 = DX(I+1)
```

```
      DX1Q = DX1 * DX1 * DX1                                            SET   14
      DX1SQ = DX1 * DX1                                                 SET   15
      DX2Q = DX2 * DX2 * DX2                                            SET   16
      DX2SQ = DX2 * DX2                                                 SET   17
      XNUM = 2. * DX1Q + 3. * DX1SQ * DX2 - DX2Q                        SET   18
      DEN = 6. * DX1 * ( DX2 + DX1 )                                    SET   19
      ESIMP(I) = XNUM / DEN                                             SET   20
      XNUM = DX2Q + DX1Q + 3. * DX2SQ * DX1 + 3. * DX1SQ * DX2          SET   21
      DEN = 6. * DX1 * DX2                                              SET   22
      FSIMP(I) = XNUM / DEN                                             SET   23
      XNUM = 2. * DX2Q + 3. * DX2SQ * DX1 - DX1Q                        SET   24
      DEN = 6. * DX2 * ( DX2 + DX1 )                                    SET   25
 1000 GSIMP(I) = XNUM / DEN                                             SET   26
      RETURN                                                            SET   27
      END                                                               SET   28
****
SIMP
C         **********      SUBROUTINE  SIMP      **********              SIMP   1
C                                                                       SIMP   2
C         THIS SUBROUTINE DOES SIMPSON'S RULE INTEGRATION               SIMP   3
C         OF A FUNCTION AND RETURNS THE INTEGRATED FUNCTION             SIMP   4
C                                                                       SIMP   5
      SUBROUTINE SIMP                                                   SIMP   6
      DIMENSION DX(150),ESIMP(150),FSIMP(150),GSIMP(150),               SIMP   7
     1          GRAND(150),GIFT(150)                                    SIMP   8
      COMMON / C1 / IMINSU,IMAXSU,IMINSI,IMAXSI,GRAND,GIFT,             SIMP   9
     1               DX,ESIMP,FSIMP,GSIMP                               SIMP  10
      GIFT(IMINSI) = 0.0                                                SIMP  11
      II = IMINSI + 2                                                   SIMP  12
C                                                                       SIMP  13
C         COMPUTE MAIN INTEGRAL                                         SIMP  14
C                                                                       SIMP  15
      DO 1000 I=II,IMAXSI,2                                             SIMP  16
      III = I                                                           SIMP  17
      SUM = GIFT(I-2) + ESIMP(I-2) * GRAND(I-2)                         SIMP  18
      SUM = SUM + FSIMP(I-2) * GRAND(I-1)                               SIMP  19
      SUM = SUM + GSIMP(I-2) * GRAND(I)                                 SIMP  20
 1000 GIFT(I) = SUM                                                     SIMP  21
C                                                                       SIMP  22
C         COMPUTE INTERMEDIATE VALUES                                   SIMP  23
C                                                                       SIMP  24
      IUP = IMAXSI                                                      SIMP  25
      IF (IMAXSI - III .EQ. 1) IUP = IMAXSI - 2                         SIMP  26
      II = IMINSI + 1                                                   SIMP  27
      DO 2000 I=II,IUP,2                                                SIMP  28
      D1 = DX(I-1)                                                      SIMP  29
      D2 = DX(I)                                                        SIMP  30
      GIFT(I) = D1 * ( 2. * D1 + 3. * D2 ) * GRAND(I-1) /               SIMP  31
     1( 6. * D2 + 6. * D1 )                                             SIMP  32
      GIFT(I) = GIFT(I) + D1 * ( D1 + 3. * D2 ) * GRAND(I) /            SIMP  33
     1( 6. * D2 )                                                       SIMP  34
 2000 GIFT(I) = GIFT(I) - D1 * D1 * D1 * GRAND(I+1) /                   SIMP  35
     1( 6. * D2 * D2 + 6. * D2 * D1 ) + GIFT(I-1)                       SIMP  36
      IF (IMAXSI - III .EQ. 0) RETURN                                   SIMP  37
      I = IMAXSI                                                        SIMP  38
      GIFT(I) = ( GRAND(I-1) + GRAND(I) ) * DX(I-1) / 2. + GIFT(I-1)    SIMP  39
      RETURN                                                            SIMP  40
      END                                                               SIMP  41
C         **********      SUBROUTINE  PART1      **********             PAR1   1
C                                                                       PAR1   2
C         THIS SUBROUTINE READS INPUT DATA AND PREPARES                 PAR1   3
C         ARRAYS FOR RELAXATION                                         PAR1   4
```

```
C
      SUBROUTINE PART1
      DIMENSION DX(150),ESIMP(150),FSIMP(150),GSIMP(150),
     1          GRAND(150),GIFT(150),PHI(150),PSI(150),
     2          DPSI(150),DPHI(150),S(150),SBAR(150),S1(150),
     3          SBAR1(150),S2(150),SBAR2(150),HEDR(15),
     4          EPHI(150),FPHI(150),GPHI(150),EPSI(150),
     5          FPSI(150),GPSI(150),RSLOTU(150),RSLOTL(150),
     6          QSLOTU(150),QSLOTL(150),ASLOTU(150),
     7          ASLOTL(150),XSLOTU(150),XSLOTL(150)
      DIMENSION Q(150,50),R(150,50),X(150,50)
      DOUBLE PRECISION P
      INTEGER FLAG1,FLAG2,FLAG3,FLAG4,CASE,DATE1,DATE2,DATE3
      COMMON / C1 / IMINSU,IMAXSU,IMINSI,IMAXSI,GRAND,GIFT,
     1              DX,ESIMP,FSIMP,GSIMP
      COMMON / C2 / FLAG1,FLAG2,FLAG3,FLAG4,HEDR,DATE1,DATE2,
     1              DATE3,CASE,MXIDE,ICKDE,IPRDE,IMIN,IMAX,
     2              JMIN,JMAX,NUT,M,N,TOLDE,DEL,EPS,JTER,
     3              ITER,DPHI,DPSI,S,SBAR,RMULT,S1,S2,SBAR1,
     4              SBAR2,MSTAGU,NSTAGU,MSLOTU,MSTAGL,NSTAGL,
     5              MSLOTL,RADIN,W,PHI,PSI,RSLOTU,RSLOTL,
     6              QSLOTU,QSLOTL,ASLOTU,ASLOTL,XSLOTU,
     7              XSLOTL,G,MIDJ,TOLSYS
      COMMON / C3 / EPHI,FPHI,GPHI,EPSI,FPSI,GPSI,MXISYS,
     1              IPRSYS
      REWIND 4
      REWIND 5
      REWIND 12
C
C         READ  CONTROL  DATA
C
      READ(1,10)FLAG1,FLAG2,FLAG3,FLAG4,HEDR,DATE1,DATE2,
     1          DATE3,CASE,RMULT,W,TOLSYS,TOLDE,MSTAGU,MSTAGL,
     2          M,N,MXISYS,MXIDE,NSTAGU,NSTAGL,RADIN,MIDJ,
     3          IPRSYS,IPRDE,MSLOTU,MSLOTL
   10 FORMAT(4I1,2X,15A4,2X,3I2,2X,I4/4F10.0,2I10/6I10/
     1          F10.0,5I10)
C
C         OUTPUT  CONTROL  DATA  AND  OPTIONS
C
      WRITE(3,20)
   20 FORMAT(1H1//T57,'CLEMSON  UNIVERSITY'//T49,
     1'MECHANICAL  ENGINEERING  DEPARTMENT')
      WRITE(3,30)HEDR,DATE1,DATE2,DATE3,CASE,FLAG1,RMULT,
     1          M,RADIN,FLAG2,W,N,MIDJ,FLAG3,TOLSYS,MXISYS,
     2          IPRSYS,FLAG4,TOLDE,MXIDE,IPRDE,MSTAGU,NSTAGU,
     3          MSLOTU,MSTAGL,NSTAGL,MSLOTL
   30 FORMAT(//T59,'PROGRAM  70-02'//T38,
     1'AXIALLY  SYMMETRIC  AND  2-D  BRANCHED  CHANNEL',
     2T85,'  DESIGN'///T11,'*****',T21,15A4,T84,I2,T86,'/',
     3T87,I2,T89,'/',T90,I2,T96,'CASE  NO.  ',T107,I4,
     4T116,'*****',/////T51,'------  CONTROL  DATA  ------'
     5/////T16,'FLAG1 = ',T23,I1,T36,'RMULT = ',T45,F11.6,
     6T66,'M       = ',T75,I10,T96,'RADIN  = ',T105,F10.5/
     7T16,'FLAG2 = ',T23,I1,T36,'W       = ',T45,F11.6,T66,
     8'N       = ',T75,I10,T96,'MIDJ   = ',T105,I10/T16,
     9'FLAG3 = ',T23,I1,T36,'TOLSYS = ',T45,F11.6,T66,
     A'MXISYS = ',T75,I10,T96,'IPRSYS = ',T105,I10/T16,
     B'FLAG4 = ',T23,I1,T36,'TOLDE  = ',T45,F11.6,T66,
     C'MXIDE  = ',T75,I10,T96,'IPRDE  = ',T105,I10/T36,
```

```
     D'MSTAGU =  ',T46,I10,T66,'NSTAGU = ',T75,I10,
     ET96,'MSLOTU = ',T105,I10/T36,'MSTAGL =  ',T46,I10,
     FT66,'NSTAGL = ',T75,I10,T96,'MSLOTL = ',T105,I10////
     GT55,'------  OPTIONS  ------')
      EPS = 0.0
      IF (FLAG1 .GT. 0) WRITE(3,40)
   40 FORMAT(///T57,'2-DIMENSIONAL')
      IF (FLAG1 .LE. 0) WRITE(3,50)
   50 FORMAT(///T57,'AXIALLY  SYMMETRIC')
      IF ( FLAG2 .EQ. 0 .OR. FLAG2 .EQ. 4 ) WRITE(3,55)
   55 FORMAT(/T57,'DIRICHLET  BOUNDARY  CONDITIONS')
      IF ( FLAG2 .EQ. 1 ) WRITE(3,60)
   60 FORMAT(/T57,'NEUMANN  BOUNDARY  CONDITIONS')
      IF ( FLAG2 .EQ. 2 ) WRITE(3,65)
   65 FORMAT(/T57,'DIRICHLET B. C. AT INLET AND NEUMANN ',
     1T94,'B. C. AT OUTLET',/T60,'WITH VELOCITY INPUT AT INLET')
      IF ( FLAG2 .EQ. 3 ) WRITE(3,70)
   70 FORMAT(/T57,'NEUMANN B. C. AT INLET AND DIRICHLET ',
     1T94,'B. C. AT OUTLET',/T60,'WITH VELOCITY INPUT AT OUTLET')
      IF (FLAG3 .GT. 0) WRITE(3,80)
   8C FORMAT(/T57,'BELL  DIFFUSER - NOZZLE')
      IF (FLAG3 .LE. 0 .AND. FLAG1 .LE. 0) WRITE(3,90)
   90 FORMAT(/T57,'ANNULAR  DIFFUSER - NOZZLE')
      IF (FLAG4 .GT. 0) WRITE(3,100)
  100 FORMAT(/T57,'GRAPHICAL  OUTPUT')
      IF (FLAG4 .LE. 0) WRITE(3,110)
  110 FORMAT(/T57,'NO  GRAPHICAL  OUTPUT  SPECIFIED')
C
C          OUTPUT  ERRORS  DETECTED
C
      IF (FLAG1 .LE. 0) GO TO 130
      IF (FLAG3 .LE. 0) GO TO 130
      WRITE(3,120)
  120 FORMAT(1H1,T1,'0IF FLAG1 GREATER THAN ZERO FLAG3 MUST BE ZERO')
      STOP
  130 IF (FLAG1 .LE. 0) GO TO 150
      IF (MXISYS .EQ. 1) GO TO 150
      WRITE(3,140)
  140 FORMAT(1H1,T1,'0IF FLAG1 GREATER THAN ZERO THEN',T32,
     1' MXISYS MUST BE ONE')
      STOP
  150 NM1 = N - 1
      NP1 = N + 1
      MM1 = M - 1
      MP1 = M + 1
      NM2 = N - 2
      NP2 = N + 2
      MM2 = M - 2
      MP2 = M + 2
      MSLUP1 = MSLOTU + 1
      MSLLP1 = MSLOTL + 1
      NSLLM1 = MSLOTL - 1
      MSTUP1 = MSTAGU + 1
      MSTUP2 = MSTAGU + 2
      MSTUP3 = MSTAGU + 3
      MSTLP1 = MSTAGL + 1
      MSTLP2 = MSTAGL + 2
      MSTLP3 = MSTAGL + 3
      NSTLP1 = NSTAGL + 1
      NSTLP2 = NSTAGL + 2
```

```
      NSTUP1 = NSTAGU + 1                                               PAR1 125
      NSTUP2 = NSTAGU + 2                                               PAR1 126
      DO 155 J=1,NP2                                                    PAR1 127
      DO 155 I=1,MP2                                                    PAR1 128
      Q(I,J) = 1.0                                                      PAR1 129
  155 R(I,J) = 1.0                                                      PAR1 130
      DO 157 I=1,MP2                                                    PAR1 131
      RSLOTU(I) = 1.0                                                   PAR1 132
      RSLOTL(I) = 1.0                                                   PAR1 133
      QSLOTU(I) = 1.0                                                   PAR1 134
  157 QSLOTL(I) = 1.0                                                   PAR1 135
C                                                                       PAR1 136
C         READ  WALL  VELOCITY  DISTRIBUTION                            PAR1 137
C                                                                       PAR1 138
      READ(1,160)(PHI(I),I=2,MP1)                                       PAR1 139
      READ(1,160)(Q(I,NP1),I=2,MSLUP1)                                  PAR1 140
      READ(1,160)(Q(I,NSTUP1),I=MSTUP2,MP1)                             PAR1 141
      READ(1,160)(QSLOTU(I),I=MSTUP2,MSLUP1)                            PAR1 142
  160 FORMAT(6F10.0)                                                    PAR1 143
      IF (FLAG1 .GT. 0  .OR. FLAG3 .GT. 0) GO TO 170                    PAR1 144
      READ(1,160)(Q(I,2),I=2,MSLLP1)                                    PAR1 145
      READ(1,160)(Q(I,NSTLP1),I=MSTLP2,MP1)                             PAR1 146
      READ(1,160)(QSLOTL(I),I=MSTLP2,MSLLP1)                            PAR1 147
  170 DO 180 I=2,M                                                      PAR1 148
  180 DPHI(I) = PHI(I+1) - PHI(I)                                       PAR1 149
      DPHI(1) = DPHI(2)                                                 PAR1 150
      DPHI(MP1) = DPHI(M)                                               PAR1 151
C                                                                       PAR1 152
C         READ  STREAM  FUNCTION  SPACING                               PAR1 153
C                                                                       PAR1 154
      READ(1,160)(PSI(I),I=2,NP1)                                       PAR1 155
      DO 190 I=2,N                                                      PAR1 156
  190 DPSI(I) = PSI(I+1) - PSI(I)                                       PAR1 157
      DPSI(1) = DPSI(2)                                                 PAR1 158
      DPSI(NP1) = DPSI(N)                                               PAR1 159
C                                                                       PAR1 160
C         COMPUTE  ARC  LENGTHS  ALONG  THE  WALL                       PAR1 161
C                                                                       PAR1 162
      DO 200 I=2,M                                                      PAR1 163
  200 DX(I) = DPHI(I)                                                   PAR1 164
      IMINSU = 2                                                        PAR1 165
      IMAXSU = MM1                                                      PAR1 166
      CALL SETUP                                                        PAR1 167
      DO 210 I=2,MSLUP1                                                 PAR1 168
  210 GRAND(I) = 1.0 / Q(I,NP1)                                         PAR1 169
      IMINSI = 2                                                        PAR1 170
      IMAXSI = MSLUP1                                                   PAR1 171
      CALL SIMP                                                         PAR1 172
      DO 220 I=2,MSLUP1                                                 PAR1 173
  220 S(I) = GIFT(I)                                                    PAR1 174
      S1(MSTUP1) = 0.0                                                  PAR1 175
      S2(MSTUP1) = 0.0                                                  PAR1 176
      Q(MSTUP1,NSTUP1) = 0.0                                            PAR1 177
      QSLOTU(MSTUP1) = 0.0                                              PAR1 178
      S1(MSTUP2) = 2. * DPHI(MSTUP1) / Q(MSTUP2,NSTUP1)                 PAR1 179
      S2(MSTUP2) = S1(MSTUP2)                                           PAR1 180
      DO 230 I=MSTUP2,MP1                                               PAR1 181
  230 GRAND(I) = 1.0 / Q(I,NSTUP1)                                      PAR1 182
      IMINSI = MSTUP2                                                   PAR1 183
      IMAXSI = MP1                                                      PAR1 184
```

```
      CALL SIMP
      DO 240 I=MSTUP3,MP1
  240 S1(I) = GIFT(I) + S1(MSTUP2)
      DO 250 I=MSTUP2,MSLUP1
  250 GRAND(I) = 1.0 / QSLOTU(I)
      IMINSI = MSTUP2
      IMAXSI = MSLUP1
      CALL SIMP
      DO 260 I=MSTUP3,MSLUP1
  260 S2(I) = GIFT(I) + S2(MSTUP2)
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 330
      DO 270 I=2,MSLLP1
  270 GRAND(I) = 1.0 / Q(I,2)
      IMINSI = 2
      IMAXSI = MSLLP1
      CALL SIMP
      DO 280 I=2,MSLLP1
  280 SBAR(I) = GIFT(I)
      SBAR1(MSTLP1) = 0.0
      SBAR2(MSTLP1) = 0.0
      Q(MSTLP1,NSTLP1) = 0.0
      QSLOTL(MSTLP1) = 0.0
      SBAR1(MSTLP2) = 2. * DPHI(MSTLP1) / Q(MSTLP2,NSTLP1)
      SBAR2(MSTLP2) = SBAR1(MSTLP2)
      DO 290 I=MSTLP2,MP1
  290 GRAND(I) = 1.0 / Q(I,NSTLP1)
      IMINSI = MSTLP2
      IMAXSI = MP1
      CALL SIMP
      DO 300 I=MSTLP3,MP1
  300 SBAR1(I) = GIFT(I) + SBAR1(MSTLP2)
      DO 310 I=MSTLP2,MSLLP1
  310 GRAND(I) = 1.0 / QSLOTL(I)
      IMINSI = MSTLP2
      IMAXSI = MSLLP1
      CALL SIMP
      DO 320 I=MSTLP3,MSLLP1
  320 SBAR2(I) = GIFT(I) + SBAR2(MSTLP2)
C
C           OUTPUT WALL VELOCITY DISTRIBUTION AND VELOCITY
C           POTENTIAL DISTRIBUTION ON UPPER SURFACE
C
  330 WRITE(3,20)
      WRITE(3,340)HEDR,DATE1,DATE2,DATE3,CASE
  340 FORMAT(//T2,15A4,T64,I2,T66,'/',T67,I2,T69,'/',T70,I2,
     1T76,'CASE  NO.  ',T87,I4///T2,'WALL VELOCITY',
     2T15,' DISTRIBUTION AND POTENTIAL SPACING',
     3T50,' UPSTREAM OF SLOT ( UPPER SURFACE )'///T16,'S',
     4T36,'Q',T54,'PHI',T74,'DPHI'//)
      ICOUNT = 1
      DO 360 I=2,MSLUP1
      ICOUNT = ICOUNT + 1
      WRITE(3,350)I,S(I),Q(I,NP1),PHI(I),DPHI(I)
  350 FORMAT(T2,I3,T10,F10.5,T30,F10.5,T50,F10.5/T70,F10.5)
      IF (ICOUNT .LT. 23) GO TO 360
      ICOUNT = 1
      WRITE(3,20)
      WRITE(3,340)HEDR,DATE1,DATE2,DATE3,CASE
  360 CONTINUE
      WRITE(3,20)
```

```
      WRITE(3,370)HEDR,DATE1,DATE2,DATE3,CASE                           PAR1 245
  370 FORMAT(//T2,15A4,T64,I2,T66,'/',T67,I2,T69,'/',T70,I2,            PAR1 246
     1T76,'CASE  NO.  ',T87,I4///T2,'WALL VELOCITY',                    PAR1 247
     2T15,' DISTRIBUTION AND POTENTIAL SPACING',                        PAR1 248
     3T50,' DOWNSTREAM OF SLOT ( UPPER SURFACE )'///T16,'S',            PAR1 249
     4T36,'Q',T54,'PHI',T74,'DPHI'//)                                   PAR1 250
      ICOUNT = 1                                                        PAR1 251
      DO 380 I=MSTUP1,MP1                                               PAR1 252
      ICOUNT = ICOUNT + 1                                               PAR1 253
      WRITE(3,350)I,S1(I),Q(I,NSTUP1),PHI(I),DPHI(I)                    PAR1 254
      IF (ICOUNT .LT. 23) GO TO 380                                     PAR1 255
      ICOUNT = 1                                                        PAR1 256
      WRITE(3,20)                                                       PAR1 257
      WRITE(3,370)HEDR,DATE1,DATE2,DATE3,CASE                           PAR1 258
  380 CONTINUE                                                          PAR1 259
      WRITE(3,20)                                                       PAR1 260
      WRITE(3,390)HEDR,DATE1,DATE2,DATE3,CASE                           PAR1 261
  390 FORMAT(//T2,15A4,T64,I2,T66,'/',T67,I2,T69,'/',T70,I2,            PAR1 262
     1T76,'CASE  NO.  ',T87,I4///T2,'WALL VELOCITY',                    PAR1 263
     2T15,' DISTRIBUTION AND POTENTIAL SPACING',                        PAR1 264
     3T50,' INSIDE SLOT ( UPPER SURFACE )'///T16,'S',                   PAR1 265
     4T36,'Q',T54,'PHI',T74,'DPHI'//)                                   PAR1 266
      ICOUNT = 1                                                        PAR1 267
      DO 400 I=MSTUP1,MSLUP1                                            PAR1 268
      ICOUNT = ICOUNT + 1                                               PAR1 269
      WRITE(3,350)I,S2(I),QSLOTU(I),PHI(I),DPHI(I)                      PAR1 270
      IF (ICOUNT .LT. 23) GO TO 400                                     PAR1 271
      ICOUNT = 1                                                        PAR1 272
      WRITE(3,20)                                                       PAR1 273
      WRITE(3,390)HEDR,DATE1,DATE2,DATE3,CASE                           PAR1 274
  400 CONTINUE                                                          PAR1 275
      Q(MSTUP1,NSTUP1) = 1.0                                            PAR1 276
      QSLOTU(MSTUP1) = 1.0                                              PAR1 277
C                                                                       PAR1 278
C         OUTPUT WALL VELOCITY DISTRIBUTION AND VELOCITY                PAR1 279
C         POTENTIAL DISTRIBUTION ON LOWER SURFACE                       PAR1 280
C                                                                       PAR1 281
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 470                     PAR1 282
      WRITE(3,20)                                                       PAR1 283
      WRITE(3,410)HEDR,DATE1,DATE2,DATE3,CASE                           PAR1 284
  410 FORMAT(//T2,15A4,T64,I2,T66,'/',T67,I2,T69,'/',T70,I2,            PAR1 285
     1T76,'CASE  NO.  ',T87,I4///T2,'WALL VELOCITY',                    PAR1 286
     2T15,' DISTRIBUTION AND POTENTIAL SPACING',                        PAR1 287
     3T50,' UPSTREAM OF SLOT ( LOWER SURFACE )'///T16,'S',              PAR1 288
     4T36,'Q',T54,'PHI',T74,'DPHI'//)                                   PAR1 289
      ICOUNT = 1                                                        PAR1 290
      DO 420 I=2,MSLLP1                                                 PAR1 291
      ICOUNT = ICOUNT + 1                                               PAR1 292
      WRITE(3,350)I,SBAR(I),Q(I,2),PHI(I),DPHI(I)                       PAR1 293
      IF (ICOUNT .LT. 23) GO TO 420                                     PAR1 294
      ICOUNT = 1                                                        PAR1 295
      WRITE(3,20)                                                       PAR1 296
      WRITE(3,410)HEDR,DATE1,DATE2,DATE3,CASE                           PAR1 297
  420 CONTINUE                                                          PAR1 298
      WRITE(3,20)                                                       PAR1 299
      WRITE(3,430)HEDR,DATE1,DATE2,DATE3,CASE                           PAR1 300
  430 FORMAT(//T2,15A4,T64,I2,T66,'/',T67,I2,T69,'/',T70,I2,            PAR1 301
     1T76,'CASE  NO.  ',T87,I4///T2,'WALL VELOCITY',                    PAR1 302
     2T15,' DISTRIBUTION AND POTENTIAL SPACING',                        PAR1 303
     3T50,' DOWNSTREAM OF SLOT ( LOWER SURFACE )'///T16,'S',            PAR1 304
```

```
     4T36,'Q',T54,'PHI',T74,'DPHI'//)
      ICOUNT = 1
      DO 440 I=MSTLP1,MP1
      ICOUNT = ICOUNT + 1
      WRITE(3,350)I,SBAR1(I),Q(I,NSTLP1),PHI(I),DPHI(I)
      IF (ICOUNT .LT. 23) GO TO 440
      ICOUNT = 1
      WRITE(3,20)
      WRITE(3,430)HEDR,DATE1,DATE2,DATE3,CASE
  440 CONTINUE
      WRITE(3,20)
      WRITE(3,450)HEDR,DATE1,DATE2,DATE3,CASE
  450 FORMAT(//T2,15A4,T64,I2,T66,'/',T67,I2,T69,'/',T70,I2,
     1T76,'CASE  NO.  ',T87,I4///T2,'WALL VELOCITY',
     2T15,' DISTRIBUTION AND POTENTIAL SPACING',
     3T50,' INSIDE OF SLOT ( LOWER SURFACE )'///T16,'S',
     4T36,'Q',T54,'PHI',T74,'DPHI'//)
      ICOUNT = 1
      DO 460 I=MSTLP1,MSLLP1
      ICOUNT = ICOUNT + 1
      WRITE(3,350)I,SBAR2(I),QSLOTL(I),PHI(I),DPHI(I)
      IF (ICOUNT .LT. 23) GO TO 460
      ICOUNT = 1
      WRITE(3,20)
      WRITE(3,450)HEDR,DATE1,DATE2,DATE3,CASE
  460 CONTINUE
      Q(MSTLP1,NSTLP1) = 1.0
      QSLOTL(MSTLP1) = 1.0
C
C           OUTPUT  STREAM  FUNCTION  DISTRIBUTION  AND
C           SPACING
C
  470 WRITE(3,20)
      WRITE(3,480)HEDR,DATE1,DATE2,DATE3,CASE
  480 FORMAT(//T2,15A4,T64,I2,T66,'/',T67,I2,T69,'/',T70,I2,
     1T76,'CASE  NO.  ',T87,I4///T2,'STREAM  FUNCTION',
     2T18,'  DISTRIBUTION',///T15,'PSI',T35,'DPSI'//)
      ICOUNT = 1
      DO 500 J=2,NP1
      ICOUNT = ICOUNT + 1
      WRITE(3,490)J,PSI(J),DPSI(J)
  490 FORMAT(T2,I3,T10,F10.5,/T30,F10.5)
      IF (ICOUNT .LT. 23) GO TO 500
      ICOUNT = 1
      WRITE(3,20)
      WRITE(3,480)HEDR,DATE1,DATE2,DATE3,CASE
  500 CONTINUE
C
C           COMPUTE INTEGRATION CONSTANTS INVOLVING DPHI
C
      DO 510 I=2,M
  510 DX(I) = DPHI(I)
      IMINSU = 2
      IMAXSU = MM1
      CALL SETUP
      DO 520 I=2,MM1
      EPHI(I) = ESIMP(I)
      FPHI(I) = FSIMP(I)
  520 GPHI(I) = GSIMP(I)
C
```

```
C         COMPUTE INTEGRATION CONSTANTS INVOLVING DPSI
C
      DO 530 J=2,N
  530 DX(J) = DPSI(J)
      IMINSU = 2
      IMAXSU = NM1
      CALL SETUP
      DO 540 J=2,NM1
      EPSI(J) = ESIMP(J)
      FPSI(J) = FSIMP(J)
  540 GPSI(J) = GSIMP(J)
C
C         COMPUTE CONSTANTS TO CONTROL INTEGRATION
C
      IMIN = 3
      IMAX = M
      JMIN = 2
      JMAX = NP1
      IF (FLAG3 .GT. 0) JMIN=3
      IF (FLAG2 .LE. 0) GO TO 550
      IMIN = 2
      IMAX = MP1
C
C         INITIALIZE  ARRAYS
C
  550 DO 560 J=2,NP1
      Q(2,J) = Q(2,NP1)
  560 Q(MP1,J) = Q(MP1,NSTUP1)
      DO 570 J=1,NP2
      DO 570 I=1,MP2
      P = Q(I,J)
  570 Q(I,J) = DLOG(P)
      DO 580 I=1,MP2
      P = QSLOTU(I)
      QSLOTU(I) = DLOG(P)
      P = QSLOTL(I)
  580 QSLOTL(I) = DLOG(P)
C
C         COMPUTE VELOCITIES AROUND STAGNATION POINTS
C
      Q(MSTAGU,NSTUP1) = QSLOTU(MSTUP2)
      Q(MSTUP1,NSTUP2) = QSLOTU(MSTUP2)
      Q(MSTUP1,NSTAGU) = QSLOTU(MSTUP2)
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 610
      Q(MSTAGL,NSTLP1) = QSLOTL(MSTLP2)
      Q(MSTLP1,NSTLP2) = QSLOTL(MSTLP2)
      Q(MSTLP1,NSTAGL) = QSLOTL(MSTLP2)
  610 CONTINUE
      DO 620 J=1,NP2
      DO 620 I=1,MP2
  620 X(I,J) = 0.0
C
C         READ INLET AND OUTLET VELOCITY PROFILES (IF USED)
C
      IF (FLAG2 .NE. 2 .AND. FLAG2 .NE. 4) GO TO 640
      IMIN = 3
      READ(1,160)(Q(2,J),J=2,NP1)
      DO 630 J=2,NP1
      P = Q(2,J)
  630 Q(2,J) = DLOG(P)
```

```
  640 IF (FLAG2 .NE. 3 .AND. FLAG3 .NE. 4) GO TO 660                    PAR1 425
      IMAX = M                                                          PAR1 426
      NSTUP1 = NSTAGU + 1                                               PAR1 427
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0)READ(1,160)(Q(MP1,J),J=2,      PAR1 428
     1NSTUP1)                                                           PAR1 429
      NSTLP1 = NSTAGL + 1                                               PAR1 430
      IF (FLAG1 .LE. 0 .AND. FLAG3 .LE. 0)READ(1,160)(Q(MP1,J),         PAR1 431
     1J=NSTLP1,NSTUP1)                                                  PAR1 432
      DO 650 J=2,NP1                                                    PAR1 433
      P = Q(MP1,J)                                                      PAR1 434
  650 Q(MP1,J) = DLOG(P)                                                PAR1 435
  660 CONTINUE                                                          PAR1 436
C                                                                       PAR1 437
C         WRITE HEADING FOR CONVERGENCE VARIABLES                       PAR1 438
C                                                                       PAR1 439
      IF (FLAG1 .LE. 0) WRITE(3,670)HEDR,DATE1,DATE2,DATE3,CASE         PAR1 440
  670 FORMAT(1H1//T57,'CLEMSON  UNIVERSITY',//T49,'MECHANICAL  ENGINEERIPAR1 441
     1NG  DEPARTMENT',///T2,15A4,T64,I2,T66,'/',T67,I2,T69,'/',T70,I2,  PAR1 442
     2T76,'CASE  NO.  ',T87,I4///)                                      PAR1 443
      IF (FLAG1 .LE. 0) WRITE(3,680)                                    PAR1 444
  680 FORMAT(T2,'IMPORTANT VARIABLES FOR EACH ITERATION OF',            PAR1 445
     1T43,' THE R COORDINATES'//)                                       PAR1 446
      WRITE(4)((Q(I,J),I=1,MP2),J=1,NP2)                                PAR1 447
      WRITE(5)((R(I,J),I=1,MP2),J=1,NP2)                                PAR1 448
      WRITE(12)((X(I,J),I=1,MP2),J=1,NP2)                               PAR1 449
      RETURN                                                            PAR1 450
      END                                                               PAR1 451
C         **********      SUBROUTINE  COEF      **********              COEF   1
C         THIS SUBROUTINE COMPUTES COEFFICIENTS TO BE USED              COEF   2
C         IN RELAXATION OF THE DIFFERENTIAL EQUATION                    COEF   3
C                                                                       COEF   4
      SUBROUTINE COEF                                                   COEF   5
      DIMENSION DPHI(150),DPSI(150),S(150),SBAR(150),                   COEF   6
     1          S1(150),S2(150),SBAR1(150),SBAR2(150),                  COEF   7
     2          PHI(150),PSI(150),RSLOTU(150),RSLOTL(150),              COEF   8
     3          QSLOTU(150),QSLOTL(150),ASLOTU(150),                    COEF   9
     4          ASLOTL(150),XSLOTU(150),XSLOTL(150),HEDR(15)            COEF  10
      DIMENSION A(150,50),B(150,50),C(150,50),D(150,50),                COEF  11
     1          E(150,50),R(150,50)                                     COEF  12
      INTEGER FLAG1,FLAG2,FLAG3,FLAG4,CASE,DATE1,DATE2,DATE3            COEF  13
      COMMON / C2 / FLAG1,FLAG2,FLAG3,FLAG4,HEDR,DATE1,DATE2,           COEF  14
     1              DATE3,CASE,MXIDE,ICKDE,IPRDE,IMIN,IMAX,             COEF  15
     2              JMIN,JMAX,NUT,M,N,TOLDE,DEL,EDS,JTER,               COEF  16
     3              ITER,DPHI,DPSI,S,SBAR,RMULT,S1,S2,SBAR1,            COEF  17
     4              SBAR2,MSTAGU,NSTAGU,MSLOTU,MSTAGL,NSTAGL,           COEF  18
     5              MSLOTL,RADIN,W,PHI,PSI,RSLOTU,RSLOTL,               COEF  19
     6              QSLOTU,QSLOTL,ASLOTU,ASLOTL,XSLOTU,                 COEF  20
     7              XSLOTL,G,MIDJ,TOLSYS                                COEF  21
      REWIND 5                                                          COEF  22
      REWIND 6                                                          COEF  23
      REWIND 7                                                          COEF  24
      REWIND 8                                                          COEF  25
      REWIND 9                                                          COEF  26
      REWIND 10                                                         COEF  27
      MP2 = M + 2                                                       COEF  28
      NP2 = N + 2                                                       COEF  29
      NSTUM1 = NSTAGU - 1                                               COEF  30
      MSTUP2 = MSTAGU + 2                                               COEF  31
      MSTLP2 = MSTAGL + 2                                               COEF  32
      MSLUP1 = MSLOTU + 1                                               COEF  33
```

```
      MSLLP1 = MSLOTL + 1
      NSTLP1 = NSTAGL + 1
      NSTLP2 = NSTAGL + 2
      NSTUP2 = NSTAGU + 2
      READ(5)((R(I,J),I=1,MP2),J=1,NP2)
C
C          COMPUTE COEFFICIENTS THROUGHOUT DIFFUSER
C
      DO 100  J=JMIN,JMAX
      DD = DPSI(J-1) * DPSI(J)
      EE = DPSI(J-1) + DPSI(J)
      FF = DPSI(J-1) - DPSI(J)
      DO 100 I=IMIN,IMAX
      AA = DPHI(I-1) * DPHI(I)
      BB = DPHI(I-1) + DPHI(I)
      CC = DPHI(I-1) -DPHI(I)
      DERX = DPHI(I-1) * R(I+1,J) / ( DPHI(I) * BB )
      DERX = DERX - DPHI(I) * R(I-1,J) / ( DPHI(I-1) * BB )
      DERX = DERX - CC * R(I,J) / AA
      DERY = DPSI(J-1) * R(I,J+1) / ( DPSI(J) * EE )
      DERY = DERY - DPSI(J) * R(I,J-1) / ( DPSI(J-1) * EE )
      DERY = DERY - FF * R(I,J) / DD
      DDERX = 2. * R(I+1,J) / ( DPHI(I) * BB )
      DDERX = DDERX + 2. * R(I-1,J) / ( DPHI(I-1) * BB )
      DDERX = DDERX - 2. * R(I,J) / AA
      F = 2. * R(I,J) * R(I,J) * R(I,J) * R(I,J) / DD
      F = F + R(I,J) * R(I,J) * R(I,J) * FF * DERY / DD
      F = F + 2. * R(I,J) * R(I,J) / AA
      F = F - R(I,J) * CC * DERX / AA
      A(I,J) = 2. * R(I,J) - DPHI(I-1) * DERX
      A(I,J) = A(I,J) * W * R(I,J) / ( F * DPHI(I) * BB )
      B(I,J) = 2. * R(I,J) + DPHI(I) * DERX
      B(I,J) = B(I,J) * W * R(I,J) / ( F * DPHI(I-1) * BB )
      R3 = R(I,J) * R(I,J) * R(I,J)
      C(I,J) = 2. * R(I,J) + DPSI(J-1) * DERY
      C(I,J) = C(I,J) * W * R3 / ( F * DPSI(J) * EE)
      D(I,J) = 2. * R(I,J) - DPSI(J) * DERY
      D(I,J) = D(I,J) * W * R3 / ( F * DPSI(J-1) * EE )
  100 E(I,J) = ( R(I,J) * DDERX - 2. * DERX * DERX ) / F
C
C          COMPUTE COEFFICIENTS INSIDE UPPER SLOT
C
      J = NSTAGU + 2
      DD = DPSI(J-1) * DPSI(J)
      EE = DPSI(J-1) + DPSI(J)
      FF = DPSI(J-1) - DPSI(J)
      DO 200 I=MSTUP2,MSLUP1
      AA = DPHI(I-1) * DPHI(I)
      BB = DPHI(I-1) + DPHI(I)
      CC = DPHI(I-1) - DPHI(I)
      DERX = DPHI(I-1) * R(I+1,J) / ( DPHI(I) * BB )
      DERX = DERX - DPHI(I) * R(I-1,J) / ( DPHI(I-1) * BB )
      DERX = DERX - CC * R(I,J) / AA
      DERY = DPSI(J-1) * R(I,J+1) / ( DPSI(J) * EE )
      DERY = DERY - DPSI(J) * RSLOTU(I) / ( DPSI(J-1) * EE )
      DERY = DERY - FF * R(I,J) / DD
      DDERX = 2. * R(I+1,J) / ( DPHI(I) * BB )
      DDERX = DDERX + 2. * R(I-1,J) / ( DPHI(I-1) * BB )
      DDERX = DDERX - 2. * R(I,J) / AA
      F = 2. * R(I,J) * R(I,J) * R(I,J) * R(I,J) / DD
```

```
      F = F + R(I,J) * R(I,J) * R(I,J) * FF * DERY / DD
      F = F + 2. * R(I,J) * R(I,J) / AA
      F = F - R(I,J) * CC * DERX / AA
      A(I,J) = 2. * R(I,J) - DPHI(I-1) * DERX
      A(I,J) = A(I,J) * W * R(I,J) / ( F * DPHI(I) * BB )
      B(I,J) = 2. * R(I,J) + DPHI(I) * DERX
      B(I,J) = B(I,J) * W * R(I,J) / ( F * DPHI(I-1) * BB )
      R3 = R(I,J) * R(I,J) * R(I,J)
      C(I,J) = 2. * R(I,J) + DPSI(J-1) * DERY
      C(I,J) = C(I,J) * W * R3 / ( F * DPSI(J) * EE )
      D(I,J) = 2. * R(I,J) - DPSI(J) * DERY
      D(I,J) = D(I,J) * W * R3 / ( F * DPSI(J-1) * EE )
  200 E(I,J) = ( R(I,J) * DDERX - 2. * DERX * DERX ) / F
C
C           COMPUTE COEFFICIENTS INSIDE LOWER SLOT
C
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 1000
      J = NSTAGL
      DD = DPSI(J-1) * DPSI(J)
      EE = DPSI(J-1) + DPSI(J)
      FF = DPSI(J-1) - DPSI(J)
      DO 300 I=MSTLP2,MSLLP1
      AA = DPHI(I-1) * DPHI(I)
      BB = DPHI(I-1) + DPHI(I)
      CC = DPHI(I-1) - DPHI(I)
      DERX = DPHI(I-1) * R(I+1,J) / ( DPHI(I) * BB )
      DERX = DERX - DPHI(I) * R(I-1,J) / ( DPHI(I-1) * BB )
      DERX = DERX - CC * R(I,J) / AA
      DERY = DPSI(J-1) * RSLOTL(I) / ( DPSI(J) * EE )
      DERY = DERY - DPSI(J) * R(I,J-1) / ( DPSI(J-1) * EE )
      DERY = DERY - FF * R(I,J) / DD
      DDERX = 2. * R(I+1,J) / ( DPHI(I) * BB )
      DDERX = DDERX + 2. * R(I-1,J) / ( DPHI(I-1) * BB )
      DDERX = DDERX - 2. * R(I,J) / AA
      F = 2. * R(I,J) * R(I,J) * R(I,J) * R(I,J) / DD
      F = F + R(I,J) * R(I,J) * R(I,J) * FF * DERY / DD
      F = F + 2. * R(I,J) * R(I,J) / AA
      F = F - R(I,J) * CC * DERX / AA
      A(I,J) = 2. * R(I,J) - DPHI(I-1) * DERX
      A(I,J) = A(I,J) * W * R(I,J) / ( F * DPHI(I) * BB )
      B(I,J) = 2. * R(I,J) + DPHI(I) * DERX
      B(I,J) = B(I,J) * W * R(I,J) / ( F * DPHI(I-1) * BB )
      R3 = R(I,J) * R(I,J) * R(I,J)
      C(I,J) = 2. * R(I,J) + DPSI(J-1) * DERY
      C(I,J) = C(I,J) * W * R3 / ( F * DPSI(J) * EE )
      D(I,J) = 2. * R(I,J) - DPSI(J) * DERY
      D(I,J) = D(I,J) * W * R3 / ( F * DPSI(J-1) * EE )
  300 E(I,J) = ( R(I,J) * DDERX - 2. * DERX * DERX ) / F
 1000 CONTINUE
      G = 1. - W
      WRITE(6)((A(I,J),I=1,MP2),J=1,NP2)
      WRITE(7)((B(I,J),I=1,MP2),J=1,NP2)
      WRITE(8)((C(I,J),I=1,MP2),J=1,NP2)
      WRITE(9)((D(I,J),I=1,MP2),J=1,NP2)
      WRITE(10)((E(I,J),I=1,MP2),J=1,NP2)
      RETURN
      END
```

```
C        **********      SUBROUTINE RELAX      **********
C
C        THIS SUBROUTINE RELAXES THE DIFFERENTIAL EQUATION
C        USING COEFFICIENTS COMPUTED IN COEF
C
      SUBROUTINE RELAX
      DIMENSION DPHI(150),DPSI(150),S(150),SBAR(150),
     1          S1(150),S2(150),SBAR1(150),SBAR2(150),
     2          PHI(150),PSI(150),RSLOTU(150),RSLOTL(150),
     3          QSLOTU(150),QSLOTL(150),ASLOTU(150),
     4          ASLOTL(150),XSLOTU(150),XSLOTL(150),HEDR(15)
      DIMENSION A(150,50),B(150,50),C(150,50),D(150,50),
     1          E(150,50),Q(150,50)
      DOUBLE PRECISION P
      INTEGER FLAG1,FLAG2,FLAG3,FLAG4,CASE,DATE1,DATE2,DATE3
      COMMON / C2 / FLAG1,FLAG2,FLAG3,FLAG4,HEDR,DATE1,
     1              DATE2,DATE3,CASE,MXIDE,ICKDE,IPRDE,IMIN,
     2              IMAX,JMIN,JMAX,NUT,M,N,TOLDE,DEL,EPS,
     3              JTER,ITER,DPHI,DPSI,S,SBAR,RMULT,S1,S2,
     4              SBAR1,SBAR2,MSTAGU,NSTAGU,MSLOTU,MSTAGL,
     5              NSTAGL,MSLOTL,RADIN,W,PHI,PSI,RSLOTU,
     6              RSLOTL,QSLOTU,QSLOTL,ASLOTU,ASLOTL,
     7              XSLOTU,XSLOTL,G,MIDJ,TOLSYS
      REWIND 4
      REWIND 6
      REWIND 7
      REWIND 8
      REWIND 9
      REWIND 10
      NP1 = N + 1
      MP1 = M + 1
      NP2 = N + 2
      MP2 = M + 2
      NM1 = N - 1
      MM1 = M - 1
      NM2 = N - 2
      MM2 = M - 2
      MSTUM1 = MSTAGU - 1
      MSTUP1 = MSTAGU + 1
      MSTUP2 = MSTAGU + 2
      MSTLM1 = MSTAGL - 1
      MSTLP2 = MSTAGL + 2
      NSTUM1 = NSTAGU - 1
      NSTUP1 = NSTAGU + 1
      NSTUP2 = NSTAGU + 2
      NSTUP3 = NSTAGU + 3
      NSTLM1 = NSTAGL - 1
      NSTLP1 = NSTAGL + 1
      NSTLP2 = NSTAGL + 2
      NSTLP3 = NSTAGL + 3
      MSLUP1 = MSLOTU + 1
      MSLUP2 = MSLOTU + 2
      MSLLP1 = MSLOTL + 1
      MSLLP2 = MSLOTL + 2
      READ(4)((Q(I,J),I=1,MP2),J=1,NP2)
      READ(6)((A(I,J),I=1,MP2),J=1,NP2)
      READ(7)((B(I,J),I=1,MP2),J=1,NP2)
      READ(8)((C(I,J),I=1,MP2),J=1,NP2)
      READ(9)((D(I,J),I=1,MP2),J=1,NP2)
      READ(10)((E(I,J),I=1,MP2),J=1,NP2)
```

```
      II = 1                                                            RELX  61
      JTER = 0                                                          RELX  62
      DO 3000 ITTT=1,MXIDE                                              RELX  63
      DEL = 0.0                                                         RELX  64
      JTER = JTER + 1                                                   RELX  65
C                                                                       RELX  66
C          RELAX LOWER PORTION FOR 2-D AND BELL                         RELX  67
C                                                                       RELX  68
      IF (FLAG1 .LE. 0 .AND. FLAG3 .LE. 0) GO TO 30                     RELX  69
      DO 10 J=3,NSTUM1                                                  RELX  70
      DO 10 I=IMIN,IMAX                                                 RELX  71
      AA = A(I,J) * Q(I+1,J)                                            RELX  72
      BB = B(I,J) * Q(I-1,J)                                            RELX  73
      CC = C(I,J) * Q(I,J+1)                                            RELX  74
      DD = D(I,J) * Q(I,J-1)                                            RELX  75
      EE = E(I,J) + G * Q(I,J)                                          RELX  76
      QLAST = Q(I,J)                                                    RELX  77
      Q(I,J) = AA + BB + CC + DD + EE                                   RELX  78
      DELSTR = ABS(QLAST - Q(I,J))                                      RELX  79
   10 IF (DELSTR .GT. DEL) DEL = DELSTR                                 RELX  80
      IF (FLAG3 .GT. 0) GO TO 110                                       RELX  81
      J = 2                                                             RELX  82
      DO 20 I=IMIN,IMAX                                                 RELX  83
      AA = A(I,J) * Q(I+1,J)                                            RELX  84
      BB = B(I,J) * Q(I-1,J)                                            RELX  85
      CC = C(I,J) * Q(I,J+1)                                            RELX  86
      DD = D(I,J) * Q(I,J-1)                                            RELX  87
      EE = E(I,J) + G * Q(I,J)                                          RELX  88
      QLAST = Q(I,J)                                                    RELX  89
      Q(I,J) = AA + BB + CC + DD + EE                                   RELX  90
      DELSTR = ABS(QLAST - Q(I,J))                                      RELX  91
   20 IF (DELSTR .GT. DEL) DEL = DELSTR                                 RELX  92
      GO TO 110                                                         RELX  93
C                                                                       RELX  94
C          RELAX LOWER PORTION OF ANNULAR DIFFUSER                      RELX  95
C                                                                       RELX  96
   30 DO 40 J=3,NSTLM1                                                  RELX  97
      DO 40 I=IMIN,MSLLP1                                               RELX  98
      AA = A(I,J) * Q(I+1,J)                                            RELX  99
      BB = B(I,J) * Q(I-1,J)                                            RELX 100
      CC = C(I,J) * Q(I,J+1)                                            RELX 101
      DD = D(I,J) * Q(I,J-1)                                            RELX 102
      EE = E(I,J) + G * Q(I,J)                                          RELX 103
      QLAST = Q(I,J)                                                    RELX 104
      Q(I,J) = AA + BB + CC + DD + EE                                   RELX 105
      DELSTR = ABS(QLAST - Q(I,J))                                      RELX 106
   40 IF (DELSTR .GT. DEL) DEL = DELSTR                                 RELX 107
      J = NSTAGL                                                        RELX 108
      DO 50 I=IMIN,MSTAGL                                               RELX 109
      AA = A(I,J) * Q(I+1,J)                                            RELX 110
      BB = B(I,J) * Q(I-1,J)                                            RELX 111
      CC = C(I,J) * Q(I,J+1)                                            RELX 112
      DD = D(I,J) * Q(I,J-1)                                            RELX 113
      EE = E(I,J) + G * Q(I,J)                                          RELX 114
      QLAST = Q(I,J)                                                    RELX 115
      Q(I,J) = AA + BB + CC + DD + EE                                   RELX 116
      DELSTR = ABS(QLAST - Q(I,J))                                      RELX 117
   50 IF (DELSTR .GT. DEL) DEL = DELSTR                                 RELX 118
      DO 60 I=MSTLP2,MSLLP1                                             RELX 119
      AA  = A(I,J) * Q(I+1,J)                                           RELX 120
```

```
      BB = B(I,J) * Q(I-1,J)
      CC = C(I,J) * QSLOTL(I)
      DD = D(I,J) * Q(I,J-1)
      EE = E(I,J) + G * Q(I,J)
      QLAST = Q(I,J)
      Q(I,J) = AA + BB + CC + DD + EE
      DELSTR = ABS(QLAST - Q(I,J))
   60 IF (DELSTR .GT. DEL) DEL = DELSTR
      J = NSTAGL + 1
      DO 70 I=IMIN,MSTLM1
      AA = A(I,J) * Q(I+1,J)
      BB = B(I,J) * Q(I-1,J)
      CC = C(I,J) * Q(I,J+1)
      DD = D(I,J) * Q(I,J-1)
      EE = E(I,J) + G * Q(I,J)
      QLAST = Q(I,J)
      Q(I,J) = AA + BB + CC + DD + EE
      DELSTR = ABS(QLAST - Q(I,J))
   70 IF (DELSTR .GT. DEL) DEL = DELSTR
      J = NSTAGL + 2
      DO 80 I=IMIN,MSTAGL
      AA = A(I,J) * Q(I+1,J)
      BB = B(I,J) * Q(I-1,J)
      CC = C(I,J) * Q(I,J+1)
      DD = D(I,J) * Q(I,J-1)
      EE = E(I,J) + G * Q(I,J)
      QLAST = Q(I,J)
      Q(I,J) = AA + BB + CC + DD + EE
      DELSTR = ABS(QLAST - Q(I,J))
   80 IF (DELSTR .GT. DEL) DEL = DELSTR
      DO 90 I=MSTLP2,IMAX
      AA = A(I,J) * Q(I+1,J)
      BB = B(I,J) * Q(I-1,J)
      CC = C(I,J) * Q(I,J+1)
      DD = D(I,J) * Q(I,J-1)
      EE = E(I,J) + G * Q(I,J)
      QLAST = Q(I,J)
      Q(I,J) = AA + BB + CC + DD + EE
      DELSTR = ABS(QLAST - Q(I,J))
   90 IF (DELSTR .GT. DEL) DEL = DELSTR
      DO 100 I=IMIN,IMAX
      DO 100 J=NSTLP3,NSTUM1
      AA = A(I,J) * Q(I+1,J)
      BB = B(I,J) * Q(I-1,J)
      CC = C(I,J) * Q(I,J+1)
      DD = D(I,J) * Q(I,J-1)
      EE = E(I,J) + G * Q(I,J)
      QLAST = Q(I,J)
      Q(I,J) = AA + BB + CC + DD + EE
      DELSTR = ABS(QLAST - Q(I,J))
  100 IF (DELSTR .GT. DEL) DEL = DELSTR
C
C           RELAX UPPER PORTION FOR ALL KINDS OF DIFFUSERS
C
  110 J = NSTAGU
      DO 120 I=IMIN,MSTAGU
      AA = A(I,J) * Q(I+1,J)
      BB = B(I,J) * Q(I-1,J)
      CC = C(I,J) * Q(I,J+1)
      DD = D(I,J) * Q(I,J-1)
```

```
      EE = E(I,J) + G * Q(I,J)
      QLAST = Q(I,J)
      Q(I,J) = AA + BB + CC + DD + EE
      DELSTR = ABS(QLAST - Q(I,J))
  120 IF (DELSTR .GT. DEL) DEL = DELSTR
      DO 130 I=MSTUP2,IMAX
      AA = A(I,J) * Q(I+1,J)
      BB = B(I,J) * Q(I-1,J)
      CC = C(I,J) * Q(I,J+1)
      DD = D(I,J) * Q(I,J-1)
      EE = E(I,J) + G * Q(I,J)
      QLAST = Q(I,J)
      Q(I,J) = AA + BB + CC + DD + EE
      DELSTR = ABS(QLAST - Q(I,J))
  130 IF (DELSTR .GT. DEL) DEL = DELSTR
      J = NSTAGU + 1
      DO 140 I=IMIN,MSTUM1
      AA = A(I,J) * Q(I+1,J)
      BB = B(I,J) * Q(I-1,J)
      CC = C(I,J) * Q(I,J+1)
      DD = D(I,J) * Q(I,J-1)
      EE = E(I,J) + G * Q(I,J)
      QLAST = Q(I,J)
      Q(I,J) = AA + BB + CC + DD + EE
      DELSTR = ABS(QLAST - Q(I,J))
  140 IF (DELSTR .GT. DEL) DEL = DELSTR
      J = NSTAGU + 2
      DO 150 I=IMIN,MSTAGU
      AA = A(I,J) * Q(I+1,J)
      BB = B(I,J) * Q(I-1,J)
      CC = C(I,J) * Q(I,J+1)
      DD = D(I,J) * Q(I,J-1)
      EE = E(I,J) + G * Q(I,J)
      QLAST = Q(I,J)
      Q(I,J) = AA + BB + CC + DD + EE
      DELSTR = ABS(QLAST - Q(I,J))
  150 IF (DELSTR .GT. DEL) DEL = DELSTR
      DO 160 I=MSTUP2,MSLUP1
      AA = A(I,J) * Q(I+1,J)
      BB = B(I,J) * Q(I-1,J)
      CC = C(I,J) * Q(I,J+1)
      DD = D(I,J) * QSLOTU(I)
      EE = E(I,J) + G * Q(I,J)
      QLAST = Q(I,J)
      Q(I,J) = AA + BB + CC + DD + EE
      DELSTR = ABS(QLAST - Q(I,J))
  160 IF (DELSTR .GT. DEL) DEL = DELSTR
      DO 170 J=NSTUP3,N
      DO 170 I=IMIN,MSLUP1
      AA = A(I,J) * Q(I+1,J)
      BB = B(I,J) * Q(I-1,J)
      CC = C(I,J) * Q(I,J+1)
      DD = D(I,J) * Q(I,J-1)
      EE = E(I,J) + G * Q(I,J)
      QLAST = Q(I,J)
      Q(I,J) = AA + BB + CC + DD + EE
      DELSTR = ABS(QLAST - Q(I,J))
  170 IF (DELSTR .GT. DEL) DEL = DELSTR
C
C          CORRECT BOUNDARY CONDITIONS
```

```
C                                                                       RELX 241
      DO 180 I=2,MSLUP1                                                 RELX 242
      Y3 = Q(I,NP1)                                                     RELX 243
      Y2 = Q(I,N)                                                       RELX 244
      Y1 = Q(I,NM1)                                                     RELX 245
      D1 = DPSI(NM1)                                                    RELX 246
      D2 = DPSI(N)                                                      RELX 247
      DER = ( Y3 - Y2 ) * ( D1 + D2 ) / ( D1 * D2 )                     RELX 248
      DER = DER - ( Y3 - Y1 ) * D2 / ( D1 * D1 + D1 * D2 )              RELX 249
  180 Q(I,NP2) = Y2 + 2. * DER * D2                                     RELX 250
      IF (FLAG1 .LE. 0 .AND. FLAG3 .LE. 0) GO TO 200                    RELX 251
      JOT = 2                                                           RELX 252
      DO 190 I=2,MP1                                                    RELX 253
  190 Q(I,1) = Q(I,3)                                                   RELX 254
  200 IF(FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 220                      RELX 255
      JOT = NSTLP1                                                      RELX 256
      DO 210 I=2,MSLLP1                                                 RELX 257
      Y3 = Q(I,4)                                                       RELX 258
      Y2 = Q(I,3)                                                       RELX 259
      Y1 = Q(I,2)                                                       RELX 260
      D1 = DPSI(2)                                                      RELX 261
      D2 = DPSI(3)                                                      RELX 262
      DER = ( Y2 - Y1 ) * ( D1 + D2 ) / ( D1 * D2 )                     RELX 263
      DER = DER - ( Y3 - Y1 ) * D1 / ( D1 * D2 + D2 * D2 )              RELX 264
  210 Q(I,1) = Y2 - 2. * DER * D1                                       RELX 265
  220 DO 230 J=2,NP1                                                    RELX 266
      Y3 = Q(4,J)                                                       RELX 267
      Y2 = Q(3,J)                                                       RELX 268
      Y1 = Q(2,J)                                                       RELX 269
      D1 = DPHI(2)                                                      RELX 270
      D2 = DPHI(3)                                                      RELX 271
      DER = ( Y2 - Y1 ) * ( D1 + D2 ) / ( D1 * D2 )                     RELX 272
      DER = DER - ( Y3 - Y1 ) * D1 / ( D1 * D2 + D2 * D2 )              RELX 273
  230 Q(1,J) = Y2 - 2. * DER * D1                                       RELX 274
      DO 240 J=JOT,NSTAGU                                               RELX 275
      Y3 = Q(MP1,J)                                                     RELX 276
      Y2 = Q(M,J)                                                       RELX 277
      Y1 = Q(MM1,J)                                                     RELX 278
      D1 = DPHI(MM1)                                                    RELX 279
      D2 = DPHI(M)                                                      RELX 280
      DER = ( Y3 - Y2 ) * ( D1 + D2 ) / ( D1 * D2 )                     RELX 281
      DER = DER - ( Y3 - Y1 ) * D2 / ( D1 * D1 + D1 * D2 )              RELX 282
  240 Q(MP2,J) = Y2 + 2. * DER * D2                                     RELX 283
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 260                     RELX 284
      DO 250 J=2,NSTAGL                                                 RELX 285
  250 Q(MSLLP2,J) = Q(MSLOTL,J)                                         RELX 286
      QSLOTL(MSLLP2) = QSLOTL(MSLOTL)                                   RELX 287
  260 DO 270 J=NSTUP2,NP1                                               RELX 288
  270 Q(MSLUP2,J) = Q(MSLOTU,J)                                         RELX 289
      QSLOTU(MSLUP2) = QSLOTU(MSLOTU)                                   RELX 290
      IF (FLAG3 .LE. 0) GO TO 285                                       RELX 291
      DO 280 I=2,MP1                                                    RELX 292
  280 Q(I,2) = Q(I,3)                                                   RELX 293
  285 IF (FLAG2 .LE. 0) GO TO 1000                                      RELX 294
      IF (FLAG1 .LE. 0 .AND. FLAG3 .LE. 0) GO TO 310                    RELX 295
      IF (FLAG2 .EQ. 2 .OR. FLAG2 .EQ. 4) GO TO 295                     RELX 296
      DO 290 J=2,NP1                                                    RELX 297
  290 Q(1,J) = Q(3,J)                                                   RELX 298
  295 IF (FLAG2 .EQ. 3 .OR. FLAG2 .EQ. 4) GO TO 1000                    RELX 299
      DO 300 J=2,NSTAGU                                                 RELX 300
```

```
  300 Q(MP2,J) = Q(M,J)
      GO TO 1000
  310 IF (FLAG2 .EQ. 2 .OR. FLAG2 .EQ. 4) GO TO 325
      DO 320 J=2,NP1
  320 Q(1,J) = Q(3,J)
  325 IF (FLAG2 .EQ. 3 .OR. FLAG2 .EQ. 4) GO TO 1000
      DO 330 J=NSTLP2,NSTAGU
  330 Q(MP2,J) = Q(M,J)
 1000 CONTINUE
C
C          CHECK FOR CONVERGENCE
C
      IF (DEL .LE. TOLDE) GO TO 4000
      IF (JTER .LT. II*IPRDE) GO TO 3000
      II = II +1
C
C          OUTPUT Q ARRAY
C
      DO 1010 I=2,MP1
      DO 1010 J=2,NP1
      P = Q(I,J)
 1010 Q(I,J) = DEXP(P)
      WRITE(3,1020)HEDR,DATE1,DATE2,DATE3,CASE
 1020 FORMAT(1H1//T57,'CLEMSON  UNIVERSITY',//T49,
     1'MECHANICAL  ENGINEERING  DEPARTMENT'///T2,15A4,T64,
     2I2,T66,'/',T67,I2,T69,'/',T70,I2,T76,'CASE  NO.  ',
     3T87,I4)
C
C          OUTPUT ARRAY
C
      WRITE(3,1030)ITER,JTER,EPS,DEL
 1030 FORMAT(///T1,'OVELOCITY THROUGHOUT DIFFUSER - NOZZLE',
     1T46,'ITER = ',T53,I4,T61,'JTER = ',T68,I4,T76,'EPS = '
     2,T82,E13.6,T99,'DEL = ',T105,E13.6,///T7,'PHI',T66,
     3'PSI')
      WRITE(3,1040)(PSI(IP),IP=2,NP1)
 1040 FORMAT(//(13X,5(7X,E13.6)))
      DO 1050 I=2,MP1
 1050 WRITE(3,1060)PHI(I),(Q(I,J),J=2,NP1)
 1060 FORMAT(//1X,E13.6,5(7X,E13.6),/(14X,5(7X,E13.6)))
C
C          CHANGE Q BACK TO LOGARITHMIC FORM
C
      DO 1070 I=2,MP1
      DO 1070 J=2,NP1
      P = Q(I,J)
 1070 Q(I,J) = DLOG(P)
      DO 1075 I=2,MP1
      P = QSLOTL(I)
      QSLOTL(I) = DEXP(P)
      P = QSLOTU(I)
 1075 QSLOTU(I) = DEXP(P)
C
C          OUTPUT Q VALUES ALONG THE SLOT INNER WALLS
C
      WRITE(3,1020)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,1080)ITER,JTER,EPS,DEL
 1080 FORMAT(///T1,'OVELOCITY INSIDE OF SLOTS',
     1T46,'ITER = ',T53,I4,T61,'JTER = ',T68,I4,T76,'EPS = '
     2,T82,E13.6,T99,'DEL = ',T105,E13.6,///T7,'PHI',T66,
```

```
     3'PSI')                                                            RELX 361
      WRITE(3,1040)(PHI(IP),IP=2,MP1)                                   RELX 362
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 1100                    RELX 363
      WRITE(3,1090)                                                     RELX 364
 1090 FORMAT(///T2,'LOWER SLOT')                                        RELX 365
      WRITE(3,1060)PSI(NSTLP1),(QSLOTL(I),I=2,MP1)                      RELX 366
 1100 WRITE(3,1110)                                                     RELX 367
 1110 FORMAT(///T2,'UPPER SLOT')                                        RELX 368
      WRITE(3,1060)PSI(NSTUP1),(QSLOTU(I),I=2,MP1)                      RELX 369
C                                                                       RELX 370
C           CHANGE SLOT VALUES BACK TO LOGARITHMIC FORM                 RELX 371
C                                                                       RELX 372
      DO 1120 I=2,MP1                                                   RELX 373
      P = QSLOTL(I)                                                     RELX 374
      QSLOTL(I) = DLOG(P)                                               RELX 375
      P = QSLOTU(I)                                                     RELX 376
 1120 QSLOTU(I) = DLOG(P)                                               RELX 377
 3000 CONTINUE                                                          RELX 378
 4000 REWIND 4                                                          RELX 379
      WRITE(4)((Q(I,J),I=1,MP2),J=1,NP2)                                RELX 380
      RETURN                                                            RELX 381
      END                                                               RELX 382
C          **********      SUBROUTINE  PART2       **********           PAR2   1
C                                                                       PAR2   2
C           THIS SUBROUTINE COMPUTES ALPHA , R AND X ARRAYS             PAR2   3
C           AND COMPUTES THE CHANGE OF THE R ARRAY FROM THE             PAR2   4
C           PREVIOUS ITERATION FOR AXISYMMETRIC COMPUTATIONS            PAR2   5
C                                                                       PAR2   6
      SUBROUTINE PART2                                                  PAR2   7
      DIMENSION DX(150),ESIMP(150),FSIMP(150),GSIMP(150),               PAR2   8
     1          GRAND(150),GIFT(150),PHI(150),PSI(150),                 PAR2   9
     2          DPSI(150),DPHI(150),S(150),SBAR(150),                   PAR2  10
     3          S1(150),SBAR1(150),S2(150),SBAR2(150),                  PAR2  11
     4          HEDR(15),EPHI(150),FPHI(150),GPHI(150),                 PAR2  12
     5          EPSI(150),ASLOTL(150),XSLOTU(150),XSLOTL(150)           PAR2  13
     6          ,FPSI(150),GPSI(150),RSLOTU(150),RSLOTL(150),           PAR2  14
     7          QSLOTU(150),QSLOTL(150),ASLOTU(150)                     PAR2  15
      DIMENSION Q(150,50),R(150,50),X(150,50),ALPHA(150,50),            PAR2  16
     1          R1(150,50)                                              PAR2  17
      DOUBLE PRECISION P                                                PAR2  18
      INTEGER FLAG1,FLAG2,FLAG3,FLAG4,CASE,DATE1,DATE2,DATE3            PAR2  19
      COMMON / C1 / IMINSU,IMAXSU,IMINSI,IMAXSI,GRAND,GIFT,             PAR2  20
     1              DX,ESIMP,FSIMP,GSIMP                                PAR2  21
      COMMON / C2 / FLAG1,FLAG2,FLAG3,FLAG4,HEDR,DATE1,DATE2            PAR2  22
     1              ,DATE3,CASE,MXIDE,ICKDE,IPRDE,IMIN,IMAX,            PAR2  23
     2              JMIN,JMAX,NUT,M,N,TOLDE,DEL,EPS,JTER,               PAR2  24
     3              ITER,DPHI,DPSI,S,SBAR,RMULT,S1,S2,SBAR1,            PAR2  25
     4              SBAR2,MSTAGU,NSTAGU,MSLOTU,MSTAGL,NSTAGL            PAR2  26
     5              ,MSLOTL,RADIN,W,PHI,PSI,RSLOTU,RSLOTL,              PAR2  27
     6              QSLOTU,QSLOTL,ASLOTU,ASLOTL,XSLOTU,                 PAR2  28
     7              XSLOTL,G,MIDJ,TOLSYS                                PAR2  29
      COMMON / C3 / EPHI,FPHI,GPHI,EPSI,FPSI,GPSI,MXISYS,               PAR2  30
     1              IPRSYS                                              PAR2  31
      REWIND 4                                                          PAR2  32
      REWIND 5                                                          PAR2  33
      REWIND 11                                                         PAR2  34
      REWIND 12                                                         PAR2  35
      NM1 = N - 1                                                       PAR2  36
      NP1 = N + 1                                                       PAR2  37
      MM1 = M - 1                                                       PAR2  38
```

```
      MP1 = M + 1
      NP2 = N + 2
      MP2 = M + 2
      NM2 = N - 2
      MM2 = M - 2
      MSTUM1 = MSTAGU - 1
      MSTUP1 = MSTAGU + 1
      MSTUP2 = MSTAGU + 2
      MSTUP3 = MSTAGU + 3
      MSTLM1 = MSTAGL - 1
      MSTLP1 = MSTAGL + 1
      MSTLP2 = MSTAGL + 2
      MSTLP3 = MSTAGL + 3
      MSLUP1 = MSLOTU + 1
      MSLLP1 = MSLOTL + 1
      NSTUP1 = NSTAGU + 1
      NSTUP2 = NSTAGU + 2
      NSTLP1 = NSTAGL + 1
      NSTLP2 = NSTAGL + 2
      READ(4)((Q(I,J),I=1,MP2),J=1,NP2)
      READ(5)((R(I,J),I=1,MP2),J=1,NP2)
      READ(12)((X(I,J),I=1,MP2),J=1,NP2)
C
C          COMPUTE STARTING ALPHAS ALONG MIDDLE STREAMLINE
C
      DO 5 I=1,MP1
      ASLOTU(I) = 1.0
    5 ASLOTL(I) = 1.0
      DO 10 I=2,MM1
      ESIMP(I) = EPHI(I)
      FSIMP(I) = FPHI(I)
   10 GSIMP(I) = GPHI(I)
      K = 2
      IF (FLAG1 .LE. 0 .AND. FLAG3 .LE. 0) K = MIDJ + 1
      J = K
      D1 = DPSI(J-1)
      D2 = DPSI(J)
      DO 20 I=2,MP1
      DER = D1 * Q(I,J+1) / ( D2 * D1 + D2 * D2 )
      DER = DER - D2 * Q(I,J-1) / ( D1 * D1 + D1 * D2 )
      DER = DER - ( D1 - D2 ) * Q(I,J) / ( D1 * D2 )
   20 GRAND(I) = R(I,J) * DER
      IMINSI = 2
      IMAXSI = MP1
      DO 30 I=2,MP1
   30 DX(I) = DPHI(I)
      CALL SIMP
      DO 40 I=2,MP1
   40 ALPHA(I,J) = GIFT(I)
C
C          COMPUTE ALPHA INTEGRANDS AND STORE THEM IN R1
C
      JOT = 2
      IF (FLAG3 .GT. 0) JOT = 3
      DO 70 I=2,MP1
      D1 = DPHI(I-1)
      D2 = DPHI(I)
      DO 70 J=JOT,NP1
      DER = D1 * Q(I+1,J) / ( D2 * D1 + D2 * D2 )
      DER = DER - D2 * Q(I-1,J) / ( D1 * D1 + D1 * D2 )
```

```
      DER = DER - ( D1 - D2 ) * Q(I,J) / ( D1 * D2 )
      R1(I,J) = - DER / R(I,J)
      DER = D1 * R(I+1,J) / ( D2 * D1 + D2 * D2 )
      DER = DER - D2 * R(I-1,J) / ( D1 * D1 + D1 * D2 )
      DER = DER - ( D1 - D2 ) * R(I,J) / ( D1 * D2 )
   70 R1(I,J) = - DER / ( R(I,J) * R(I,J) ) + R1(I,J)
      IF (FLAG3 .LE. 0) GO TO 60
      DO 50 I=2,MP1
   50 R1(I,2) = 0.0
   60 CONTINUE
C
C           COMPUTE ALPHAS IN THE UPPER PORTION OF THE DIFFUSER
C
      I = MSTAGU
      J = NSTUP1
      D1 = DPHI(I-1)
      D2 = DPHI(I)
      DER = ( Q(I,J) - Q(I-1,J) ) / D1
      R1(I,J) = - DER / R(I,J)
      DER = D1 * R(I+1,J) / ( D2 * D1 + D2 * D2 )
      DER = DER - D2 * R(I-1,J) / ( D1 * D1 + D1 * D2 )
      DER = DER - ( D1 - D2 ) * R(I,J) / ( D1 * D2 )
      R1(I,J) = - DER / ( R(I,J) * R(I,J) ) + R1(I,J)
      I = MSTUP2
      J = NSTUP1
      D1 = DPHI(I-1)
      D2 = DPHI(I)
      DER = ( Q(I+1,J) - Q(I,J) ) / D2
      R1(I,J) = - DER / R(I,J)
      DER = D1 * R(I+1,J) / ( D2 * D1 + D2 * D2 )
      DER = DER - D2 * R(I-1,J) / ( D1 * D1 + D1 * D2 )
      DER = DER - ( D1 - D2 ) * R(I,J) / ( D1 * D2 )
      R1(I,J) = - DER / ( R(I,J) * R(I,J) ) + R1(I,J)
      IF ( FLAG1 .GT. 0 .OR. FLAG3 .GT. 0 ) GO TO 80
      I = MSTAGL
      J = NSTLP1
      D1 = DPHI(I-1)
      D2 = DPHI(I)
      DER = ( Q(I,J) - Q(I-1,J) ) / D1
      R1(I,J) = - DER / R(I,J)
      DER = D1 * R(I+1,J) / ( D2 * D1 + D2 * D2 )
      DER = DER - D2 * R(I-1,J) / ( D1 * D1 + D1 * D2 )
      DER = DER - ( D1 - D2 ) * R(I,J) / ( D1 * D2 )
      R1(I,J) = - DER / ( R(I,J) * R(I,J) ) + R1(I,J)
      I = MSTLP2
      J = NSTLP1
      D1 = DPHI(I-1)
      D2 = DPHI(I)
      DER = ( Q(I+1,J) - Q(I,J) ) / D2
      R1(I,J) = - DER / R(I,J)
      DER = D1 * R(I+1,J) / ( D2 * D1 + D2 * D2 )
      DER = DER - D2 * R(I-1,J) / ( D1 * D1 + D1 * D2 )
      DER = DER - ( D1 - D2 ) * R(I,J) / ( D1 * D2 )
      R1(I,J) = - DER / ( R(I,J) * R(I,J) ) + R1(I,J)
   80 DO 90 J=2,NM1
      ESIMP(J) = EPSI(J)
      FSIMP(J) = FPSI(J)
   90 GSIMP(J) = GPSI(J)
      DO 100 J=2,NP1
  100 DX(J) = DPSI(J)
```

```
      IMINSI = K
      IMAXSI = NP1
      DO 120 I=2,MP1
      DO 110 J=K,NP1
  110 GRAND(J) = R1(I,J)
      IF (I .GT. MSTUP1) GRAND(NSTUP2) = GRAND(NSTAGU)
      CALL SIMP
      TEMP = ALPHA(I,K)
      DO 120 J=K,NP1
  120 ALPHA(I,J) = GIFT(J) + TEMP
C
C          COMPUTE ALPHAS IN UPPER SLOT
C
      DO 150 I=2,MM1
      ESIMP(I) = EPHI(I)
      FSIMP(I) = FPHI(I)
  150 GSIMP(I) = GPHI(I)
      DO 160 I=2,MP1
  160 DX(I) = DPHI(I)
      IMINSI = MSTUM1
      IMAXSI = MSLUP1
      J = NP1
      D1 = DPSI(N)
      D2 = D1
      DO 170 I=MSTUM1,MSLUP1
      DER = D1 * Q(I,J+1) / ( D2 * D1 + D2 * D2 )
      DER = DER - D2 * Q(I,J-1) / ( D1 * D1 + D1 * D2 )
      DER = DER - ( D1 - D2 ) * Q(I,J) / ( D1 * D2 )
  170 GRAND(I) = R(I,J) * DER
      CALL SIMP
      DO 180 I=MSTAGU,MSLUP1
  180 ALPHA(I,J) = ALPHA(MSTUM1,J) + GIFT(I)
      I = MSTUP2
      J = NSTUP1
      D1 = DPHI(I-1)
      D2 = DPHI(I)
      DER = (QSLOTU(I+1) - QSLOTU(I) ) / D2
      R1(I,J) = - DER / RSLOTU(I)
      DER = D1 * RSLOTU(I+1) / ( D2 * D1 + D2 * D2 )
      DER = DER - D2 * RSLOTU(I-1) / ( D1 * D1 + D1 * D2 )
      DER = DER - ( D1 - D2 ) * RSLOTU(I) / ( D1 * D2 )
      R1(I,J) = - DER / ( RSLOTU(I) * RSLOTU(I) ) + R1(I,J)
      DO 190 I=MSTUP3,MSLUP1
      D1 = DPHI(I-1)
      D2 = DPHI(I)
      DER = D1 * QSLOTU(I+1) / ( D2 * D1 + D2 * D2 )
      DER = DER - D2 * QSLOTU(I-1) / ( D1 * D1 + D1 * D2 )
      DER = DER - ( D1 - D2 ) * QSLOTU(I) / ( D1 * D2 )
      R1(I,J) = - DER / RSLOTU(I)
      DER = D1 * RSLOTU(I+1) / ( D2 * D1 + D2 * D2 )
      DER = DER - D2 * RSLOTU(I-1) / ( D1 * D1 + D1 * D2 )
      DER = DER - ( D1 - D2 ) * RSLOTU(I) / ( D1 * D2 )
  190 R1(I,J) = - DER / ( RSLOTU(I) * RSLOTU(I) ) + R1(I,J)
      DO 200 J=2,NM1
      ESIMP(J) = EPSI(J)
      FSIMP(J) = FPSI(J)
  200 GSIMP(J) = GPSI(J)
      DO 205 J=2,N
  205 DX(J) = DPSI(J)
      IMINSI = NSTUP1
```

```
      IMAXSI = NP1
      DO 220 I=MSTAGU,MSLUP1
      DO 210 J=NSTUP1,NP1
  210 GRAND(J) = R1(I,J)
      CALL SIMP
      TEMP = ALPHA(I,NP1) - GIFT(NP1)
      DO 215 J=NSTUP2,NP1
  215 ALPHA(I,J) = GIFT(J) + TEMP
  220 ASLOTU(I) = GIFT(NSTUP1) + TEMP
      I = MSTUP1
      J = NSTUP1
      ALPHA(I,J) = ASLOTU(I)
C
C         COMPUTE ALPHAS IN LOWER PORTION OF DIFFUSER
C
      IF ( FLAG1 .GT. 0 .OR. FLAG3 .GT. 0 ) GO TO 400
      DO 240 I=2,MSTAGL
      DO 230 J=2,K
  230 GRAND(J) = R1(I,J)
      KM1 = K - 1
      IMINSI = 2
      IMAXSI = K
      CALL SIMP
      TEMP = ALPHA(I,K) - GIFT(K)
      DO 240 J=2,KM1
  240 ALPHA(I,J) = GIFT(J) + TEMP
      I = MSTLP1
      DO 260 J=NSTLP2,K
  260 GRAND(J) = R1(I,J)
      IMINSI = NSTLP2
      IMAXSI = K
      CALL SIMP
      TEMP = ALPHA(I,K) - GIFT(K)
      DO 270 J=NSTLP2,KM1
  270 ALPHA(I,J) = GIFT(J) + TEMP
      IMINSI = NSTLP1
      IMAXSI = K
      DO 300 I=MSTLP2,MP1
      DO 290 J=NSTLP1,K
  290 GRAND(J) = R1(I,J)
      CALL SIMP
      TEMP = ALPHA(I,K) - GIFT(K)
      DO 300 J=NSTLP1,KM1
  300 ALPHA(I,J) = GIFT(J) + TEMP
C
C         COMPUTE ALPHAS INSIDE LOWER SLOT
C
      DO 310 I=2,MM1
      ESIMP(I) = EPHI(I)
      FSIMP(I) = FPHI(I)
  310 GSIMP(I) = GPHI(I)
      DO 320 I=2,MP1
  320 DX(I) = DPHI(I)
      IMINSI = MSTLM1
      IMAXSI = MSLLP1
      J = 2
      D1 = DPSI(2)
      D2 = D1
      DO 330 I=MSTLM1,MSLLP1
      DER = D1 * Q(I,J+1) / ( D2 * D1 + D2 * D2)
```

```
      DER = DER - D2 * Q(I,J-1) / ( D1 * D1 + D1 * D2 )
      DER = DER - ( D1 - D2 ) * Q(I,J) / ( D1 * D2 )
  330 GRAND(I) = R(I,J) * DER
      CALL SIMP
      DO 340 I=MSTAGL,MSLLP1
  340 ALPHA(I,J) = ALPHA(MSTLM1,J) + GIFT(I)
      I = MSTLP2
      J = NSTUP1
      D1 = DPHI(I-1)
      D2 = DPHI(I)
      DER = ( QSLOTL(I+1) - QSLOTL(I) ) / D2
      R1(I,J) = - DER / RSLOTL(I)
      DER = D1 * RSLOTL(I+1) / ( D2 * D1 + D2 * D2 )
      DER = DER - D2 * RSLOTL(I-1) / ( D1 * D1 + D1 * D2 )
      DER = DER - ( D1 - D2 ) * RSLOTL(I) / ( D1 * D2 )
      R1(I,J) = - DER / ( RSLOTL(I) * RSLOTL(I) ) + R1(I,J)
      DO 350 I=MSTLP3,MSLLP1
      D1 = DPHI(I-1)
      D2 = DPHI(I)
      DER = D1 * QSLOTL(I+1) / ( D2 * D1 + D2 * D2 )
      DER = DER - D2 * QSLOTL(I-1) / ( D1 * D1 + D1 * D2 )
      DER = DER - ( D1 - D2 ) * QSLOTL(I) / ( D1 * D2 )
      R1(I,J) = - DER / RSLOTL(I)
      DER = D1 * RSLOTL(I+1) / ( D2 * D1 + D2 * D2 )
      DER = DER - D2 * RSLOTL(I-1) / ( D1 * D1 + D1 * D2 )
      DER = DER - ( D1 - D2 ) * RSLOTL(I) / ( D1 * D2 )
  350 R1(I,J) = - DER / ( RSLOTL(I) * RSLOTL(I) ) + R1(I,J)
      DO 360 J=2,NM1
      ESIMP(J) = EPSI(J)
      FSIMP(J) = FPSI(J)
  360 GSIMP(J) = GPSI(J)
      DO 370 J=2,N
  370 DX(J) = DPSI(J)
      IMINSI = 2
      IMAXSI = NSTLP1
      DO 390 I=MSTAGL,MSLLP1
      DO 380 J=2,NSTLP1
  380 GRAND(J) = R1(I,J)
      CALL SIMP
      DO 385 J=3,NSTAGL
  385 ALPHA(I,J) = ALPHA(I,2) + GIFT(J)
  390 ASLOTL(I) = GIFT(NSTLP1) + ALPHA(I,2)
      I = MSTAGL
      J = NSTLP1
      ALPHA(I,J) = ASLOTL(I)
      ALPHA(I+1,J) = ALPHA(I,J)
  400 CONTINUE
C
C          COMPUTE INLET R COORDINATES
C
      DO 410 J=2,NM1
      ESIMP(J) = EPSI(J)
      FSIMP(J) = FPSI(J)
  410 GSIMP(J) = GPSI(J)
      DO 420 J=2,N
  420 DX(J) = DPSI(J)
      IMINSI = 2
      IMAXSI = NP1
      IF (RADIN .GT. 0.0) R(2,2) = RADIN
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) R(2,2) = 0.0
```

```
      DO 430 J=2,NP1                                                    PAR2 339
      P= Q(2,J)                                                         PAR2 340
  430 GRAND(J) = 2. * COS(ALPHA(2,J)) / DEXP(P)                         PAR2 341
      CALL SIMP                                                         PAR2 342
      POW = .5                                                          PAR2 343
      DIV = 1.0                                                         PAR2 344
      IF (FLAG1 .GT. 0) POW = 1.0                                       PAR2 345
      IF (FLAG1 .GT. 0) DIV = 2.0                                       PAR2 346
      DO 440 J=3,NP1                                                    PAR2 347
      RAT = ABS( GIFT(J) / DIV + R(2,2) * R(2,2) )                      PAR2 348
  440 R(2,J) = ( RAT )**POW                                             PAR2 349
C                                                                       PAR2 350
C           COMPUTE R ALONG MIDDLE STREAMLINE                           PAR2 351
C                                                                       PAR2 352
      DO 450 I=2,MM1                                                    PAR2 353
      ESIMP(I) = EPHI(I)                                                PAR2 354
      FSIMP(I) = FPHI(I)                                                PAR2 355
  450 GSIMP(I) = GPHI(I)                                                PAR2 356
      DO 460 I=2,M                                                      PAR2 357
  460 DX(I) = DPHI(I)                                                   PAR2 358
      IMINSI = 2                                                        PAR2 359
      IMAXSI = MP1                                                      PAR2 360
      J = K                                                             PAR2 361
      DO 470 I=2,MP1                                                    PAR2 362
      P = Q(I,J)                                                        PAR2 363
  470 GRAND(I) = SIN(ALPHA(I,J)) / DEXP(P)                              PAR2 364
      CALL SIMP                                                         PAR2 365
      DO 480 I=3,MP1                                                    PAR2 366
  480 R(I,J) = R(2,J) + GIFT(I)                                         PAR2 367
C                                                                       PAR2 368
C           COMPUTE R THROUGHOUT UPPER PORTION OF DIFFUSER              PAR2 369
C                                                                       PAR2 370
      DO 490 J=2,NM1                                                    PAR2 371
      ESIMP(J) = EPSI(J)                                                PAR2 372
      FSIMP(J) = FPSI(J)                                                PAR2 373
  490 GSIMP(J) = GPSI(J)                                                PAR2 374
      DO 500 J=2,N                                                      PAR2 375
  500 DX(J) = DPSI(J)                                                   PAR2 376
      IMINSI = K                                                        PAR2 377
      IMAXSI = NP1                                                      PAR2 378
      DO 520 I=3,MP1                                                    PAR2 379
      DO 510 J=K,NP1                                                    PAR2 380
      P = Q(I,J)                                                        PAR2 381
  510 GRAND(J) = 2. * COS(ALPHA(I,J)) / DEXP(P)                         PAR2 382
      IF (I .GT. MSTUP1) GRAND(NSTUP2) = GRAND(NSTAGU)                  PAR2 383
      CALL SIMP                                                         PAR2 384
      KP1 = K + 1                                                       PAR2 385
      DO 520 J= KP1,NP1                                                 PAR2 386
      RAT = ABS( GIFT(J) / DIV + R(I,K) * R(I,K) )                      PAR2 387
  520 R(I,J) = ( RAT )**POW                                             PAR2 388
C                                                                       PAR2 389
C           COMPUTE R COORDINATES INSIDE UPPER SLOT                     PAR2 390
C                                                                       PAR2 391
  560 DO 570 I=2,MM1                                                    PAR2 392
      ESIMP(I) = EPHI(I)                                                PAR2 393
      FSIMP(I) = FPHI(I)                                                PAR2 394
  570 GSIMP(I) = GPHI(I)                                                PAR2 395
      DO 580 I=2,M                                                      PAR2 396
  580 DX(I) = DPHI(I)                                                   PAR2 397
      IMINSI = MSTUM1                                                   PAR2 398
```

```
      IMAXSI = MSLUP1                                                   PAR2 399
      J = NP1                                                           PAR2 400
      DO 590 I=MSTUM1,MSLUP1                                            PAR2 401
      P = Q(I,J)                                                        PAR2 402
  590 GRAND(I) = SIN(ALPHA(I,J)) / DEXP(P)                              PAR2 403
      CALL SIMP                                                         PAR2 404
      DO 600 I=MSTAGU,MSLUP1                                            PAR2 405
  600 R(I,J) = R(MSTUM1,J) + GIFT(I)                                    PAR2 406
      DO 605 J=2,NM1                                                    PAR2 407
      ESIMP(J) = EPSI(J)                                                PAR2 408
      FSIMP(J) = FPSI(J)                                                PAR2 409
  605 GSIMP(J) = GPSI(J)                                                PAR2 410
      DO 610 J=2,N                                                      PAR2 411
  610 DX(J) = DPSI(J)                                                   PAR2 412
      IMINSI = NSTUP1                                                   PAR2 413
      IMAXSI = NP1                                                      PAR2 414
      DO 650 I=MSTAGU,MSLUP1                                            PAR2 415
      DO 620 J=NSTUP2,NP1                                               PAR2 416
      P = Q(I,J)                                                        PAR2 417
  620 GRAND(J) = 2. * COS(ALPHA(I,J)) / DEXP(P)                         PAR2 418
      P = QSLOTU(I)                                                     PAR2 419
      GRAND(NSTUP1) = 2. * COS(ASLOTU(I)) / DEXP(P)                     PAR2 420
      CALL SIMP                                                         PAR2 421
      TEMP = R(I,NP1) * R(I,NP1) - GIFT(NP1)                            PAR2 422
      DO 630 J=NSTUP2,N                                                 PAR2 423
  630 R(I,J) = ( ABS( GIFT(J) + TEMP ) )**.5                            PAR2 424
      RSLOTU(I) = ( ABS( GIFT(NSTUP1) + TEMP ) )**.5                    PAR2 425
      IF (FLAG1 .LE. 0) GO TO 650                                       PAR2 426
      TEMP = R(I,NP1) - GIFT(NP1) / 2.                                  PAR2 427
      DO 640 J=NSTUP2,N                                                 PAR2 428
  640 R(I,J) = GIFT(J) / 2. + TEMP                                      PAR2 429
      RSLOTU(I) = GIFT(NSTUP1) / 2. + TEMP                              PAR2 430
  650 CONTINUE                                                          PAR2 431
      I = MSTUP1                                                        PAR2 432
      J = NSTUP1                                                        PAR2 433
      P = Q(I-1,J)                                                      PAR2 434
      R(I,J) = R(I-1,J) + SIN(ALPHA(I-1,J)) * DPHI(I-1) / DEXP(P)       PAR2 435
C                                                                       PAR2 436
C         COMPUTE R OVER LOWER PORTION OF DIFFUSER                      PAR2 437
C                                                                       PAR2 438
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 840                     PAR2 439
      DO 670 I=2,MSTAGL                                                 PAR2 440
      DO 655 J=2,K                                                      PAR2 441
      P = Q(I,J)                                                        PAR2 442
  655 GRAND(J) = 2. * COS(ALPHA(I,J)) / DEXP(P)                         PAR2 443
      IMINSI = 2                                                        PAR2 444
      IMAXSI = K                                                        PAR2 445
      CALL SIMP                                                         PAR2 446
      TEMP = R(I,K) * R(I,K) - GIFT(K)                                  PAR2 447
      DO 660 J=2,KM1                                                    PAR2 448
  660 R(I,J) = ( ABS( GIFT(J) + TEMP ) )**.5                            PAR2 449
  670 CONTINUE                                                          PAR2 450
      I = MSTLP1                                                        PAR2 451
      DO 675 J=NSTLP2,K                                                 PAR2 452
      P = Q(I,J)                                                        PAR2 453
  675 GRAND(J) = 2. * COS(ALPHA(I,J)) / DEXP(P)                         PAR2 454
      IMINSI = NSTLP2                                                   PAR2 455
      IMAXSI = K                                                        PAR2 456
      CALL SIMP                                                         PAR2 457
      TEMP = R(I,K) * R(I,K) - GIFT(K)                                  PAR2 458
```

```
      DO 680 J=NSTLP2,KM1
  680 R(I,J) = ( ABS( GIFT(J) + TEMP ) )**.5
  690 CONTINUE
      DO 702 I=MSTLP2,MP1
      DO 695 J=NSTLP1,K
      P = Q(I,J)
  695 GRAND(J) = 2. * COS(ALPHA(I,J) ) / DEXP(P)
      IMINSI = NSTLP1
      IMAXSI = K
      CALL SIMP
      TEMP = R(I,K) * R(I,K) - GIFT(K)
      DO 697 J=NSTLP1,KM1
  697 R(I,J) = ( ABS( GIFT(J) + TEMP ) )**.5
  702 CONTINUE
C
C         COMPUTE R INSIDE LOWER SLOTS
C
      DO 740 I=2,MM1
      ESIMP(I) = EPHI(I)
      FSIMP(I) = FPHI(I)
  740 GSIMP(I) = GPHI(I)
      DO 750 I=2,M
  750 DX(I) = DPHI(I)
      DO 760 I=MSTLM1,MSLLP1
      P = Q(I,2)
  760 GRAND(I) = SIN(ALPHA(I,2)) / DEXP(P)
      IMINSI = MSTLM1
      IMAXSI = MSLLP1
      CALL SIMP
      DO 770 I=MSTAGL,MSLLP1
  770 R(I,2) = R(MSTLM1,2) + GIFT(I)
      DO 780 J=2,NM1
      ESIMP(J) = EPSI(J)
      FSIMP(J) = FPSI(J)
  780 GSIMP(J) = GPSI(J)
      DO 790 J=2,N
  790 DX(J) = DPSI(J)
      IMINSI = 2
      IMAXSI = NSTLP1
      DO 830 I=MSTAGL,MSLLP1
      DO 800 J=2,NSTAGL
      P = Q(I,J)
  800 GRAND(J) = 2. * COS(ALPHA(I,J)) / DEXP(P)
      P = QSLOTL(I)
      GRAND(NSTLP1) = 2. * COS(ASLOTL(I)) / DEXP(P)
      CALL SIMP
      DO 810 J=3,NSTAGL
      RAT = ABS( GIFT(J) + R(I,2) * R(I,2) )
  810 R(I,J) = ( RAT )**.5
      RAT = ABS( GIFT(NSTLP1) + R(I,2) * R(I,2) )
      RSLOTL(I) = ( RAT )**.5
      IF (FLAG1 .GT. 0) GO TO 830
      DO 820 J=3,NSTAGL
  820 R(I,J) = R(I,2) + GIFT(J) / 2.
      RSLOTL(I) = R(I,2) + GIFT(NSTLP1) / 2.
  830 CONTINUE
      I = MSTLP1
      J = NSTLP1
      P = Q(I-1,J)
      R(I,J) = R(I-1,J) + SIN(ALPHA(I-1,J)) * DPHI(I-1) / DEXP(P)
```

```
      R(I-1,J) = RSLCTL(I-1)
  840 CONTINUE
C
C           COMPUTE CONVERGENCE CRITERION
C
      REWIND 5
      READ(5)((R1(I,J),I=1,MP2),J=1,NP2)
      EPS = 0.0
      DO  845 J=2,NP1
      DO  845 I=2,MP1
      EPSTR = ABS(R1(I,J) - R(I,J))
      IF (EPSTR .LE. EPS) GO TO 845
      EPS = EPSTR
  845 CONTINUE
C
C           COMPUTE INLET X COORDINATES
C
      IF (ITER .LT. MXISYS .AND. EPS .GT. TOLSYS) GO TO 3000
      DO 848 I=1,MP1
      XSLOTU(I) = 1.0
  848 XSLOTL(I) = 1.0
      DO 850 J=2,NM1
      ESIMP(J) = EPSI(J)
  850 GSIMP(J) = GPSI(J)
      FSIMP(J) = FPSI(J)
      DO 860 J=2,N
  860 DX(J) = DPSI(J)
      IMINSI = 2
      IMAXSI = NP1
      JOT = 2
      IF (FLAG3 .GT. 0) JOT = 3
      DO 870 J=JOT,NP1
      P = Q(2,J)
      GRAND(J) = - SIN(ALPHA(2,J)) / DEXP(P)
  870 IF (FLAG1 .LE. 0) GRAND(J) = GRAND(J) / R(2,J)
      IF (FLAG3 .GT. 0) GRAND(2) = 0.0
      CALL SIMP
      DO 880 J=2,NP1
  880 X(2,J) = GIFT(J)
C
C           COMPUTE X ALONG THE MIDDLE STREAMLINE
C
      DO 890 I=2,MM1
      ESIMP(I) = EPHI(I)
      FSIMP(I) = FPHI(I)
  890 GSIMP(I) = GPHI(I)
      DO 900 I=2,M
  900 DX(I) = DPHI(I)
      IMINSI = 2
      IMAXSI = MP1
      J = K
      DO 910 I=2,MP1
      P = Q(I,J)
  910 GRAND(I) = COS(ALPHA(I,J)) / DEXP(P)
      CALL SIMP
      DO 920 I=3,MP1
  920 X(I,J) = X(2,J) + GIFT(I)
C
C           COMPUTE X THROUGHOUT UPPER PORTION OF DIFFUSER
C
```

```
      DO 930 J=2,NM1                                                    PAR2 579
      ESIMP(J) = EPSI(J)                                                PAR2 580
      FSIMP(J) = FPSI(J)                                                PAR2 581
  930 GSIMP(J) = GPSI(J)                                                PAR2 582
      DO 940 J=2,N                                                      PAR2 583
  940 DX(J) = DPSI(J)                                                   PAR2 584
      IMINSI = K                                                        PAR2 585
      IMAXSI = NP1                                                      PAR2 586
      JET = K                                                           PAR2 586
      IF (FLAG3 .GT. 0) JET = 3                                         PAR2 586
      DO 960 I=3,MP1                                                    PAR2 587
      DO 950 J=JET,NP1                                                  PAR2 588
      P = Q(I,J)                                                        PAR2 589
      GRAND(J) = - SIN(ALPHA(I,J)) / DEXP(P)                            PAR2 590
  950 IF (FLAG1 .LE. 0) GRAND(J) = GRAND(J) / R(I,J)                    PAR2 591
      IF (FLAG3 .GT. 0) GRAND(2) = 0.0                                  PAR2 592
      IF (I .GT. MSTUP1) GRAND(NSTUP2) = GRAND(NSTAGU)                  PAR2 593
      CALL SIMP                                                         PAR2 594
      KP1 = K + 1                                                       PAR2 595
      DO 960 J=KP1,NP1                                                  PAR2 596
  960 X(I,J) = X(I,K) + GIFT(J)                                         PAR2 597
C                                                                       PAR2 598
C          COMPUTE X COORDINATES INSIDE UPPER SLOT                      PAR2 599
C                                                                       PAR2 600
      DO 990 I=2,MM1                                                    PAR2 601
      ESIMP(I) = EPHI(I)                                                PAR2 602
      FSIMP(I) = FPHI(I)                                                PAR2 603
  990 GSIMP(I) = GPHI(I)                                                PAR2 604
      DO 1000 I=2,M                                                     PAR2 605
 1000 DX(I) = DPHI(I)                                                   PAR2 606
      IMINSI = MSTUM1                                                   PAR2 607
      IMAXSI = MSLUP1                                                   PAR2 608
      J = NP1                                                           PAR2 609
      DO 1010 I=MSTUM1,MSLUP1                                           PAR2 610
      P = Q(I,J)                                                        PAR2 611
 1010 GRAND(I) = COS(ALPHA(I,J)) / DEXP(P)                              PAR2 612
      CALL SIMP                                                         PAR2 613
      DO 1020 I=MSTAGU,MSLUP1                                           PAR2 614
 1020 X(I,J) = X(MSTUM1,J) + GIFT(I)                                    PAR2 615
      DO 1030 J=2,NM1                                                   PAR2 616
      ESIMP(J) = EPSI(J)                                                PAR2 617
      FSIMP(J) = FPSI(J)                                                PAR2 618
 1030 GSIMP(J) = GPSI(J)                                                PAR2 619
      DO 1035 J=2,N                                                     PAR2 620
 1035 DX(J) = DPSI(J)                                                   PAR2 621
      IMINSI = NSTUP1                                                   PAR2 622
      IMAXSI = NP1                                                      PAR2 623
      DO 1060 I=MSTAGU,MSLUP1                                           PAR2 624
      DO 1040 J=NSTUP2,NP1                                              PAR2 625
      P = Q(I,J)                                                        PAR2 626
      GRAND(J) = - SIN(ALPHA(I,J)) / DEXP(P)                            PAR2 627
 1040 IF (FLAG1 .LE. 0) GRAND(J) = GRAND(J) / R(I,J)                    PAR2 628
      P = QSLOTU(I)                                                     PAR2 629
      GRAND(NSTUP1) = - SIN(ASLOTU(I)) / DEXP(P)                        PAR2 630
      IF ( FLAG1 .LE. 0 ) GRAND(NSTUP1) = GRAND(NSTUP1) /               PAR2 631
     1RSLOTU(I)                                                         PAR2 632
      CALL SIMP                                                         PAR2 633
      TEMP = X(I,NP1) - GIFT(NP1)                                       PAR2 634
      DO 1050 J=NSTUP2,N                                                PAR2 635
 1050 X(I,J) = GIFT(J) + TEMP                                           PAR2 636
 1060 XSLOTU(I) = GIFT(NSTUP1) + TEMP                                   PAR2 637
      I = MSTUP1                                                        PAR2 638
```

```
      J = NSTUP1                                                        PAR2 639
      P = Q(I-1,J)                                                      PAR2 640
      X(I,J) = X(I-1,J) + COS(ALPHA(I-1,J)) * DPHI(I-1) / DEXP(P)       PAR2 641
C                                                                       PAR2 642
C         COMPUTE X OVER LOWER PORTION OF DIFFUSER                      PAR2 643
C                                                                       PAR2 644
      IF ( FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 3000                   PAR2 645
      DO 1070 I=2,MSTAGL                                                PAR2 646
      DO 1065 J=2,K                                                     PAR2 647
      P = Q(I,J)                                                        PAR2 648
 1065 GRAND(J) = - SIN(ALPHA(I,J)) / ( DEXP(P) * R(I,J))                PAR2 649
      IMINSI = 2                                                        PAR2 650
      IMAXSI = K                                                        PAR2 651
      CALL SIMP                                                         PAR2 652
      TEMP = X(I,K) - GIFT(K)                                           PAR2 653
      DO 1070 J=2,KM1                                                   PAR2 654
 1070 X(I,J) = GIFT(J) + TEMP                                           PAR2 655
      I = MSTLP1                                                        PAR2 656
      DO 1075 J =NSTLP2,NP1                                             PAR2 657
      P = Q(I,J)                                                        PAR2 658
 1075 GRAND(J) = - SIN(ALPHA(I,J)) / ( DEXP(P) * R(I,J))                PAR2 659
      IMINSI = NSTLP2                                                   PAR2 660
      IMAXSI = K                                                        PAR2 661
      CALL SIMP                                                         PAR2 662
      TEMP = X(I,K) - GIFT(K)                                           PAR2 663
      DO 1080 J=NSTLP2,KM1                                              PAR2 664
 1080 X(I,J) = GIFT(J) + TEMP                                           PAR2 665
      DO 1090 I=MSTLP2,MP1                                              PAR2 666
      DO 1085 J=NSTLP1,K                                                PAR2 667
      P = Q(I,J)                                                        PAR2 668
 1085 GRAND(J) = - SIN(ALPHA(I,J)) / ( DEXP(P) * R(I,J) )               PAR2 669
      IMINSI = NSTLP1                                                   PAR2 670
      IMAXSI = K                                                        PAR2 671
      CALL SIMP                                                         PAR2 672
      TEMP = X(I,K) - GIFT(K)                                           PAR2 673
      DO 1090 J=NSTLP1,KM1                                              PAR2 674
 1090 X(I,J) = GIFT(J) + TEMP                                           PAR2 675
C                                                                       PAR2 676
C         COMPUTE X INSIDE LOWER SLOTS                                  PAR2 677
C                                                                       PAR2 678
      DO 1120 I=2,MM1                                                   PAR2 679
      ESIMP(I) = EPHI(I)                                                PAR2 680
      FSIMP(I) = FPHI(I)                                                PAR2 681
 1120 GSIMP(I) = GPHI(I)                                                PAR2 682
      DO 1130 I=2,M                                                     PAR2 683
 1130 DX(I) = DPHI(I)                                                   PAR2 684
      DO 1140 I=MSTLM1,MSLLP1                                           PAR2 685
      P = Q(I,2)                                                        PAR2 686
 1140 GRAND(I) = COS(ALPHA(I,2)) / DEXP(P)                              PAR2 687
      IMINSI = MSTLM1                                                   PAR2 688
      IMAXSI = MSLLP1                                                   PAR2 689
      CALL SIMP                                                         PAR2 690
      DO 1150 I=MSTAGL,MSLLP1                                           PAR2 691
 1150 X(I,2) = X(MSTLM1,2) + GIFT(I)                                    PAR2 692
      DO 1160 J=2,NM1                                                   PAR2 693
      ESIMP(J) = EPSI(J)                                                PAR2 694
      FSIMP(J) = FPSI(J)                                                PAR2 695
 1160 GSIMP(J) = GPSI(J)                                                PAR2 696
      DO 1170 J=2,N                                                     PAR2 697
 1170 DX(J) = DPSI(J)                                                   PAR2 698
```

```
      IMINSI = 2                                                        PAR2 699
      IMAXSI = NSTLP1                                                   PAR2 700
      DO 1200 I=MSTAGL,MSLLP1                                           PAR2 701
      DO 1180 J=2,NSTAGL                                                PAR2 702
      P = Q(I,J)                                                        PAR2 703
      GRAND(J) = - SIN(ALPHA(I,J)) / DEXP(P)                            PAR2 704
 1180 IF (FLAG1 .LE. 0) GRAND(J) = GRAND(J) / R(I,J)                    PAR2 705
      P = QSLOTL(I)                                                     PAR2 706
      GRAND(NSTLP1) = - SIN(ASLOTL(I)) / DEXP(P)                        PAR2 707
      IF (FLAG1 .LE. 0) GRAND(NSTLP1) = GRAND(NSTLP1) /                 PAR2 708
     1RSLOTL(I)                                                         PAR2 709
      CALL SIMP                                                         PAR2 710
      DO 1190 J=3,NSTAGL                                                PAR2 711
 1190 X(I,J) = X(I,2) + GIFT(J)                                         PAR2 712
 1200 XSLOTL(I) = X(I,2) + GIFT(NSTLP1)                                 PAR2 713
 1210 I = MSTLP1                                                        PAR2 714
      J = NSTLP1                                                        PAR2 715
      P = Q(I-1,J)                                                      PAR2 716
      X(I,J) = X(I-1,J) + COS(ALPHA(I-1,J)) * DPHI(I-1) / DEXP(P)       PAR2 717
 3000 CONTINUE                                                          PAR2 718
      ASLOTU(MSTUP1) = ALPHA(MSTUP1,NSTUP1)                             PAR2 719
      RSLOTU(MSTUP1) = R(MSTUP1,NSTUP1)                                 PAR2 720
      XSLOTU(MSTUP1) = X(MSTUP1,NSTUP1)                                 PAR2 721
      ASLOTL(MSTLP1) = ALPHA(MSTLP1,NSTLP1)                             PAR2 722
      RSLOTL(MSTLP1) = R(MSTLP1,NSTLP1)                                 PAR2 723
      XSLOTL(MSTLP1) = X(MSTLP1,NSTLP1)                                 PAR2 724
C                                                                       PAR2 725
C           SET UP BOUNDARY CONDITIONS FOR NEXT ITERATION               PAR2 726
C                                                                       PAR2 727
      IOT = MP1                                                         PAR2 728
      IF (FLAG1 .LE. 0 .AND. FLAG3 .LE. 0) IOT = MSLLP1                 PAR2 729
      DO 2010 I=2,IOT                                                   PAR2 730
      Y3 = R(I,4)                                                       PAR2 731
      Y2 = R(I,3)                                                       PAR2 732
      Y1 = R(I,2)                                                       PAR2 733
      D1 = DPSI(2)                                                      PAR2 734
      D2 = DPSI(3)                                                      PAR2 735
      DER = ( Y2 - Y1 ) * ( D1 + D2 ) / ( D1 * D2 )                     PAR2 736
      DER = DER - ( Y3 - Y1 ) * D1 / ( D1 * D2 + D2 * D2 )              PAR2 737
 2010 R(I,1) = Y2 - 2. * DER * D1                                       PAR2 738
      IOT = MSLUP1                                                      PAR2 739
      DO 2020 I=2,IOT                                                   PAR2 740
      Y3 = R(I,NP1)                                                     PAR2 741
      Y2 = R(I,N)                                                       PAR2 742
      Y1 = R(I,NM1)                                                     PAR2 743
      D1 = DPSI(NM1)                                                    PAR2 744
      D2 = DPSI(N)                                                      PAR2 745
      DER = ( Y3 - Y2 ) * ( D1 + D2 ) / ( D1 * D2 )                     PAR2 746
      DER = DER - ( Y3 - Y1 ) * D2 / ( D1 * D1 + D1 * D2 )              PAR2 747
 2020 R(I,NP2) = Y2 + 2. * DER * D2                                     PAR2 748
      DO 2030 J=2,NP1                                                   PAR2 749
      Y3 = R(4,J)                                                       PAR2 750
      Y2 = R(3,J)                                                       PAR2 751
      Y1 = R(2,J)                                                       PAR2 752
      D1 = DPHI(2)                                                      PAR2 753
      D2 = DPHI(3)                                                      PAR2 754
      DER = ( Y2 - Y1 ) * ( D1 + D2 ) / ( D1 * D2 )                     PAR2 755
      DER = DER - ( Y3 - Y1 ) * D1 / ( D1 * D2 + D2 * D2 )              PAR2 756
 2030 R(1,J) = Y2 - 2. * DER * D1                                       PAR2 757
      DO 2035 J=2,NP1                                                   PAR2 758
      Y3 = R(MP1,J)                                                     PAR2 759
```

```
      Y2 = R(M,J)
      Y1 = R(MM1,J)
      D1 = DPHI(MM1)
      D2 = DPHI(M)
      DER = ( Y3 - Y2 ) * ( D1 + D2 ) / ( D1 * D2 )
      DER = DER - ( Y3 - Y1 ) * D2 / ( D1 * D1 + D1 * D2 )
 2035 R(MP2,J) = Y2 + 2. * DER * D2
      IF (FLAG2 .LE. 0) GO TO 2050
      DO 2040 J=2,NP1
      R(1,J) = R(3,J)
 2040 R(MP2,J) = R(M,J)
 2050 CONTINUE
C
C         WRITE CONVERGENCE VARIABLES
C
      IF (FLAG1 .LE. 0) WRITE(3,2060)ITER,JTER,EPS,DEL
 2060 FORMAT(/T2,'ITER = ',T9,I5,T25,'JTER = ',T32,I5,T50,
     1'EPS = ',T56,E13.6,T75,'DEL = ',T80,E13.6)
C
C         CHECK  BELL  CHANNEL  SOLUTION
C
      IF (FLAG3 .LE. 0) GO TO 2130
      IF (EPS .GT. TOLSYS .AND. ITER .LT. MXISYS) GO TO 2130
      IMINSU = 2
      IMAXSU = NSTAGU - 1
      IMINSI = 2
      IMAXSI = NSTAGU + 1
      ERROR = 0.0
      DO 2090 I=2,MP1
      DO 2070 J=2,NSTAGU
 2070 DX(J) = SQRT( ( ABS( X(I,J+1) - X(I,J) ) )**2.
     1+ ( ABS( R(I,J+1) - R(I,J) ) )**2. )
      CALL SETUP
      NSTUP1 = NSTAGU + 1
      DO 2080 J=2,NSTUP1
 2080 GRAND(J) = R(I,J) * EXP(Q(I,J))
      CALL SIMP
      ERR = ABS( PSI(NSTUP1) - GIFT(NSTUP1) )
      IF ( ERR .GT. ERROR ) FNC = GIFT(NSTUP1)
 2090 IF ( ERR .GT. ERROR ) ERROR = ERR
      PCE = 100. * ERROR / PSI(NSTUP1)
      WRITE(3,2100)HEDR,DATE1,DATE2,DATE3,CASE
 2100 FORMAT(1H1//T57,'CLEMSON  UNIVERSITY',//T49,'MECHANICAL  ENGINEERI
     1NG  DEPARTMENT',///T2,15A4,T64,I2,T66,'/',T67,I2,T69,'/',T70,I2,
     2T76,'CASE  NO.  ',T87,I4///)
      WRITE(3,2110)
 2110 FORMAT(T2,'THE ACCURACY OF THE SOLUTION FOR THE',
     1T38,' BELL CHANNEL DESIGNED')
      WRITE(3,2120)ERROR,FNC,PCE
 2120 FORMAT(////T2,'ERROR = ',F10.6////
     1T2,'INCORRECT VALUE = ',F10.6////
     2T2,'PERCENT ERROR = ',F10.6)
 2130 CONTINUE
      REWIND 5
      REWIND 12
      WRITE(5)((R(I,J),I=1,MP2),J=1,NP2)
      WRITE(11)((ALPHA(I,J),I=1,MP2),J=1,NP2)
      WRITE(12)((X(I,J),I=1,MP2),J=1,NP2)
      RETURN
      END
```

```
C         **********      SUBROUTINE  OUTPUT      **********
C
C         THIS SUBROUTINE PROVIDES PRINTER OUTPUT OF THE
C         RESULTS OF THE MAIN PROGRAM
C
      SUBROUTINE OUTPUT
      DIMENSION DX(150),ESIMP(150),FSIMP(150),GSIMP(150),
     1          GRAND(150),GIFT(150),PHI(150),PSI(150),
     2          DPSI(150),DPHI(150),S(150),SBAR(150),
     3          S1(150),SBAR1(150),S2(150),SBAR2(150),
     4          HEDR(15),EPHI(150),FPHI(150),GPHI(150),
     5          EPSI(150),ASLOTL(150),XSLOTU(150),XSLOTL(150)
     6          ,FPSI(150),GPSI(150),RSLOTU(150),RSLOTL(150),
     7          QSLOTU(150),QSLOTL(150),ASLOTU(150)
      DIMENSION Q(150,50),R(150,50),X(150,50),ALPHA(150,50)
      DOUBLE PRECISION P
      INTEGER FLAG1,FLAG2,FLAG3,FLAG4,CASE,DATE1,DATE2,DATE3
      COMMON / C2 / FLAG1,FLAG2,FLAG3,FLAG4,HEDR,DATE1,DATE2
     1              ,DATE3,CASE,MXIDE,ICKDE,IPRDE,IMIN,IMAX,
     2              JMIN,JMAX,NUT,M,N,TOLDE,DEL,EPS,JTER,
     3              ITER,DPHI,DPSI,S,SBAR,RMULT,S1,S2,SBAR1,
     4              SBAR2,MSTAGU,NSTAGU,MSLOTU,MSTAGL,NSTAGL
     5              ,MSLOTL,RADIN,W,PHI,PSI,RSLOTU,RSLOTL,
     6              QSLOTU,QSLOTL,ASLOTU,ASLOTL,XSLOTU,
     7              XSLOTL,G,MIDJ
C
C        CHANGE Q BACK TO VELOCITY AND OUTPUT
C
      MP1 = M + 1
      NP1 = N + 1
      MM1 = M - 1
      NM1 = N - 1
      MP2 = M + 2
      NP2 = N + 2
      NSTLP1 = NSTAGL + 1
      NSTUP1 = NSTAGU + 1
      MSTUP1 = MSTAGU + 1
      MSTLP1 = MSTAGL + 1
      MSLLP1 = MSLOTL + 1
      MSLUP1 = MSLOTU + 1
      REWIND 4
      REWIND 5
      REWIND 11
      REWIND 12
      READ(4)((Q(I,J),I=1,MP2),J=1,NP2)
      READ(5)((R(I,J),I=1,MP2),J=1,NP2)
      READ(11)((ALPHA(I,J),I=1,MP2),J=1,NP2)
      READ(12)((X(I,J),I=1,MP2),J=1,NP2)
      IF (NUT .EQ. 1) GO TO 1000
      DO 10 I=2,MP1
      DO 10 J=2,NP1
      P = Q(I,J)
   10 Q(I,J) = DEXP(P)
      Q(MSTUP1,NSTUP1) = 0.0
      Q(MSTLP1,NSTLP1) = 0.0
C
C        OUTPUT Q ARRAY
C
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
   20 FORMAT(1H1//T57,'CLEMSON  UNIVERSITY',//T49,
```

```
     1'MECHANICAL  ENGINEERING  DEPARTMENT',///T2,15A4,T64,
     2I2,T66,'/',T67,I2,T69,'/',T70,I2,T76,'CASE  NO.  ',
     3T87,I4)
      WRITE(3,30)ITER,JTER,EPS,DEL
   30 FORMAT(///T1,'0VELOCITY THROUGHOUT DIFFUSER',
     1T46,'ITER = ',T53,I4,T61,'JTER = ',T68,I4,T76,'EPS = '
     2,T82,E13.6,T99,'DEL = ',T105,E13.6,///T7,'PHI',T66,
     3'PSI')
      WRITE(3,40)(PSI(IP),IP=2,NP1)
   40 FORMAT(//(13X,5(7X,E13.6)))
      DO 50 I=2,MP1
   50 WRITE(3,60)PHI(I),(Q(I,J),J=2,NP1)
   60 FORMAT(//1X,E13.6,5(7X,E13.6),/(14X,5(7X,E13.6)))
C
C           CHANGE Q BACK TO LOGARITHMIC FORM
C
      Q(MSTUP1,NSTUP1) = 1.0
      Q(MSTLP1,NSTLP1) = 1.0
      DO 70 I=2,MP1
      DO 70 J=2,NP1
      P = Q(I,J)
   70 Q(I,J) = DLOG(P)
C
C           OUTPUT VELOCITY IN SLOTS
C
      DO 75 I=2,MP1
      P = QSLOTL(I)
      QSLOTL(I) = DEXP(P)
      P = QSLOTU(I)
   75 QSLOTU(I) = DEXP(P)
      QSLOTU(MSTUP1) = 0.0
      QSLOTL(MSTLP1) = 0.0
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,80)ITER,JTER,EPS,DEL
   80 FORMAT(///T2,'VELOCITY INSIDE SLOTS',T46,
     1'ITER = ',T53,I4,T61,'JTER = ',T68,I4,T76,'EPS = ',
     2T82,E13.6,T99,'DEL = ',T105,E13.6,///T7,'PSI',T66,
     3'PHI')
      WRITE(3,40)(PHI(IP),IP=2,MP1)
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 90
      WRITE(3,85)
   85 FORMAT(///T2,'LOWER SLOT')
      WRITE(3,60)PSI(NSTLP1),(QSLOTL(IP),IP=2,MP1)
   90 WRITE(3,100)
  100 FORMAT(///T2,'UPPER SLOT')
      WRITE(3,60)PSI(NSTUP1),(QSLOTU(IP),IP=2,MP1)
      QSLOTU(MSTUP1) = 1.0
      QSLOTL(MSTLP1) = 1.0
      DO 105 I=2,NP1
      P = QSLOTL(I)
      QSLOTL(I) = DLOG(P)
      P = QSLOTU(I)
  105 QSLOTU(I) = DLOG(P)
C
C           OUTPUT OF R-COORDINATES ( Y-COORDINATES FOR 2-D )
C
 1000 WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,110)ITER,JTER,EPS,DEL
  110 FORMAT(///T1,'0R - COORDINATES  ( UNTRANSFORMED )',
     1T46,'ITER = ',T53,I4,T61,'JTER = ',T68,I4,T76,'EPS = '
```

```
     2,T82,E13.6,T99,'DEL = ',T105,E13.6,///T7,'PHI',T66,
     3'PSI')
      WRITE(3,40)(PSI(IP),IP=2,NP1)
      DO 120 I=2,MP1
  120 WRITE(3,60)PHI(I),(R(I,J),J=2,NP1)
C
C         OUTPUT R COORDINATES INSIDE SLOTS
C
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,130)ITER,JTER,EPS,DEL
  130 FORMAT(///T2,'R - COORDINATES  INSIDE  SLOTS',
     1T46,'ITER = ',T53,I4,T61,'JTER = ',T68,I4,T76,'EPS = '
     2,T82,E13.6,T99,'DEL = ',T105,E13.6,///T7,'PSI',T66,
     3'PHI')
      WRITE(3,40)(PHI(IP),IP=2,MP1)
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 150
      WRITE(3,140)
  140 FORMAT(//T2,'LOWER SLOT')
      WRITE(3,60)PSI(NSTLP1),(RSLOTL(IP),IP=2,MP1)
  150 WRITE(3,160)
  160 FORMAT(//T2,'UPPER SLOT')
      WRITE(3,60)PSI(NSTUP1),(RSLOTU(IP),IP=2,MP1)
      IF (NUT .EQ.1)RETURN
C
C         OUTPUT OF X - COORDINATES
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,170)ITER,JTER,EPS,DEL
  170 FORMAT(///T2,'X - COORDINATES  ( UNTRANSFORMED )',
     1T46,'ITER = ',T53,I4,T61,'JTER = ',T68,I4,T76,'EPS = '
     2,T82,E13.6,T99,'DEL = ',T105,E13.6,///T7,'PHI',T66,
     3'PSI')
      WRITE(3,40)(PSI(IP),IP=2,NP1)
      DO 180 I=2,MP1
  180 WRITE(3,60)PHI(I),(X(I,J),J=2,NP1)
C
C         OUTPUT X VALUES INSIDE SLOTS
C
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,190)ITER,JTER,EPS,DEL
  190 FORMAT(///T2,'X - COORDINATES INSIDE SLOTS',T46,
     1'ITER = ',T53,I4,T61,'JTER = ',T68,I4,T76,'EPS = ',
     2T82,E13.6,T99,'DEL = ',T105,E13.6,///T7,'PSI',T66,
     3'PHI')
      WRITE(3,40)(PHI(IP),IP=2,MP1)
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 200
      WRITE(3,140)
      WRITE(3,60)PSI(NSTLP1),(XSLOTL(IP),IP=2,MP1)
  200 WRITE(3,160)
      WRITE(3,60)PSI(NSTUP1),(XSLOTU(IP),IP=2,MP1)
C
C         OUTPUT VELOCITY VECTOR ANGLES ( ALPHA )
C
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,210)ITER,JTER,EPS,DEL
  210 FORMAT(///T1,'OFLOW ANGLES ( ALPHA )',T46,'ITER = ',
     1T53,I4,T61,'JTER = ',T68,I4,T76,'EPS = ',T82,E13.6,
     2T99,'DEL = ',T105,E13.6,///T7,'PHI',T66,'PSI')
      WRITE(3,40)(PSI(IP),IP=2,NP1)
      DO 220 I=2,MP1
  220 WRITE(3,60)PHI(I),(ALPHA(I,J),J=2,NP1)
```

```
C
C          OUTPUT FLOW ANGLES INSIDE SLOTS
C
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,230)ITER,JTER,EPS,DEL
  230 FORMAT(///T2,'FLOW ANGLES INSIDE SLOTS',T46,'ITER = ',
     1T53,I4,T61,'JTER = ',T68,I4,T76,'EPS = ',T82,E13.6,
     2T99,'DEL = ',T105,E13.6,///T7,'PSI',T66,'PHI')
      WRITE(3,40)(PHI(IP),IP=2,MP1)
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 240
      WRITE(3,140)
      WRITE(3,60)PSI(NSTLP1),(ASLOTL(IP),IP=2,MP1)
  240 WRITE(3,160)
      WRITE(3,60)PSI(NSTUP1),(ASLOTU(IP),IP=2,MP1)
C
C          OUTPUT X AND R - COORDINATES ( TRANSFORMED)
C          IF RMULT > 0
C
      IF (RMULT .LE. 0.0) GO TO 2000
      YMIN=0.999E6
      DO 166 I=2,MP1
      IF(R(I,NP1).LT.YMIN)YMIN=R(I,NP1)
  166 CONTINUE
      RMULT=RMULT/YMIN
      DO 250 J=2,NP1
      DO 250 I=2,MP1
      R(I,J) = RMULT * R(I,J)
  250 X(I,J) = RMULT * X(I,J)
      DO 260 I=2,MP1
      XSLOTL(I) = RMULT * XSLOTL(I)
      XSLOTU(I) = RMULT * XSLOTU(I)
      RSLOTL(I) = RMULT * RSLOTL(I)
  260 RSLOTU(I) = RMULT * RSLOTU(I)
      REWIND 5
      REWIND 12
      WRITE(5)((R(I,J),I=1,MP2),J=1,NP2)
      WRITE (12)((X(I,J),I=1,MP2),J=1,NP2)
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,270)ITER,JTER,EPS,DEL
  270 FORMAT(///T1,'OR - COORDINATES ( TRANSFORMED )',T46,
     1'ITER = ',T53,I4,T61,'JTER = ',T68,I4,T76,'EPS = ',
     2T82,E13.6,T99,'DEL = ',T105,E13.6,///T7,'PHI',T66,
     3'PSI')
      WRITE(3,40)(PSI(IP),IP=2,NP1)
      DO 280 I=2,MP1
  280 WRITE(3,60)PHI(I),(R(I,J),J=2,NP1)
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,290)ITER,JTER,EPS,DEL
  290 FORMAT(///T2,'R - COORDINATES IN SLOT ( TRANSFORMED )'
     1,T46,'ITER = ',T53,I4,T61,'JTER = ',T68,I4,T76,
     2'EPS = ',T82,E13.6,T99,'DEL = ',T105,E13.6,///T7,
     3'PSI',T66,'PHI')
      WRITE(3,40)(PHI(IP),IP=2,MP1)
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 300
      WRITE(3,140)
      WRITE(3,60)PSI(NSTLP1),(RSLOTL(IP),IP=2,MP1)
  300 WRITE(3,160)
      WRITE(3,60)PSI(NSTUP1),(RSLOTU(IP),IP=2,MP1)
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,310)ITER,JTER,EPS,DEL
```

```
  310 FORMAT(///T1,'0X - COORDINATES ( TRANSFORMED )',T46,
     1'ITER = ',T53,I4,T61,'JTER = ',T68,I4,T76,'EPS = ',
     2T82,E13.6,T99,'DEL = ',T105,E13.6,///T7,'PHI',T66,
     3'PSI')
      WRITE(3,40)(PSI(IP),IP=2,NP1)
      DO 320 I=2,MP1
  320 WRITE(3,60)PHI(I),(X(I,J),J=2,NP1)
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,330)ITER,JTER,EPS,DEL
  330 FORMAT(///T2,'X - COORDINATES IN SLOT ( TRANSFORMED )'
     1,T46,'ITER = ',T53,I4,T61,'JTER = ',T68,I4,T76,
     2'EPS = ',T82,E13.6,T99,'DEL = ',T105,E13.6,///T7,
     3'PSI',T66,'PHI')
      WRITE(3,40)(PHI(IP),IP=2,MP1)
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) GO TO 340
      WRITE(3,140)
      WRITE(3,60)PSI(NSTLP1),(XSLOTL(IP),IP=2,MP1)
  340 WRITE(3,160)
      WRITE(3,60)PSI(NSTUP1),(XSLOTU(IP),IP=2,MP1)
C
C         SUMMARY OF WALL STREAMLINE COORDINATES
C
 2000 I = MSTAGU + 1
      J = NSTAGU + 1
      Q(MSTUP1,NSTUP1) = 0.0
      Q(MSTLP1,NSTLP1) = 0.0
      QSLOTU(MSTUP1) = 0.0
      QSLOTL(MSTLP1) = 0.0
      SS = SQRT( (X(I,J)-X(I+1,J))*(X(I,J)-X(I+1,J))
     1         + (R(I,J)-R(I+1,J))*(R(I,J)-R(I+1,J)) )
      SS = SS - S1(I+1)
      SSS = SQRT( (XSLOTU(I)-XSLOTU(I+1))*(XSLOTU(I)-XSLOTU(I+1))
     1           + (RSLOTU(I) -RSLOTU(I+1))*(RSLOTU(I) -RSLOTU(I+1)) )
      SSS = SSS - S2(I+1)
      MSTUP2 = MSTAGU + 2
      DO 345 I=MSTUP2,MP1
      S1(I) = S1(I) + SS
  345 S2(I) = S2(I) + SSS
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      DO 350 I=2,MP1
      DO 350 J=2,NP1
      P = Q(I,J)
  350 Q(I,J) = DEXP(P)
      WRITE(3,360)
  360 FORMAT(///T1,'0SUMMARY OF INLET WALL COORDINATES OF ',
     1T39,'DIFFUSER (UPPER WALL)',///T8,'X',T27,'R',T47,'Q',
     2T68,'S',T85,'ALPHA'/)
      ICOUNT = 1
      DO 370 I=2,MSLUP1
      ICOUNT = ICOUNT + 1
      IF (ICOUNT .LT. 23) GO TO 370
      ICOUNT = 1
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,360)
  370 WRITE(3,375)X(I,NP1),R(I,NP1),Q(I,NP1),S(I),
     1ALPHA(I,NP1)
  375 FORMAT(/1X,E13.6,4(7X,E13.6))
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,380)
  380 FORMAT(///T2,'SUMMARY OF DOWNSTREAM WALL COORDINATES',
```

```
     1T40,' OF DIFFUSER (UPPER WALL)',///T8,'X',T27,'R',T47,           OUT  292
     2'Q',T67,'S',T85,'ALPHA'/)                                         OUT  293
      ICOUNT = 1                                                        OUT  294
      DO 390 I=MSTUP1,MP1                                               OUT  295
      ICOUNT = ICOUNT + 1                                               OUT  296
      IF (ICOUNT .LT. 23) GO TO 390                                     OUT  297
      ICOUNT = 1                                                        OUT  298
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE                            OUT  299
      WRITE(3,380)                                                      OUT  300
  390 WRITE(3,375)X(I,NSTUP1),R(I,NSTUP1),Q(I,NSTUP1),                  OUT  301
     1S1(I),ALPHA(I,NSTUP1)                                             OUT  302
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE                            OUT  303
      WRITE(3,400)                                                      OUT  304
  400 FORMAT(///T2,'SUMMARY OF WALL COORDINATES INSIDE OF ',            OUT  305
     1T40,'DIFFUSER SLOT (UPPER WALL)',///T8,'X',T27,'R',               OUT  306
     2T47,'Q',T67,'S',T85,'ALPHA'/)                                     OUT  307
      ICOUNT = 1                                                        OUT  308
      DO 410 I=MSTUP1,MSLUP1                                            OUT  309
      ICOUNT = ICOUNT + 1                                               OUT  310
      IF (ICOUNT .LT. 23) GO TO 410                                     OUT  311
      ICOUNT = 1                                                        OUT  312
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE                            OUT  313
      WRITE(3,400)                                                      OUT  314
  410 WRITE(3,375)XSLOTU(I),RSLOTU(I),QSLOTU(I),S2(I),                  OUT  315
     1ASLOTU(I)                                                         OUT  316
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) RETURN                        OUT  317
      I = MSTAGL + 1                                                    OUT  318
      J = NSTAGL + 1                                                    OUT  319
      SS = SQRT( (X(I,J)-X(I+1,J))*(X(I,J)-X(I+1,J))                    OUT  320
     1          + (R(I,J)-R(I+1,J))*(R(I,J)-R(I+1,J)) )                 OUT  321
      SS = SS - SBAR1(I+1)                                              OUT  322
      SSS = SQRT( (XSLOTL(I)-XSLOTL(I+1))*(XSLOTL(I)-XSLOTL(I+1))       OUT  323
     1          + (RSLOTL(I)-RSLOTL(I+1))*(RSLOTL(I)-RSLOTL(I+1)) )     OUT  324
      SSS = SSS - SBAR2(I+1)                                            OUT  325
      MSTLP2 = MSTAGL + 2                                               OUT  326
      DO 415 I=MSTLP2,MP1                                               OUT  327
      SBAR1(I) = SBAR1(I) + SS                                          OUT  328
  415 SBAR2(I) = SBAR2(I) + SSS                                         OUT  329
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE                            OUT  330
      WRITE(3,420)                                                      OUT  331
  420 FORMAT(///T2,'SUMMARY OF INLET WALL COORDINATES OF ',             OUT  332
     1T39,'DIFFUSER (LOWER WALL)',///T8,'X',T27,'R',T47,'Q',            OUT  333
     2T67,'S',T85,'ALPHA'/)                                             OUT  334
      ICOUNT = 1                                                        OUT  335
      DO 430 I=2,MSLLP1                                                 OUT  336
      ICOUNT = ICOUNT + 1                                               OUT  337
      IF (ICOUNT .LT. 23) GO TO 430                                     OUT  338
      ICOUNT = 1                                                        OUT  339
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE                            OUT  340
      WRITE(3,420)                                                      OUT  341
  430 WRITE(3,375)X(I,2),R(I,2),Q(I,2),SBAR(I),ALPHA(I,2)               OUT  342
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE                            OUT  343
      WRITE(3,440)                                                      OUT  344
  440 FORMAT(///T2,'SUMMARY OF DOWNSTREAM WALL COORDINATES',            OUT  345
     1T40,' OF DIFFUSER (LOWER WALL)',///T8,'X',T27,'R',T47,'Q'         OUT  346
     2,T67,'S',T85,'ALPHA'/)                                            OUT  347
      ICOUNT = 1                                                        OUT  348
      DO 450 I=MSTLP1,MP1                                               OUT  349
      ICOUNT = ICOUNT + 1                                               OUT  350
      IF (ICOUNT .LT. 23) GO TO 450                                     OUT  351
```

```
      ICOUNT = 1
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,440)
  450 WRITE(3,375)X(I,NSTLP1),R(I,NSTLP1),Q(I,NSTLP1),
     1SBAR1(I),ALPHA(I,NSTLP1)
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,460)
  460 FORMAT(///T2,'SUMMARY OF WALL COORDINATES INSIDE OF ',
     1T40,'DIFFUSER SLOT (LOWER WALL)',///T8,'X',T27,'R',
     2T47,'Q',T67,'S',T85,'ALPHA'/)
      ICOUNT = 1
      DO 470 I=MSTLP1,MSLLP1
      ICOUNT = ICOUNT + 1
      IF (ICOUNT .LT. 23) GO TO 470
      ICOUNT = 1
      WRITE(3,20)HEDR,DATE1,DATE2,DATE3,CASE
      WRITE(3,460)
  470 WRITE(3,375)XSLOTL(I),RSLOTL(I),QSLOTL(I),SBAR2(I),
     1ASLOTL(I)
      RETURN
      END
C         **********     SUBROUTINE  GRAPIC     **********
C
C         THIS SUBROUTINE PROVIDES GRAPHICAL OUTPUT FOR MAIN
C
      SUBROUTINE GRAPIC
      DIMENSION BUFR(500)
      DIMENSION DX(150),ESIMP(150),FSIMP(150),GSIMP(150),
     1         GRAND(150),GIFT(150),PHI(150),PSI(150),
     2         DPSI(150),DPHI(150),S(150),SBAR(150),
     3         S1(150),SBAR1(150),S2(150),SBAR2(150),
     4         HEDR(15),EPHI(150),FPHI(150),GPHI(150),
     5         EPSI(150),ASLOTL(150),XSLOTU(150),XSLOTL(150)
     6         ,FPSI(150),GPSI(150),RSLOTU(150),RSLOTL(150),
     7         QSLOTU(150),QSLOTL(150),ASLOTU(150)
      DIMENSION Q(150,50),R(150,50),X(150,50)
      DIMENSION XX(150),XY(150),LABL(20),RABL(20),WABL(20)
      DOUBLE PRECISION P
      INTEGER FLAG1,FLAG2,FLAG3,FLAG4,CASE,DATE1,DATE2,DATE3
      COMMON / C2 / FLAG1,FLAG2,FLAG3,FLAG4,HEDR,DATE1,DATE2
     1              ,DATE3,CASE,MXIDE,ICKDE,IPRDE,IMIN,IMAX,
     2              JMIN,JMAX,NUT,M,N,TOLDE,DEL,EPS,JTER,
     3              ITER,DPHI,DPSI,S,SBAR,RMULT,S1,S2,SBAR1,
     4              SBAR2,MSTAGU,NSTAGU,MSLOTU,MSTAGL,NSTAGL
     5              ,MSLOTL,RADIN,W,PHI,PSI,RSLOTU,RSLOTL,
     6              QSLOTU,QSLOTL,ASLOTU,ASLOTL,XSLOTU,
     7              XSLOTL,G,MIDJ
      REWIND 4
      REWIND 5
      REWIND 12
      MP1 = M + 1
      NP1 = N + 1
      MP2 = M + 2
      NP2 = N + 2
      MM1 = M - 1
      NM1 = N - 1
      MSTUP2 = MSTAGU + 2
      MSTUP1 = MSTAGU + 1
      MSTUM1 = MSTAGU - 1
      NSTUP1 = NSTAGU + 1
```

```
      NSTUM1 = NSTAGU - 1                                               GRAP  40
      MSTLP2 = MSTAGL + 2                                               GRAP  41
      MSTLP1 = MSTAGL + 1                                               GRAP  42
      MSTLM1 = MSTAGL - 1                                               GRAP  43
      NSTLP1 = NSTAGL + 1                                               GRAP  44
      NSTLM1 = NSTAGL - 1                                               GRAP  45
      MSLUP1 = MSLOTU + 1                                               GRAP  46
      MSLUM1 = MSLOTU - 1                                               GRAP  47
      NSLUP1 = NSLOTU + 1                                               GRAP  48
      NSLUM1 = NSLOTU - 1                                               GRAP  49
      MSLLP1 = MSLOTL + 1                                               GRAP  50
      MSLLM1 = MSLOTL - 1                                               GRAP  51
      NSLLP1 = NSLOTL + 1                                               GRAP  52
      NSLLM1 = NSLOTL - 1                                               GRAP  53
      READ(4)((Q(I,J),I=1,MP2),J=1,NP2)                                 GRAP  54
      READ(5)((R(I,J),I=1,MP2),J=1,NP2)                                 GRAP  55
      READ(12)((X(I,J),I=1,MP2),J=1,NP2)                                GRAP  56
      CALL PLOTS(BUFR,100)                                              GRAP  57
      DATA LABL(1)/'    '/,LABL(2)/'    '/,LABL(3)/'    '/,             GRAP  58
     1     LABL(4)/'    '/                                              GRAP  59
      DATA LABL(5)/' X A'/,LABL(6)/'XIS '/                              GRAP  60
      DATA LABL(7)/'    '/,LABL(8)/'    '/,LABL(9)/'    '/,             GRAP  61
     1     LABL(10)/'    '/                                             GRAP  62
      DATA LABL(11)/'    '/,LABL(12)/'    '/,                           GRAP  63
     1     LABL(13)/'    '/,LABL(14)/'    '/                            GRAP  64
      DATA LABL(15)/'Y-R '/,LABL(16)/'AXIS'/                            GRAP  65
      DATA LABL(17)/'    '/,LABL(18)/'    '/,                           GRAP  66
     1     LABL(19)/'    '/,LABL(20)/'    '/                            GRAP  67
C                                                                       GRAP  68
C         WRITE  HEDR                                                   GRAP  69
C                                                                       GRAP  70
      CALL SYMBOL(0.5,0.0,0.14,HEDR,90.0,60)                            GRAP  71
C                                                                       GRAP  72
C         GRAPH 2-D AND BELL DIFFUSERS                                  GRAP  73
C                                                                       GRAP  74
      IF (FLAG1 .LE. 0 .AND. FLAG3 .LE. 0) GO TO 1000                   GRAP  75
      DO 10 J=1,N                                                       GRAP  76
   10 XX(J) = R(MP1,J+1)                                                GRAP  77
      CALL ADJST(XX,N,1,9.0,0.0,SCAL,XMIN,XY,1)                         GRAP  78
      DO 20 I=2,MP1                                                     GRAP  79
      DO 20 J=2,NP1                                                     GRAP  80
      X(I,J) = X(I,J) / SCAL                                            GRAP  81
   20 R(I,J) = R(I,J) / SCAL                                            GRAP  82
      DO 30 I=MSTUP1,MSLUP1                                             GRAP  83
      XSLOTU(I) = XSLOTU(I) / SCAL                                      GRAP  84
   30 RSLOTU(I) = RSLOTU(I) / SCAL                                      GRAP  85
C                                                                       GRAP  86
C         DRAW AXES FOR DIFFUSER PLOT                                   GRAP  87
C                                                                       GRAP  88
      CALL CGAXES(3.0,0.5,X(MP1,2),9.0,SCAL,SCAL,X(2,2),                GRAP  89
     1R(2,2),LABL,0)                                                    GRAP  90
      DO 40 I=1,M                                                       GRAP  91
      DO 40 J=2,NP1                                                     GRAP  92
      X(I,J) = X(I+1,J)                                                 GRAP  93
   40 R(I,J) = R(I+1,J)                                                 GRAP  94
      DO 50 I=1,M                                                       GRAP  95
      DO 50 J=1,N                                                       GRAP  96
      X(I,J) = X(I,J+1)                                                 GRAP  97
   50 R(I,J) = R(I,J+1)                                                 GRAP  98
C                                                                       GRAP  99
```

```
C          GRAPH  STREAM LINES
C
      DO 70 J=1,NSTAGU
      DO 60 I=1,M
      XX(I) = X(I,J) + 3.0
   60 XY(I) = R(I,J) + 0.5
   70 CALL LINE(XX,XY,M,1,0,0)
      DO 90 J=NSTUP1,N
      DO 80 I=1,MSLOTU
      XX(I) = X(I,J) + 3.0
   80 XY(I) = R(I,J) + 0.5
   90 CALL LINE(XX,XY,MSLOTU,1,0,0)
      IOT = MSLUP1 - MSTAGU
      DO 100 I=1,IOT
      XX(I) = XSLOTU(I+MSTAGU) + 3.0
  100 XY(I) = RSLOTU(I+MSTAGU) + 0.5
      CALL LINE(XX,XY,IOT,1,0,0)
C
C          GRAPH POTENTIAL LINES
C
      DO 120 I=1,MSTAGU
      DO 110 J=1,N
      XX(J) = X(I,J) + 3.0
  110 XY(J) = R(I,J) + 0.5
  120 CALL LINE(XX,XY,N,1,0,0)
      DO 140 I= MSTUP1,M
      DO 130 J=1,NSTAGU
      XX(J) = X(I,J) + 3.0
  130 XY(J) = R(I,J) + 0.5
  140 CALL LINE(XX,XY,NSTAGU,1,0,0)
      JOT = N - NSTUM1
      DO 160 I=MSTUP1,MSLOTU
      DO 150 J=2,JOT
      XX(J) = X(I,J+NSTUM1) + 3.0
  150 XY(J) = R(I,J+NSTUM1) + 0.5
      XX(1) = XSLOTU(I+1) + 3.0
      XY(1) = RSLOTU(I+1) + 0.5
  160 CALL LINE(XX,XY,JOT,1,0,0)
      XLIN = X(M,1) + 9.0
      GO TO 2000
C
C          GRAPH  ANNULAR  DIFFUSER
C
 1000 CONTINUE
      ASH1 = R(MP1,NSTLP1)
      ASH = ASH1
      ASH2 = RSLOTL(MSLLP1)
      IF (ASH2 .LT. ASH1) ASH = ASH2
      IF ( R(2,2) .LT. ASH ) ASH = R(2,2)
      DO 170 J=2,NP1
      DO 170 I=2,MP1
  170 R(I,J) = R(I,J) - ASH
      DO 173 I=MSTUP1,MSLUP1
  173 RSLOTU(I) = RSLOTU(I) - ASH
      DO 176 I=MSTLP1,MSLLP1
  176 RSLOTL(I) = RSLOTL(I) - ASH
      JOT = NSTAGU - NSTAGL + 1
      TRASH1 = R(MP1,NSTUP1)
      TRASH = TRASH1
      TRASH2 = RSLOTU(MSLUP1)
```

```
      IF (TRASH2 .GT. TRASH1) TRASH = TRASH2
      IF ( R(2,NP1) .GT. TRASH ) TRASH = R(2,NP1)
      DO 180 J=1,JOT
  180 XX(J) = R(MP1,J+NSTAGL) + ASH
      XX(1) = ASH
      XX(JOT) = TRASH
      CALL ADJST(XX,JOT,1,9.0,0.5,SCAL,XMIN,XY,1)
      DO 190 I=2,MP1
      DO 190 J=2,NP1
      X(I,J) = X(I,J) / SCAL
  190 R(I,J) = R(I,J) / SCAL + ASH - XMIN
      CALL CGAXES(3.0,0.5,X(MP1,NSTLP1),9.0,SCAL,SCAL,X(2,2),XMIN,
     1LABL,0)
      DO 200 I=MSTLP1,MSLLP1
      XSLOTL(I) = XSLOTL(I) / SCAL
  200 RSLOTL(I) = RSLOTL(I) / SCAL + ASH - XMIN
      DO 210 I=MSTUP1,MSLUP1
      XSLOTU(I) = XSLOTU(I) / SCAL
  210 RSLOTU(I) = RSLOTU(I) / SCAL + ASH - XMIN
      DO 220 I=1,M
      DO 220 J=2,NP1
      X(I,J) = X(I+1,J)
  220 R(I,J) = R(I+1,J)
      DO 230 I=1,M
      DO 230 J=1,N
      X(I,J) = X(I,J+1)
  230 R(I,J) = R(I,J+1)
C
C         GRAPH  STREAMLINES
C
      DO 250 J=1,NSTLM1
      DO 240 I=1,MSLOTL
      XX(I) = X(I,J) + 3.0
  240 XY(I) = R(I,J) + 0.5
  250 CALL LINE(XX,XY,MSLOTL,1,0,0)
      DO 270 J=NSTAGL,NSTAGU
      DO 260 I=1,MP1
      XX(I) = X(I,J) + 3.0
  260 XY(I) = R(I,J) + 0.5
  270 CALL LINE(XX,XY,M,1,0,0)
      DO 290 J=NSTUP1,N
      DO 280 I=1,MSLOTU
      XX(I) = X(I,J) + 3.0
  280 XY(I) = R(I,J) + 0.5
  290 CALL LINE(XX,XY,MSLOTU,1,0,0)
      IOT = MSLOTL - MSTAGL + 1
      DO 300 I=1,IOT
      XX(I) = XSLOTL(I+MSTAGL) + 3.0
  300 XY(I) = RSLOTL(I+MSTAGL) + 0.5
      CALL LINE(XX,XY,IOT,1,0,0)
      IOT = MSLOTU - MSTAGU + 1
      DO 310 I=1,IOT
      XX(I) = XSLOTU(I+MSTAGU) + 3.0
  310 XY(I) = RSLOTU(I+MSTAGU) + 0.5
      CALL LINE(XX,XY,IOT,1,0,0)
C
C         GRAPH  POTENTIAL  LINES
C
      MINN = MSTAGL
      MAXX = MSTAGU
```

```
      IF (MSTAGU .LT. MSTAGL) MAXX = MSTAGL                             GRAP 220
      IF (MSTAGU .LT. MSTAGL) MINN = MSTAGU                             GRAP 221
      DO 330 I=1,MINN                                                   GRAP 222
      DO 320 J=1,N                                                      GRAP 223
      XX(J) = X(I,J) + 3.0                                              GRAP 224
  320 XY(J) = R(I,J) + 0.5                                              GRAP 225
  330 CALL LINE(XX,XY,N,1,0,0)                                          GRAP 226
      MINNP1 = MINN + 1                                                 GRAP 227
      IF (MSTAGL .EQ. MSTAGU) GO TO 415                                 GRAP 228
      DO 410 I=MINNP1,MAXX                                              GRAP 229
      IF (MSTAGL .GT. MSTAGU) GO TO 380                                 GRAP 230
      DO 360 J=1,NSTLM1                                                 GRAP 231
      XX(J) = X(I,J) + 3.0                                              GRAP 232
  360 XY(J) = R(I,J) + 0.5                                              GRAP 233
      XX(NSTAGL) = XSLOTL(I) + 3.0                                      GRAP 234
      XY(NSTAGL) = RSLOTL(I) + 0.5                                      GRAP 235
      CALL LINE(XX,XY,NSTAGL,1,0,0)                                     GRAP 236
      JOT = N - NSTAGL + 1                                              GRAP 237
      DO 370 J=1,JOT                                                    GRAP 238
      XX(J) = X(I,J+NSTLM1) + 3.0                                       GRAP 239
  370 XY(J) = R(I,J+NSTLM1) + 0.5                                       GRAP 240
      CALL LINE(XX,XY,JOT,1,0,0)                                        GRAP 241
      GO TO 410                                                         GRAP 242
  380 DO 390 J=1,NSTAGU                                                 GRAP 243
      XX(J) = X(I,J) + 3.0                                              GRAP 244
  390 XY(J) = R(I,J) + 0.5                                              GRAP 245
      CALL LINE(XX,XY,NSTAGU,1,0,0)                                     GRAP 246
      JOT = N - NSTAGU + 1                                              GRAP 247
      DO 400 J=2,JOT                                                    GRAP 248
      XX(J) = X(I,J+NSTUM1) + 3.0                                       GRAP 249
  400 XY(J) = R(I,J+NSTUM1) + 0.5                                       GRAP 250
      XX(1) = XSLOTU(I) + 3.0                                           GRAP 251
      XY(1) = RSLOTU(I) + 0.5                                           GRAP 252
      CALL LINE(XX,XY,JOT,1,0,0)                                        GRAP 253
  410 CONTINUE                                                          GRAP 254
  415 CONTINUE                                                          GRAP 255
      JOT = NSTAGU - NSTAGL + 1                                         GRAP 256
      DO 430 I=MAXX,M                                                   GRAP 257
      DO 420 J=1,JOT                                                    GRAP 258
      XX(J) = X(I,NSTLM1+J) + 3.0                                       GRAP 259
  420 XY(J) = R(I,NSTLM1+J) + 0.5                                       GRAP 260
  430 CALL LINE(XX,XY,JOT,1,0,0)                                        GRAP 261
      JOT = NSTAGL - 1                                                  GRAP 262
      DO 450 I=MSTLP1,MSLOTL                                            GRAP 263
      DO 440 J=1,JOT                                                    GRAP 264
      XX(J) = X(I,J) + 3.0                                              GRAP 265
  440 XY(J) = R(I,J) + 0.5                                              GRAP 266
      XX(NSTAGL) = XSLOTL(I+1) + 3.0                                    GRAP 267
      XY(NSTAGL) = RSLOTL(I+1) + 0.5                                    GRAP 268
  450 CALL LINE(XX,XY,NSTAGL,1,0,0)                                     GRAP 269
      JOT = N - NSTAGU + 1                                              GRAP 270
      DO 470 I=MSTUP1,MSLOTU                                            GRAP 271
      DO 460 J=2,JOT                                                    GRAP 272
      XX(J) = X(I,J+NSTUM1) + 3.0                                       GRAP 273
  460 XY(J) = R(I,J+NSTUM1) + 0.5                                       GRAP 274
      XX(1) = XSLOTU(I+1) + 3.0                                         GRAP 275
      XY(1) = RSLOTU(I+1) + 0.5                                         GRAP 276
  470 CALL LINE(XX,XY,JOT,1,0,0)                                        GRAP 277
      XLIN = X(M,NSTAGL) + 9.0                                          GRAP 278
C                                                                       GRAP 279
```

```
C          GRAPH  ANNULAR  DIFFUSER  IN  PERSPECTIVE
C
      DO 480 J=1,N
      DO 480 I=1,M
      R(I,J) = SCAL * R(I,J) + ASH
  480 X(I,J) = SCAL * X(I,J)
      JOT = NSTAGU - NSTAGL + 1
      DO 490 J=1,JOT
  490 XX(J) = R(M,NSTAGL+J)
      XX(1) = 0.0
      XX(JOT) = TRASH + ASH
      CALL ADJST(XX,JOT,1,9.0,0.5,ZCAL,XMIN,XY,1)
      DO 500 I=1,M
      DO 500 J=1,N
      X(I,J) = X(I,J) / ZCAL
  500 R(I,J) = R(I,J) / ZCAL
      DO 510 I=MSTLP1,MSLLP1
      XSLOTL(I) = ( XSLOTL(I) * SCAL ) / ZCAL
  510 RSLOTL(I) = ( RSLOTL(I) * SCAL + ASH ) / ZCAL
      DO 520 I=MSTUP1,MSLUP1
      XSLOTU(I) = ( XSLOTU(I) * SCAL ) / ZCAL
  520 RSLOTU(I) = ( RSLOTU(I) * SCAL + ASH ) / ZCAL
      CALL CGAXES(XLIN,0.5,X(M,NSTAGL),9.0,ZCAL,ZCAL,0.0,0.0,LABL,0)
C
C          GRAPH  STREAMLINES
C
      DO 530 I=1,MSLOTL
      XX(I) = X(I,1) + XLIN
  530 XY(I) = R(I,1) + 0.5
      CALL LINE(XX,XY,MSLOTL,1,0,0)
      IOT = M - MSTAGL +1
      DO 540 I=1,IOT
      XX(I) = X(I+MSTLM1,NSTAGL) + XLIN
  540 XY(I) = R(I+MSTLM1,NSTAGL) + 0.5
      CALL LINE(XX,XY,IOT,1,0,0)
      IOT = MSLOTL - MSTAGL + 1
      DO 550 I=1,IOT
      XX(I) = XSLOTL(I+MSTAGL) + XLIN
  550 XY(I) = RSLOTL(I+MSTAGL) + 0.5
      CALL LINE(XX,XY,IOT,1,0,0)
      DO 560 I=1,MSLOTU
      XX(I) = X(I,N) + XLIN
  560 XY(I) = R(I,N) + 0.5
      CALL LINE(XX,XY,MSLOTU,1,0,0)
      IOT = M - MSTAGU + 1
      DO 570 I=1,IOT
      XX(I) = X(I+MSTUM1,NSTAGU) + XLIN
  570 XY(I) = R(I+MSTUM1,NSTAGU) + 0.5
      CALL LINE(XX,XY,IOT,1,0,0)
      IOT = MSLOTU - MSTAGU + 1
      DO 580 I=1,IOT
      XX(I) = XSLOTU(I+MSTAGU) + XLIN
  580 XY(I) = RSLOTU(I+MSTAGU) + 0.5
      CALL LINE(XX,XY,IOT,1,0,0)
C
C          GRAPH  POTENTIAL  LINES
C
      DO 590 J=1,N
      XX(J) = X(1,J) + XLIN
  590 XY(J) = R(1,J) + 0.5
```

```
      CALL LINE(XX,XY,N,1,0,0)
      JOT = NSTAGU - NSTAGL + 1
      DO 600 J=1,JOT
      XX(J) = X(M,J+NSTLM1) + XLIN
  600 XY(J) = R(M,J+NSTLM1) + 0.5
      CALL LINE(XX,XY,JOT,1,0,0)
      DO 610 I=1,M
      DO 610 J=1,N
      X(I,J) = ( X(I,J) * ZCAL ) / SCAL
  610 R(I,J) = ( R(I,J) * ZCAL - ASH ) / SCAL
      XLIN = XLIN + X(M,NSTAGL) + 9.0
C
C         GRAPH  VELOCITY PROFILES FOR DIFFUSER WITH
C         NEUMANN BOUNDARY CONDITIONS
C
 2000 IF (FLAG2 .LE. 0) GO TO 3000
      DATA WABL(1)/'    '/,WABL(2)/'    '/,WABL(3)/'    '/,
     1     WABL(4)/'    '/
      DATA WABL(5)/' Q A'/,WABL(6)/'XIS '/
      DATA WABL(7)/'    '/,WABL(8)/'    '/,WABL(9)/'    '/,
     1     WABL(10)/'    '/
      DATA WABL(11)/'    '/,WABL(12)/'    '/,
     1     WABL(13)/'    '/,WABL(14)/'    '/
      DATA WABL(15)/' N A'/,WABL(16)/'XIS '/
      DATA WABL(17)/'    '/,WABL(18)/'    '/,
     1     WABL(19)/'    '/,WABL(20)/'    '/
      DO 615 J=2,NP1
      DO 615 I=2,MP1
      P = Q(I,J)
  615 Q(I,J) = DEXP(P)
      XX(NP1) = 0.0
      DO 620 J=1,N
  620 XX(J) = Q(2,J+1)
      CALL ADJST(XX,NP1,1,9.0,0.5,SSCAL,XMIN,XY,1)
      CALL CGAXES(XLIN,0.5,12.0,9.0,SSCAL,SCAL,XMIN,0.0,WABL,0)
      DO 630 J=1,N
      XX(J) = ( Q(2,J+1) - XMIN ) / SSCAL + XLIN
  630 XY(J) = R(1,J) + 0.5
      CALL LINE(XX,XY,N,1,0,0)
      IF (FLAG1 .GT. 0 .OR. FLAG3 .GT. 0) NSTAGL = 1
      JOT = NSTAGU - NSTAGL + 1
      DO 640 J=1,JOT
      XX(J) = ( Q(MP1,J+NSTAGL) - XMIN ) / SSCAL + XLIN
  640 XY(J) = R(M,J+NSTAGL-1) + 0.5
      CALL LINE(XX,XY,JOT,1,0,0)
C
C         MOVE PEN TO FINISHED POSITION
C
 3000 XSET = XLIN + 15.0
      CALL PLOT1(XSET,0.0,-3)
      WRITE(3,650)
  650 FORMAT(1H1,T2,'GRAPHICAL OUTPUT COMPLETED PROGRAM ',
     1T37,'SIGNING OFF')
      RETURN
      END
```

APPENDIX B - DIFFUSER DESIGN

This appendix describes the procedure used in designing the 18-inch (46 cm) annular diffuser. One should realize that the final design contour is rarely if ever the result of a first trial program. Several reruns are nearly always necessary to achieve the desired geometry because such parameters as AR and L/W are not input directly even though specific values of these may be the ultimate goal.

For the 18-inch (46 cm) annular diffuser, the aim was to design a diffuser with the following features: (1) an area ratio of about 3:1 (2) an inlet width of about 2 inches (5.08 cm) (3) an exit outer diameter of about 18-inches (46 cm) (4) an effective length for boundary layer growth upstream of the slot of about 3 inches (7.6 cm). Items (1) and (4) were the same as the previously designed 9-inch (23 cm) diffuser, while items (2) and (3) were twice as large.

The design program was used to design both the accelerating nozzle (inlet section) and the diffuser itself as one flow channel. Wall velocity vector magnitude ratio (Q) increasing from 1.0 to 3.0 in the nozzle portion and from 3.0 to 3.5 from the narrowest portion of the channel to the slot were prescribed. By prescribing a favorable velocity gradient upstream of the slot, it was thought that slot widths ranging from the design width to several times the design width could be tested. (As the slot is widened, there is a tendency for the wall velocity just upstream of the slot to decrease, thus producing a possibly unfavorable velocity gradient.) Downstream of the slot, Q along the wall was prescribed to be constant at 0.83. The Q ratio of 3.0 at diffuser inlet to 0.83 at exit is larger than the desired AR because suction removes some of the flow and because the diffuser inlet and exit profiles were not precisely uniform. The value 0.83 was arrived at after several trial runs. (A good first trial for Q is $\frac{(1 - FS)\, Q_i}{AR}$ which is 0.95 in this case.) To get the appropriate values of ϕ for the desired velocities, recognize that

$$\phi_2 - \phi_1 = \int_{S_1}^{S_2} Q \, dS$$

After adopting a model such as Figure 21 corresponding values of Q and ϕ can be found.

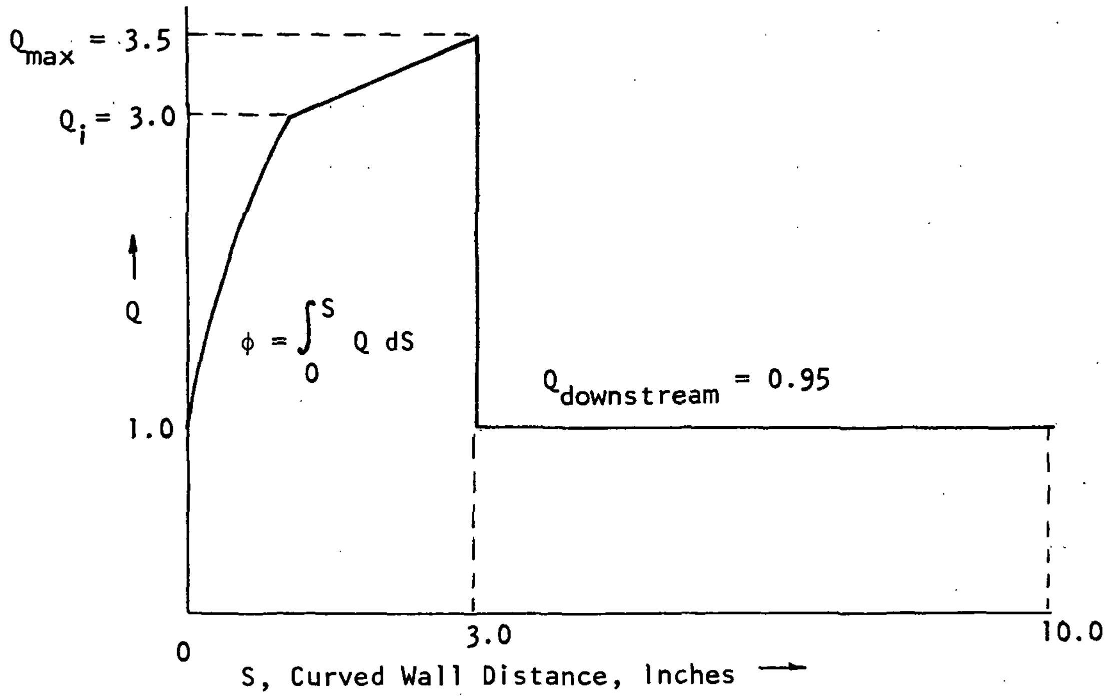


Figure 21. Prescribed wall velocity as design input.

The proper range of ψ can be determined by applying $R_2^{\,2} - R_1^{\,2} = 2 \int_{\phi} \frac{\cos \alpha}{Q} \, d\psi$ at either the inlet or the exit passage. At inlet, a 7-inch (17.8 cm) outer radius and a 5-inch (12.7 cm) inner radius were desirable since these dimensions would enable the first three items of the previously listed characteristics to be attainable. Thus

$$\Psi \cong \frac{3(7^2 - 5^2)}{2} = 36$$

was a good first trial value. (For the final design, Ψ was adjusted to 33.5.) Increments $\Delta\psi$ were selected to be much smaller at each wall, and increments $\Delta\phi$ were selected smaller in the vicinity of the slot. This enabled better resolution to be obtained in the troublesome slot region and a minimizing of the error introduced by the wedge flow patch at the stagnation point. The program requires that $\Delta\phi = \Delta\psi$ in the grid network immediately around the stagnation point, and these were arbitrarily given a value of 0.05 for this program. The upper slot contained 7 streamlines (including wall and stagnation point streamlines) and 4 potential flow lines. The lower slot contained 6 streamlines and 4 potential flow lines. The number of streamlines contained in each slot and the exact value of each $\Delta\psi$ in the vicinity of the slot must be adjusted so that the flow area represented by these $\Delta\psi$'s is sufficient to pass the design suction flow for that slot. If, for example, 2% suction is desired through the inner slot and 4% through the outer slot (these values were close to but not precisely the design suction rates), then

$$\underset{\text{across inner slot}}{\Sigma\,\Delta\psi} = .02 \underset{\text{total}}{\Sigma\,\Delta\psi} = .02\Psi$$

and

$$\underset{\text{across outer slot}}{\Sigma\,\Delta\psi} = .04 \underset{\text{total}}{\Sigma\,\Delta\psi} = .04\Psi$$

For the previously named first trial Ψ of 36, this would yield 0.72 as the value of ψ for the lower slot stagnation streamline and 36.0 - 1.44 or 34.56 for the upper slot stagnation streamline. The final design had 0.65 as the lower slot stagnation streamline, 32.4 for the upper slot stagnation streamline, and 33.5 for Ψ. This is equivalent to 0.65/33.5 or 1.94% lower slot suction and (33.5 - 32.4)/33.5 or 3.28% upper slot suction and 5.22% total suction. The design suction percentage was determined by assuming the boundary layer growth to be the same as that along a flat plate, and the velocity profile to obey the 1/7 power law, and applying Taylor's criterion to determine how much fluid must be removed.

The total number of potential flow lines, M, was arbitrarily selected as 127 (a maximum of 148 may be used), while 45 total streamlines (N) were used (maximum 48). MIDJ, the number of the "center streamline," was 21. MIDJ and the $\Delta\psi$ values in the vicinity of this streamline may have to be adjusted from the first trial values to prevent the design from becoming extremely asymmetrical, possibly crossing the X-axis. This situation will result in a divide check, an overflow, or an underflow, with subsequent program termination.

Dirichlet boundary conditions were used at the flow channel inlet (not to be confused with the diffuser inlet) with a constant velocity magnitude ratio, Q, (not parallel) across the channel of 1.0 being specified. At the flow channel exit (which was also the diffuser exit), Neumann boundary conditions were specified to yield parallel (not necessarily uniform) flow across the exit plane.

The next two pages show the input data which was used to design the 18-inch (46 cm) annular diffuser. Following the input data is the first page of output which summarizes the control input data. Then the graphical output (figure 22) is shown reduced in size from the original output. Also shown is a listing of inner and outer wall radii.

THE FOLLOWING DATA WAS INPUT TO DESIGN THE 18-INCHANNULAR DIFFUSER

```
2 1  BIG ANNULAR DIFFUSER DESIGN                                        070171   0011
     0.00       1.70 .00005000  .0000500          84          34
     127          45        10        50          39           6
   3.9100         21        11        51          87          87
      0.0        0.1       0.2       0.3         0.4         0.5
      0.6        0.7       0.8       0.9         1.0         1.1
      1.2        1.3       1.4       1.5         1.6         1.7
      1.8        1.9       2.0       2.1         2.2         2.3
      2.4        2.5       2.6       2.7         2.8         2.9
      3.0        3.1       3.2       3.3         3.4         3.5
      3.6        3.7       3.8       3.9         4.0         4.1
      4.2        4.3       4.4       4.5         4.6         4.7
      4.8        4.9       5.0       5.1         5.2         5.3
      5.4        5.5       5.6       5.7         5.8         5.9
      6.0        6.1       6.2       6.3         6.4         6.5
      6.6        6.7       6.8       6.9         7.0         7.1
      7.2        7.3       7.4       7.5         7.6         7.7
      7.8        7.9       8.0       8.1        8.15        8.20
     8.25       8.30      8.35       8.4        8.45         8.5
      8.6        8.7       8.8       8.9        9.05         9.2
     9.35       9.50      9.65      9.80        9.95       10.10
    10.25      10.40     10.55     10.70       10.85       11.00
    11.15      11.30     11.45     11.60       11.75       11.90
    12.05      12.20     12.35     12.50       12.65       12.80
    12.95      13.10     13.25     13.40       13.55       13.70
    13.85
  1.000      1.28412   1.49755   1.65381    1.79784     1.91427
  2.02216    2.10813   2.18835   2.26306    2.33251     2.39693
  2.44696    2.50280   2.54602   2.58637    2.62400     2.65903
  2.69126    2.72190   2.75002   2.77611    2.80031     2.81842
  2.83958    2.85926   2.87403   2.89137    2.90763     2.91993
  2.93457    2.94576   2.95923   2.96967    2.98243     2.99249
  3.00500    3.01500   3.02750   3.03750    3.05000     3.06000
  3.07000    3.08000   3.09247   3.10250    3.11250     3.12500
  3.13500    3.14500   3.15500   3.16750    3.17748     3.18750
  3.19748    3.20747   3.21748   3.22998    3.23998     3.24998
  3.25998    3.26998   3.27998   3.28998    3.29998     3.30998
  3.31998    3.32998   3.33998   3.34998    3.35998     3.36998
  3.37998    3.38998   3.39998   3.40998    3.41998     3.42997
  3.43997    3.44747   3.45747   3.46747    3.47247     3.47747
  3.48247    3.48747   3.49247
    0.830      0.830     0.830     0.830      0.830       0.830
    0.830      0.830     0.830     0.830      0.830       0.830
    0.830      0.830     0.830     0.830      0.830       0.830
    0.830      0.830     0.830     0.830      0.830       0.830
    0.830      0.830     0.830     0.830      0.830       0.830
    0.830      0.830     0.830     0.830      0.830       0.830
    0.830      0.830     0.830     0.830      0.830       0.830
    0.830
    0.830      0.830     0.830
  1.00000    1.28412   1.49755   1.65381    1.79784     1.91427
  2.02216    2.10813   2.18835   2.26306    2.33251     2.39693
  2.44696    2.50280   2.54602   2.58637    2.62400     2.65903
  2.69126    2.72190   2.75002   2.77611    2.80031     2.81842
  2.83958    2.85926   2.87403   2.89137    2.90763     2.91993
  2.93457    2.94576   2.95923   2.96967    2.98243     2.99249
```

```
  3.00499   3.01499   3.02749   3.03749   3.04999   3.05999
  3.06999   3.07999   3.09247   3.10249   3.11249   3.12499
  3.13499   3.14499   3.15499   3.16749   3.17748   3.18749
  3.19748   3.20747   3.21748   3.22998   3.23998   3.24998
  3.25998   3.26998   3.27998   3.28998   3.29998   3.30998
  3.31998   3.32998   3.33998   3.34998   3.35998   3.36998
  3.37998   3.38998   3.39998   3.40998   3.41998   3.42997
  3.43997   3.44747   3.45747   3.46747   3.47247   3.47747
  3.48247   3.48747   3.49247
    0.830     0.830     0.830     0.830     0.830     0.830
    0.830     0.830     0.830     0.830     0.830     0.830
    0.830     0.830     0.830     0.830     0.830     0.830
    0.830     0.830     0.830     0.830     0.830     0.830
    0.830     0.830     0.830     0.830     0.830     0.830
    0.830     0.830     0.830     0.830     0.830     0.830
    0.830     0.830     0.830     0.830     0.830     0.830
    0.830
    0.830     0.830     0.830
   0.0000    0.2000    0.3750    0.5250    0.6000    0.6500
   0.7000    0.9750    1.5750    2.5750    3.8250    5.3050
   7.0550    8.5550    9.5550   10.5550   11.5550   12.5550
  13.5550   14.5550   15.1200   16.0000   17.0000   18.0000
  19.0000   20.0000   21.0000   22.0000   23.0000   24.0000
  25.0000   26.0000   27.0000   28.5000   29.7500   31.2500
  32.2500   32.3500   32.4000   32.4500   32.5500   32.7000
  32.9000   33.2000   33.5000
      1.0       1.0       1.0       1.0       1.0       1.0
      1.0       1.0       1.0       1.0       1.0       1.0
      1.0       1.0       1.0       1.0       1.0       1.0
      1.0       1.0       1.0       1.0       1.0       1.0
      1.0       1.0       1.0       1.0       1.0       1.0
      1.0       1.0       1.0       1.0       1.0       1.0
      1.0       1.0       1.0       1.0       1.0       1.0
      1.0       1.0       1.0
```

```
                                        CLEMSON  UNIVERSITY

                                  MECHANICAL  ENGINEERING  DEPARTMENT

                                          PROGRAM  70-02

                          AXIALLY  SYMMETRIC  AND  2-D  BRANCHED  CHANNEL  DESIGN

*****     BIG ANNULAR DIFFUSER DESIGN                                     7/ 1/71    CASE  NO.     11      *****



                                       ------  CONTROL  DATA  ------



    FLAG1  =0        RMULT  =       0.0               M       =         127          RADIN    =    3.91000
    FLAG2  =2        W      =       1.700000          N       =          45          MIDJ     =         21
    FLAG3  =0        TOLSYS =       0.000050          MXISYS  =          10          IPRSYS   =         11
    FLAG4  =1        TOLDE  =       0.000050          MXIDE   =          50          IPRDE    =         51
                     MSTAGU =             84          NSTAGU  =          39          MSLOTU   =         87
                     MSTAGL =             84          NSTAGL  =           6          MSLOTL   =         87


                                          ------  OPTIONS  ------



                                        AXIALLY  SYMMETRIC

                                        DIRICHLET B. C. AT INLET AND NEUMANN B. C. AT OUTLET
                                           WITH VELOCITY INPUT AT INLET

                                        ANNULAR  DIFFUSER - NOZZLE

                                        GRAPHICAL  OUTPUT
```

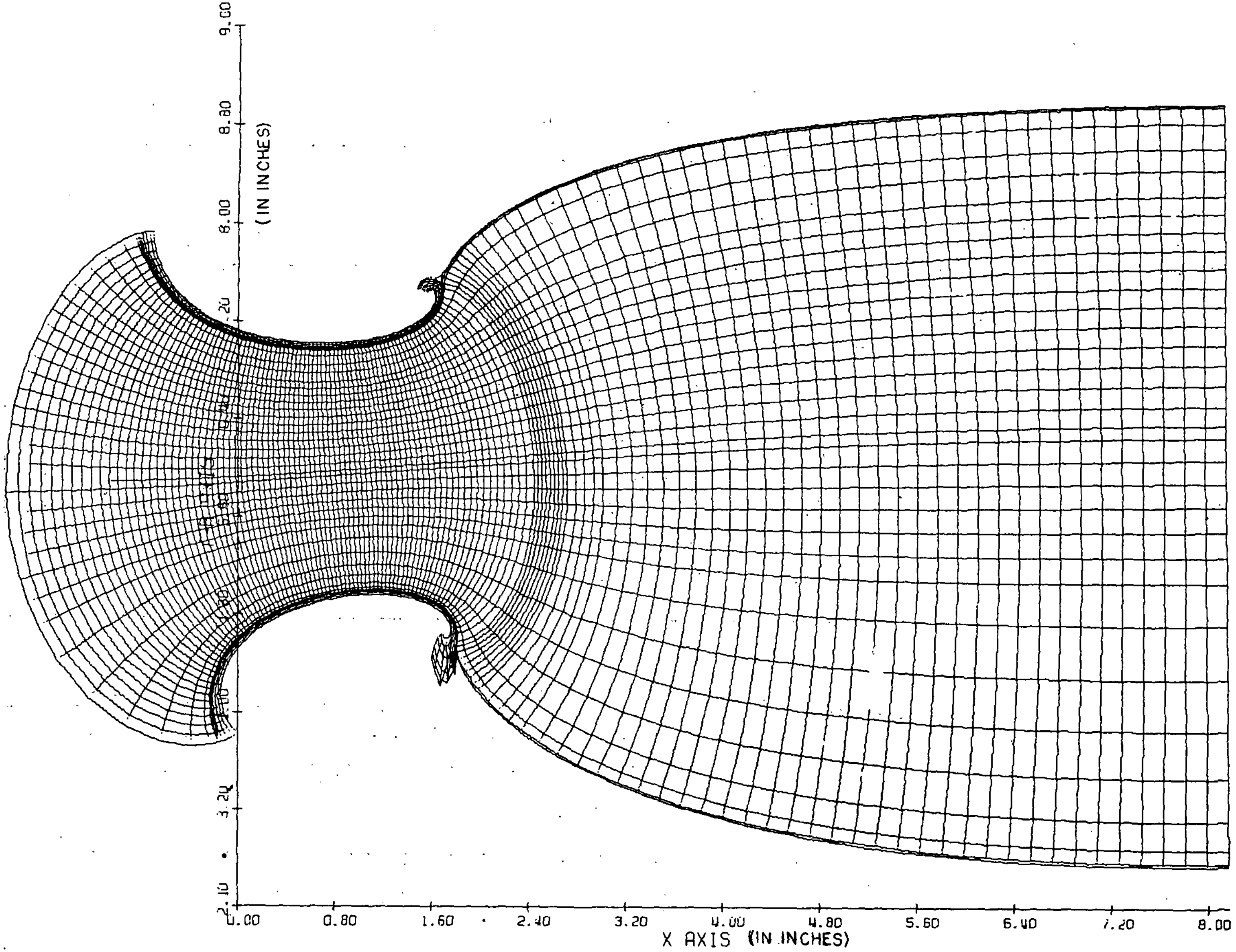


Figure 22. Graphical output showing streamlines and equipotential lines and wall coordinates for 18-inch (46 cm) annular diffuser.

```
SUMMARY OF INLET WALL COORDINATES OF DIFFUSER (UPPER WALL)

       X                  R                   Q                   S                  ALPHA

-0.699307E 00       0.801371E 01       0.100000E 01       0.0                 -0.157854E 01
-0.697799E 00       0.793128E 01       0.128412E 01       0.880181E-01        -0.147606E 01
-0.685813E 00       0.785777E 01       0.149755E 01       0.159424E 00        -0.137102E 01
-0.668520E 00       0.779451E 01       0.165381E 01       0.222923E 00        -0.127491E 01
-0.647939E 00       0.773845E 01       0.179784E 01       0.280845E 00        -0.118974E 01
-0.624546E 00       0.768814E 01       0.191427E 01       0.334726E 00        -0.111102E 01
-0.599122E 00       0.764241E 01       0.202216E 01       0.385522E 00        -0.103905E 01
-0.571913E 00       0.760064E 01       0.210813E 01       0.433943E 00        -0.973163E 00
-0.543409E 00       0.756214E 01       0.218835E 01       0.480485E 00        -0.914747E 00
-0.513902E 00       0.752656E 01       0.226306E 01       0.525411E 00        -0.860717E 00
-0.483603E 00       0.749362E 01       0.233251E 01       0.568925E 00        -0.809925E 00
-0.452665E 00       0.746310E 01       0.239693E 01       0.611196E 00        -0.760975E 00
-0.421096E 00       0.743480E 01       0.244696E 01       0.652465E 00        -0.716066E 00
-0.389138E 00       0.740842E 01       0.250280E 01       0.692856E 00        -0.673961E 00
-0.356773E 00       0.738393E 01       0.254602E 01       0.732453E 00        -0.633235E 00
-0.324080E 00       0.736108E 01       0.258637E 01       0.771419E 00        -0.595931E 00
-0.291140E 00       0.733977E 01       0.262400E 01       0.809801E 00        -0.560900E 00
-0.257984E 00       0.731986E 01       0.265903E 01       0.847655E 00        -0.527809E 00
-0.224671E 00       0.730128E 01       0.269126E 01       0.885033E 00        -0.496518E 00
-0.191245E 00       0.728393E 01       0.272190E 01       0.921978E 00        -0.466752E 00
-0.157726E 00       0.726773E 01       0.275001E 01       0.958525E 00        -0.438289E 00
-0.124130E 00       0.725264E 01       0.277611E 01       0.994715E 00        -0.411107E 00
-0.904713E-01       0.723856E 01       0.280030E 01       0.103058E 01        -0.384790E 00
-0.567531E-01       0.722548E 01       0.281841E 01       0.106618E 01        -0.360296E 00
-0.230089E-01       0.721326E 01       0.283957E 01       0.110153E 01        -0.337472E 00
 0.107403E-01       0.720190E 01       0.285925E 01       0.113662E 01        -0.314906E 00
 0.445117E-01       0.719135E 01       0.287403E 01       0.117149E 01        -0.293746E 00
 0.782807E-01       0.718154E 01       0.289136E 01       0.120618E 01        -0.273918E 00
 0.112004E 00       0.717245E 01       0.290763E 01       0.124067E 01        -0.254252E 00
 0.145722E 00       0.716407E 01       0.291992E 01       0.127499E 01        -0.235669E 00
 0.179415E 00       0.715632E 01       0.293456E 01       0.130915E 01        -0.217955E 00
 0.213065E 00       0.714920E 01       0.294576E 01       0.134317E 01        -0.200973E 00
 0.246683E 00       0.714266E 01       0.295922E 01       0.137704E 01        -0.184706E 00
 0.280251E 00       0.713669E 01       0.296967E 01       0.141077E 01        -0.169009E 00
 0.313775E 00       0.713125E 01       0.298243E 01       0.144438E 01        -0.153909E 00
 0.347243E 00       0.712633E 01       0.299249E 01       0.147785E 01        -0.139307E 00
 0.380622E 00       0.712190E 01       0.300499E 01       0.151120E 01        -0.125108E 00
 0.413945E 00       0.711797E 01       0.301498E 01       0.154443E 01        -0.111278E 00
 0.447179E 00       0.711449E 01       0.302749E 01       0.157753E 01        -0.977013E-01
```

```
0.480333E 00    0.711148E 01    0.303748E 01    0.161050E 01    -0.843255E-01
0.513400E 00    0.710892E 01    0.304999E 01    0.164336E 01    -0.710913E-01
0.546371E 00    0.710681E 01    0.305999E 01    0.167609E 01    -0.577618E-01
0.579260E 00    0.710513E 01    0.306998E 01    0.170872E 01    -0.448380E-01
0.612046E 00    0.710388E 01    0.307998E 01    0.174124E 01    -0.322427E-01
0.644735E 00    0.710303E 01    0.309246E 01    0.177364E 01    -0.194190E-01
0.677315E 00    0.710262E 01    0.310248E 01    0.180593E 01    -0.629646E-02
0.709790E 00    0.710264E 01    0.311249E 01    0.183811E 01     0.647169E-02
0.742153E 00    0.710306E 01    0.312498E 01    0.187017E 01     0.196210E-01
0.774394E 00    0.710392E 01    0.313499E 01    0.190211E 01     0.332333E-01
0.806537E 00    0.710521E 01    0.314498E 01    0.193396E 01     0.467528E-01
0.838545E 00    0.710693E 01    0.315499E 01    0.196571E 01     0.603038E-01
0.870418E 00    0.710907E 01    0.316748E 01    0.199734E 01     0.744864E-01
0.902163E 00    0.711168E 01    0.317747E 01    0.202886E 01     0.893180E-01
0.933763E 00    0.711475E 01    0.318748E 01    0.206028E 01     0.104379E 00
0.965232E 00    0.711829E 01    0.319747E 01    0.209160E 01     0.119833E 00
0.996552E 00    0.712232E 01    0.320746E 01    0.212282E 01     0.135771E 00
0.102771E 01    0.712683E 01    0.321748E 01    0.215395E 01     0.152148E 00
0.105871E 01    0.713184E 01    0.322997E 01    0.218497E 01     0.169524E 00
0.108951E 01    0.713740E 01    0.323997E 01    0.221588E 01     0.188011E 00
0.112012E 01    0.714351E 01    0.324997E 01    0.224670E 01     0.207168E 00
0.115053E 01    0.715021E 01    0.325998E 01    0.227742E 01     0.227265E 00
0.118073E 01    0.715751E 01    0.326997E 01    0.230804E 01     0.248460E 00
0.121068E 01    0.716545E 01    0.327997E 01    0.233858E 01     0.270872E 00
0.124036E 01    0.717405E 01    0.328997E 01    0.236902E 01     0.294692E 00
0.126977E 01    0.718335E 01    0.329997E 01    0.239937E 01     0.320056E 00
0.129887E 01    0.719339E 01    0.330997E 01    0.242962E 01     0.347209E 00
0.132758E 01    0.720421E 01    0.331997E 01    0.245979E 01     0.376464E 00
0.135591E 01    0.721587E 01    0.332997E 01    0.248986E 01     0.408066E 00
0.138377E 01    0.722842E 01    0.333997E 01    0.251985E 01     0.442398E 00
0.141111E 01    0.724191E 01    0.334997E 01    0.254974E 01     0.479924E 00
0.143785E 01    0.725643E 01    0.335997E 01    0.257955E 01     0.521236E 00
0.146388E 01    0.727204E 01    0.336998E 01    0.260926E 01     0.567117E 00
0.148907E 01    0.728884E 01    0.337997E 01    0.263889E 01     0.618589E 00
0.151325E 01    0.730691E 01    0.338997E 01    0.266844E 01     0.677078E 00
0.153619E 01    0.732637E 01    0.339997E 01    0.269789E 01     0.744553E 00
0.155760E 01    0.734732E 01    0.340997E 01    0.272726E 01     0.823958E 00
0.157704E 01    0.736986E 01    0.341998E 01    0.275654E 01     0.919744E 00
0.159392E 01    0.739412E 01    0.342996E 01    0.278574E 01     0.103900E 01
0.160736E 01    0.742018E 01    0.343996E 01    0.281485E 01     0.119361E 01
0.161615E 01    0.744842E 01    0.344746E 01    0.284389E 01     0.140277E 01
0.161876E 01    0.748096E 01    0.345747E 01    0.287285E 01     0.169840E 01
0.161507E 01    0.752623E 01    0.346746E 01    0.290173E 01     0.213280E 01
0.160530E 01    0.753676E 01    0.347246E 01    0.291614E 01     0.250805E 01
```

```
    0.159233E 01      0.754261E 01      0.347746E 01      0.293053E 01      0.294370E 01
    0.157811E 01      0.754202E 01      0.348246E 01      0.294490E 01      0.341566E 01
    0.156573E 01      0.753508E 01      0.348746E 01      0.295924E 01      0.389990E 01
    0.155818E 01      0.752339E 01      0.349246E 01      0.297357E 01      0.437780E 01

   SUMMARY OF DOWNSTREAM WALL COORDINATES OF DIFFUSER (UPPER WALL)

          X                 R                 Q                 S                ALPHA

    0.167012E 01      0.765809E 01      0.100000E 01      0.0               0.285718E 01
    0.170109E 01      0.770311E 01      0.830000E 00      0.546426E-01      0.111739E 01
    0.173193E 01      0.775789E 01      0.830000E 00      0.114883E 00      0.101841E 01
    0.176704E 01      0.780935E 01      0.830000E 00      0.175124E 00      0.937877E 00
    0.180540E 01      0.785774E 01      0.830000E 00      0.235365E 00      0.871287E 00
    0.184630E 01      0.790338E 01      0.830000E 00      0.295606E 00      0.814893E 00
    0.188910E 01      0.794649E 01      0.830000E 00      0.355847E 00      0.766357E 00
    0.197973E 01      0.802632E 01      0.830000E 00      0.476328E 00      0.685222E 00
    0.207568E 01      0.809920E 01      0.830000E 00      0.596810E 00      0.619281E 00
    0.217554E 01      0.816603E 01      0.830000E 00      0.717291E 00      0.554065E 00
    0.227840E 01      0.822757E 01      0.830000E 00      0.837773E 00      0.516799E 00
    0.243722E 01      0.831117E 01      0.830000E 00      0.101850E 01      0.456696E 00
    0.260025E 01      0.838592E 01      0.830000E 00      0.119922E 01      0.406356E 00
    0.276643E 01      0.845301E 01      0.830000E 00      0.137994E 01      0.363449E 00
    0.293509E 01      0.851342E 01      0.830000E 00      0.156066E 01      0.326374E 00
    0.310570E 01      0.856796E 01      0.830000E 00      0.174138E 01      0.293998E 00
    0.327788E 01      0.861730E 01      0.830000E 00      0.192210E 01      0.265491E 00
    0.345135E 01      0.866203E 01      0.830000E 00      0.210283E 01      0.240230E 00
    0.362587E 01      0.870262E 01      0.830000E 00      0.228355E 01      0.217724E 00
    0.380126E 01      0.873950E 01      0.830000E 00      0.246427E 01      0.197582E 00
    0.397739E 01      0.877304E 01      0.830000E 00      0.264499E 01      0.179487E 00
    0.415413E 01      0.880357E 01      0.830000E 00      0.282571E 01      0.163181E 00
    0.433140E 01      0.883136E 01      0.830000E 00      0.300644E 01      0.148448E 00
    0.450910E 01      0.885669E 01      0.830000E 00      0.318716E 01      0.135097E 00
    0.468719E 01      0.887976E 01      0.830000E 00      0.336788E 01      0.122973E 00
    0.486561E 01      0.890077E 01      0.830000E 00      0.354860E 01      0.111936E 00
    0.504430E 01      0.891992E 01      0.830000E 00      0.372932E 01      0.101870E 00
    0.522323E 01      0.893736E 01      0.830000E 00      0.391004E 01      0.926709E-01
    0.540238E 01      0.895322E 01      0.830000E 00      0.409076E 01      0.842395E-01
    0.558170E 01      0.896764E 01      0.830000E 00      0.427149E 01      0.764982E-01
    0.576119E 01      0.898073E 01      0.830000E 00      0.445221E 01      0.693699E-01
    0.594080E 01      0.899261E 01      0.830000E 00      0.463293E 01      0.627882E-01
    0.612055E 01      0.900334E 01      0.830000E 00      0.481365E 01      0.566900E-01
```

```
   0.630040E 01       0.901302E 01       0.830000E 00       0.499438E 01       0.510197E-01
   0.648033E 01       0.902172E 01       0.830000E 00       0.517510E 01       0.457353E-01
   0.666034E 01       0.902951E 01       0.830000E 00       0.535582E 01       0.407826E-01
   0.684046E 01       0.903642E 01       0.830000E 00       0.553654E 01       0.360934E-01
   0.702063E 01       0.904253E 01       0.830000E 00       0.571726E 01       0.316626E-01
   0.720085E 01       0.904785E 01       0.830000E 00       0.589798E 01       0.274436E-01
   0.738112E 01       0.905242E 01       0.830000E 00       0.607871E 01       0.234045E-01
   0.756142E 01       0.905630E 01       0.830000E 00       0.625943E 01       0.195101E-01
   0.774174E 01       0.905946E 01       0.830000E 00       0.644015E 01       0.157398E-01
   0.792213E 01       0.906198E 01       0.830000E 00       0.662087E 01       0.120344E-01
   0.810067E 01       0.906380E 01       0.830000E 00       0.680159E 01       0.974582E-02


 SUMMARY OF WALL COORDINATES INSIDE OF DIFFUSER SLOT (UPPER WALL)

         X                  R                  Q                  S                ALPHA

   0.167012E 01       0.765809E 01       0.0                0.0                0.285718E 01
   0.155353E 01       0.763134E 01       0.830000E 00       0.119616E 00       0.341848E 01
   0.150178E 01       0.760217E 01       0.830000E 00       0.179856E 00       0.390385E 01
   0.146956E 01       0.754968E 01       0.830000E 00       0.240097E 00       0.444185E 01


 SUMMARY OF INLET WALL COORDINATES OF DIFFUSER (LOWER WALL)

         X                  R                  Q                  S                ALPHA

   0.0                0.391000E 01       0.100000E 01       0.0                0.224947E 01
  -0.522614E-01       0.396167E 01       0.128412E 01       0.880181E-01       0.211946E 01
  -0.888348E-01       0.401695E 01       0.149755E 01       0.159424E 00       0.198959E 01
  -0.113682E 00       0.406892E 01       0.165381E 01       0.222923E 00       0.186774E 01
  -0.130419E 00       0.411868E 01       0.179784E 01       0.280845E 00       0.175798E 01
  -0.140355E 00       0.416647E 01       0.191427E 01       0.334726E 00       0.165572E 01
  -0.144936E 00       0.421239E 01       0.202216E 01       0.385522E 00       0.156114E 01
  -0.144800E 00       0.425654E 01       0.210813E 01       0.433943E 00       0.147362E 01
  -0.140841E 00       0.429900E 01       0.218835E 01       0.480485E 00       0.139454E 01
  -0.133610E 00       0.433975E 01       0.226306E 01       0.525411E 00       0.132079E 01
  -0.123518E 00       0.437879E 01       0.233251E 01       0.568925E 00       0.125124E 01
  -0.110930E 00       0.441609E 01       0.239693E 01       0.611196E 00       0.118445E 01
  -0.960225E-01       0.445176E 01       0.244696E 01       0.652465E 00       0.112266E 01
  -0.792452E-01       0.448588E 01       0.250280E 01       0.692856E 00       0.106459E 01
  -0.606857E-01       0.451840E 01       0.254602E 01       0.732453E 00       0.100879E 01
```

```
-0.405791E-01      0.454947E 01      0.258637E 01      0.771419E 00      0.957263E 00
-0.191262E-01      0.457912E 01      0.262400E 01      0.809801E 00      0.908799E 00
 0.352591E-02      0.460741E 01      0.265903E 01      0.847655E 00      0.863020E 00
 0.272374E-01      0.463437E 01      0.269126E 01      0.885033E 00      0.819772E 00
 0.518910E-01      0.466007E 01      0.272190E 01      0.921978E 00      0.778762E 00
 0.773776E-01      0.468454E 01      0.275001E 01      0.958525E 00      0.739717E 00
 0.103615E 00      0.470784E 01      0.277611E 01      0.994715E 00      0.702566E 00
 0.130536E 00      0.473001E 01      0.280030E 01      0.103058E 01      0.666868E 00
 0.158077E 00      0.475108E 01      0.281841E 01      0.106618E 01      0.633534E 00
 0.186144E 00      0.477117E 01      0.283957E 01      0.110153E 01      0.602375E 00
 0.214680E 00      0.479026E 01      0.285925E 01      0.113662E 01      0.571968E 00
 0.243651E 00      0.480840E 01      0.287403E 01      0.117149E 01      0.543395E 00
 0.273003E 00      0.482567E 01      0.289136E 01      0.120618E 01      0.516572E 00
 0.302662E 00      0.484207E 01      0.290763E 01      0.124067E 01      0.490305E 00
 0.332642E 00      0.485764E 01      0.291992E 01      0.127499E 01      0.465504E 00
 0.362884E 00      0.487243E 01      0.293456E 01      0.130915E 01      0.441922E 00
 0.393351E 00      0.488645E 01      0.294576E 01      0.134317E 01      0.419361E 00
 0.424039E 00      0.489977E 01      0.295922E 01      0.137704E 01      0.397831E 00
 0.454915E 00      0.491239E 01      0.296967E 01      0.141077E 01      0.377158E 00
 0.485929E 00      0.492434E 01      0.298243E 01      0.144438E 01      0.357297E 00
 0.517078E 00      0.493564E 01      0.299249E 01      0.147785E 01      0.338157E 00
 0.548330E 00      0.494633E 01      0.300499E 01      0.151120E 01      0.319648E 00
 0.579686E 00      0.495640E 01      0.301498E 01      0.154443E 01      0.301689E 00
 0.611099E 00      0.496589E 01      0.302749E 01      0.157753E 01      0.284139E 00
 0.642576E 00      0.497480E 01      0.303748E 01      0.161050E 01      0.266941E 00
 0.674103E 00      0.498315E 01      0.304999E 01      0.164336E 01      0.250018E 00
 0.705672E 00      0.499093E 01      0.305999E 01      0.167609E 01      0.233135E 00
 0.737271E 00      0.499817E 01      0.306998E 01      0.170872E 01      0.216764E 00
 0.768873E 00      0.500487E 01      0.307998E 01      0.174124E 01      0.200804E 00
 0.800478E 00      0.501105E 01      0.309246E 01      0.177364E 01      0.184669E 00
 0.832075E 00      0.501669E 01      0.310248E 01      0.180593E 01      0.168287E 00
 0.863653E 00      0.502180E 01      0.311249E 01      0.183811E 01      0.152290E 00
 0.895206E 00      0.502639E 01      0.312498E 01      0.187017E 01      0.135944E 00
 0.926719E 00      0.503043E 01      0.313499E 01      0.190211E 01      0.119151E 00
 0.958192E 00      0.503394E 01      0.314498E 01      0.193396E 01      0.102403E 00
 0.989616E 00      0.503690E 01      0.315499E 01      0.196571E 01      0.856250E-01
 0.102098E 01      0.503933E 01      0.316748E 01      0.199734E 01      0.682125E-01
 0.105227E 01      0.504119E 01      0.317747E 01      0.202886E 01      0.500863E-01
 0.108348E 01      0.504247E 01      0.318748E 01      0.206028E 01      0.316566E-01
 0.111461E 01      0.504317E 01      0.319747E 01      0.209160E 01      0.127478E-01
 0.114564E 01      0.504327E 01      0.320746E 01      0.212282E 01     -0.674397E-02
 0.117657E 01      0.504276E 01      0.321748E 01      0.215395E 01     -0.267645E-01
 0.120737E 01      0.504162E 01      0.322997E 01      0.218497E 01     -0.479249E-01
```

```
0.123800E 01       0.503981E 01       0.323997E 01       0.221588E 01       -0.703674E-01
0.126848E 01       0.503731E 01       0.324997E 01       0.224670E 01       -0.936477E-01
0.129879E 01       0.503410E 01       0.325998E 01       0.227742E 01       -0.118046E 00
0.132890E 01       0.503015E 01       0.326997E 01       0.230804E 01       -0.143745E 00
0.135878E 01       0.502542E 01       0.327997E 01       0.233858E 01       -0.170900E 00
0.138838E 01       0.501987E 01       0.328997E 01       0.236902E 01       -0.199717E 00
0.141771E 01       0.501347E 01       0.329997E 01       0.239937E 01       -0.230362E 00
0.144670E 01       0.500618E 01       0.330997E 01       0.242962E 01       -0.263106E 00
0.147530E 01       0.499794E 01       0.331997E 01       0.245979E 01       -0.298234E 00
0.150346E 01       0.498870E 01       0.332997E 01       0.248986E 01       -0.336049E 00
0.153111E 01       0.497839E 01       0.333997E 01       0.251985E 01       -0.376996E 00
0.155817E 01       0.496697E 01       0.334997E 01       0.254974E 01       -0.421537E 00
0.158454E 01       0.495435E 01       0.335997E 01       0.257955E 01       -0.470326E 00
0.161008E 01       0.494046E 01       0.336998E 01       0.260926E 01       -0.524174E 00
0.163467E 01       0.492521E 01       0.337997E 01       0.263889E 01       -0.584081E 00
0.165811E 01       0.490850E 01       0.338997E 01       0.266844E 01       -0.651456E 00
0.168017E 01       0.489023E 01       0.339997E 01       0.269789E 01       -0.728252E 00
0.170051E 01       0.487028E 01       0.340997E 01       0.272726E 01       -0.817324E 00
0.171874E 01       0.484854E 01       0.341998E 01       0.275654E 01       -0.922945E 00
0.173429E 01       0.482486E 01       0.342996E 01       0.278574E 01       -0.105212E 01
0.174630E 01       0.479917E 01       0.343996E 01       0.281485E 01       -0.121754E 01
0.175348E 01       0.477157E 01       0.344746E 01       0.284389E 01       -0.144259E 01
0.175356E 01       0.474217E 01       0.345747E 01       0.287285E 01       -0.177140E 01
0.174206E 01       0.470084E 01       0.346746E 01       0.290173E 01       -0.223506E 01
0.173120E 01       0.469145E 01       0.347246E 01       0.291614E 01       -0.262596E 01
0.171761E 01       0.468730E 01       0.347746E 01       0.293053E 01       -0.308083E 01
0.170361E 01       0.469006E 01       0.348246E 01       0.294490E 01       -0.359190E 01
0.169320E 01       0.469947E 01       0.348746E 01       0.295924E 01       -0.418724E 01
0.169154E 01       0.471257E 01       0.349246E 01       0.297357E 01       -0.498664E 01
0.175667E 01       0.458115E 01       0.100000E 01       0.0                -0.218438E 01
0.181034E 01       0.451745E 01       0.830000E 00       0.832936E-01       -0.119029E 01
0.183357E 01       0.446427E 01       0.830000E 00       0.143534E 00       -0.112576E 01
0.186179E 01       0.441100E 01       0.830000E 00       0.203775E 00       -0.105386E 01
0.189374E 01       0.435939E 01       0.830000E 00       0.264016E 00       -0.991186E 00
0.192869E 01       0.430960E 01       0.830000E 00       0.324257E 00       -0.937046E 00
0.196589E 01       0.426172E 01       0.830000E 00       0.384498E 00       -0.889833E 00
0.204624E 01       0.417120E 01       0.830000E 00       0.504979E 00       -0.810058E 00
0.213314E 01       0.408652E 01       0.830000E 00       0.625461E 00       -0.744371E 00
0.222510E 01       0.400710E 01       0.830000E 00       0.745942E 00       -0.688474E 00
0.232113E 01       0.393247E 01       0.830000E 00       0.866424E 00       -0.639811E 00
0.247151E 01       0.382874E 01       0.830000E 00       0.104715E 01       -0.576649E 00
0.262797E 01       0.373373E 01       0.830000E 00       0.122787E 01       -0.522485E 00
0.278921E 01       0.364658E 01       0.830000E 00       0.140859E 01       -0.475141E 00
0.295431E 01       0.356659E 01       0.830000E 00       0.158931E 01       -0.433169E 00
```

```
   0.312255E 01        0.349313E 01        0.830000E 00        0.177003E 01        -0.395560E 00
   0.329335E 01        0.342566E 01        0.830000E 00        0.195075E 01        -0.361601E 00
   0.346626E 01        0.336371E 01        0.830000E 00        0.213148E 01        -0.330746E 00
   0.364092E 01        0.330685E 01        0.830000E 00        0.231220E 01        -0.302564E 00
   0.381699E 01        0.325471E 01        0.830000E 00        0.249292E 01        -0.276731E 00
   0.399425E 01        0.320693E 01        0.830000E 00        0.267364E 01        -0.252966E 00


  SUMMARY OF DOWNSTREAM WALL COORDINATES OF DIFFUSER (LOWER WALL)

         X                   R                   Q                   S                 ALPHA

   0.417248E 01        0.316320E 01        0.830000E 00        0.285436E 01        -0.231034E 00
   0.435152E 01        0.312323E 01        0.830000E 00        0.303509E 01        -0.210757E 00
   0.453121E 01        0.308675E 01        0.830000E 00        0.321581E 01        -0.191969E 00
   0.471141E 01        0.305353E 01        0.830000E 00        0.339653E 01        -0.174548E 00
   0.489205E 01        0.302335E 01        0.830000E 00        0.357725E 01        -0.158349E 00
   0.507306E 01        0.299599E 01        0.830000E 00        0.375797E 01        -0.143253E 00
   0.525433E 01        0.297127E 01        0.830000E 00        0.393869E 01        -0.129187E 00
   0.543581E 01        0.294901E 01        0.830000E 00        0.411942E 01        -0.116060E 00
   0.561747E 01        0.292906E 01        0.830000E 00        0.430014E 01        -0.103781E 00
   0.579930E 01        0.291127E 01        0.830000E 00        0.448086E 01        -0.922359E-01
   0.598122E 01        0.289551E 01        0.830000E 00        0.466158E 01        -0.814096E-01
   0.616319E 01        0.288166E 01        0.830000E 00        0.484231E 01        -0.712363E-01
   0.634525E 01        0.286960E 01        0.830000E 00        0.502303E 01        -0.616213E-01
   0.652733E 01        0.285924E 01        0.830000E 00        0.520375E 01        -0.525319E-01
   0.670947E 01        0.285049E 01        0.830000E 00        0.538447E 01        -0.438809E-01
   0.689171E 01        0.284327E 01        0.830000E 00        0.556519E 01        -0.356087E-01
   0.707392E 01        0.283755E 01        0.830000E 00        0.574591E 01        -0.277206E-01
   0.725615E 01        0.283321E 01        0.830000E 00        0.592663E 01        -0.201385E-01
   0.743837E 01        0.283023E 01        0.830000E 00        0.610736E 01        -0.128521E-01
   0.762058E 01        0.282853E 01        0.830000E 00        0.628808E 01        -0.581392E-02
   0.780279E 01        0.282812E 01        0.830000E 00        0.646880E 01         0.996035E-03
   0.798498E 01        0.282889E 01        0.830000E 00        0.664952E 01         0.752290E-02
   0.816144E 01        0.283079E 01        0.830000E 00        0.683024E 01         0.974635E-02


  SUMMARY OF WALL COORDINATES INSIDE OF DIFFUSER SLOT (LOWER WALL)

         X                   R                   Q                   S                 ALPHA

   0.175667E 01        0.458115E 01        0.0                 0.0                 -0.218438E 01
   0.166674E 01        0.430126E 01        0.830000E 00        0.293985E 00        -0.354095E 01
   0.161743E 01        0.442754E 01        0.830000E 00        0.354225E 00        -0.397032E 01
   0.159443E 01        0.453362E 01        0.830000E 00        0.414466E 00        -0.395741E 01
```

APPENDIX C. DEVELOPMENT OF PERFORMANCE EQUATIONS

The following assumptions were made in the derivation of the performance equations:

(1) Tests were conducted under steady state steady flow conditions.

(2) The temperature (and thus the internal energy) and the pressure were uniform across the inlet section and across the exit section.

(3) The fluid density was constant. Since the maximum velocities encountered resulted in a Mach number of about 0.25, the resultant error should be negligible.

Define the effectiveness ε as follows:

$$\varepsilon = \frac{\dot{m}_e \, (P_{s,e} - P_{s,i})/\rho}{\dot{m}_i \, (KE_i - KE_e)} \tag{27}$$

where $\dot{m}$ represents the mass flow rate and KE is the kinetic energy per unit mass flowing.

$$\dot{m}_i \, KE_i = \int_{A_i} \frac{U_i^2}{2} \, (U_i \, \rho) \, dA_i \tag{28}$$

$$\dot{m}_i \, KE_e = \left(\frac{\dot{m}_i}{\dot{m}_e}\right) \dot{m}_e \, KE_e = \left(\frac{\dot{m}_i}{\dot{m}_e}\right) \int_{A_e} \frac{U_e^2}{2} \, (U_e \, \rho) \, dA_e \tag{29}$$

$$\varepsilon = \frac{\dot{m}_e \, (P_{s,e} - P_{s,i})/\rho}{\int_{A_i} \frac{U_i^2}{2} \, (U_i \, \rho) \, dA_i - \left(\frac{\dot{m}_i}{\dot{m}_e}\right) \int_{A_e} \frac{U_e^2}{2} \, (U_e \, \rho) \, dA_e} \tag{30}$$

Now define a kinetic energy coefficient κ:

$$\kappa_i = \frac{\int_{A_i} \frac{U_i^2}{2} (\rho\, U_i)\, dA_i}{\frac{\rho}{2}\, \overline{U}_i^{\,3}\, A_i} \quad \text{and} \quad \kappa_e = \frac{\int_{A_e} \frac{U_e^2}{2} (\rho\, U_e)\, dA_e}{\frac{\rho}{2}\, \overline{U}_e^{\,3}\, A_e} \tag{31}$$

so that the effectiveness expression becomes

$$\varepsilon = \frac{\dot{m}_e\, (P_{s,e} - P_{s,i})}{\kappa_i\, \frac{\rho}{2}\, \overline{U}_i^{\,3}\, A_i - \kappa_e\, \frac{\dot{m}_i}{\dot{m}_e}\, \frac{\rho}{2}\, \overline{U}_e^{\,3}\, A_e} = \frac{\dot{m}_e\, (P_{s,e} - P_{s,i})}{(\kappa_i\, \dot{m}_i\, \overline{U}_i^{\,2} - \kappa_e\, \dot{m}_i\, \overline{U}_e^{\,2})/2} \tag{32}$$

$$\varepsilon = \frac{\dot{m}_e\, (P_{s,e} - P_{s,i})}{\frac{\dot{m}_i}{2} \left(\kappa_i\, \overline{U}_i^{\,2} - \kappa_e\, \overline{U}_e^{\,2}\right)} = \frac{\dot{m}_e\, (P_{s,e} - P_{s,i})}{\frac{\dot{m}_i\, \overline{U}_i^{\,2}}{2} \left[\kappa_i - \kappa_e \left(\frac{\overline{U}_e}{\overline{U}_i}\right)^2\right]} \tag{33}$$

But $\left(\frac{\overline{U}_e}{\overline{U}_i}\right)^2 = \left(\frac{1.0 - FS}{AR}\right)^2$ (34)

and $\frac{\dot{m}_e}{\dot{m}_i} = (1.0 - FS)$ (35)

so

$$\varepsilon = \frac{(1.0 - FS)(P_{s,e} - P_{s,i})}{P_{d,i} \left[\kappa_i - \kappa_e \left(\frac{1.0 - FS}{AR}\right)^2\right]} \tag{25}$$

This was the expression used to compute the effectiveness ε. The more conventional expression for the effectiveness is similar to equation 25 except that the ratio of mass flow rates is deleted.

Thus

$$\eta = \frac{(P_{s,e} - P_{s,i})}{P_{d,i} \left[\kappa_i - \kappa_e \left(\frac{1.0 - FS}{AR}\right)^2\right]} \tag{24}$$

Considering the total pressure loss,

$$P_{t,i} - P_{t,e} = (P_{s,i} + \rho\ KE_i) - (P_{s,e} + \rho\ KE_e) \qquad (36)$$

$$= \frac{\rho}{2} (\kappa_i\ \overline{U}_i^{\,2} - \kappa_e\ \overline{U}_e^{\,2}) - (P_{s,e} - P_{s,i})$$

$$= \frac{\rho}{2}\ \overline{U}_i^{\,2} \left[\kappa_i - \kappa_e\left(\frac{\overline{U}_e}{\overline{U}_i}\right)^2\right] - \eta\ \frac{\rho\overline{U}_i^{\,2}}{2} \left[\kappa_i - \kappa_e \left(\frac{1.0 - FS}{AR}\right)^2\right]$$

$$P_{t,i} - P_{t,e} = (1-\eta)\ P_{d,i} \left[\kappa_i - \kappa_e \left(\frac{1.0 - FS}{AR}\right)^2\right] \qquad (26)$$

Values of κ_i (and κ_e) were computed from measured velocities as follows. Consider n velocities U_j at n radii r_j incremented by a constant interval (where j = 1, 2, ..., n). If we let the velocity U_j be representative of the annular area between radii

$$\frac{r_{j+1} + r_j}{2} \quad \text{and} \quad \frac{r_j + r_{j-1}}{2}\ \text{, then}$$

$$\kappa = \frac{1}{A} \int_A \left(\frac{U}{\overline{U}}\right)^3 dA \cong \frac{1}{A} \sum_{j=1}^{n} \left(\frac{U_j}{\overline{U}}\right)^3 \frac{\pi}{4}\, [r^2_{j+1} - r^2_{j-1} + 2r_j(r_{j+1} - r_{j-1})] \qquad (37)$$

where

$$\overline{U} \cong \frac{1}{A} \sum_{j=1}^{n} U_j\ [r^2_{j+1} - r^2_{j-1} + 2r_j\ (r_{j+1} - r_{j-1})] \qquad (38)$$

In equation (38) the radii r_o and r_{n+1} are equal to the inner and outer diffuser wall radii respectively. All other radii, r_1 to r_n, are readily obtained since the increment between radii is a constant. Using equation (37), typical values obtained were 1.025 for K_i and 1.011 for K_e.