(4)

AD-A201 353

CONTRACT NO. N00014-85-C-0050

A NEW PUMPJET DESIGN THEORY

HONEYWELL INC.
600 SECOND STREET N.E.
HOPKINS, MN 55343

964-8-TP
101388

88 10 21 001

Contract No. N00014-85-C-0050
Final Report

4

A NEW PUMPJET DESIGN THEORY

HONEYWELL INC.
600 SECOND STREET N.E.
HOPKINS, MINNESOTA 55343

by
OKITSUGU FURUYA
WEN-LI CHIANG

JUNE 19[illegible]

Prepared for:

DAVID TAYLOR
RESEARCH CENTER
BETHESDA, MARYLAND 20084

OFFICE OF NAVAL RESEARCH
800 NORTH QUINCY STREET
ARLINGTON, VIRGINIA 22217-5000

Approved for public release;
distribution unlimited

TABLE OF CONTENTS

TABLE OF CONTENTS (CON'T)

A NEW PUMPJET DESIGN THEORY

Final Technical Report
01/01/87 – 12/31/88

O. Furuya
W.L. Chiang

N00014-85-C-0050

DTRC
Dept. of the Navy
Bethesda, MD 20084

June 30, 1988

Office of Naval Research
800 North Qunicy St.
Arlington, VA 22217-5000

Unclassified

Sponsored by the Naval Sea Systems Command General Hydrodynamic Research Program and administered by the David Taylor Research Center, Code 1505, Bethesda, MD 20084.

Pumpjet
Streamline Curvature Method
Blade-through Flow
Blade-to-Blade Method

The pumpjet is a unique fluid machine which utilizes retarded wake flow and produces high propulsive efficiency such as 90%. The existing pumpjet design method is based on a simple two-dimensional graphic method which was used for pump design. As the demand for the speed of underwater vehicles increased in recent years, the existing design method became inappropriate. Effort has been made to develop a new quasi-three-dimensional pump design method by combining a blade-through flow theory with blade-to-blade flow theory.

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES

LIST OF TABLES

1.0 BACKGROUND

The pumpjet is considered to be one of the most promising candidate propulsors for high speed underwater vehicles and, as a matter of fact, it has recently been employed for MK 48 torpedoes, ALWT Advanced Light Weight Torpedo, now called MK 50 and other underwater vehicles. The pumpjet superiority over other propulsion devices is represented by two major factors, i.e., high efficiency and quietness.

The pumpjet is one of few fluid devices which positively utilizes retarded wake flow and produces high propulsive efficiency. This peculiar situation may be understood readily by considering the momentum equation applied to a control volume surrounding an underwater vehicle, fixed to the inertial coordinate system. In the conventional propeller, for example, the velocity of flow coming into a propeller blade is approximately equal to the vehicle speed since the propeller diameter is large enough to enjoy the free stream flow. In order for the propeller to generate any effective thrust, it should accelerate the flow, the ejected flow speed being faster than the incoming flow. If one observes this situation from the inertial frame, the ejected flow has a finite positive flow speed against the surrounding environment. It means that certain amount of the energy imparted on the fluid by the thruster is dumped in the surrounding water. On the other hand, the pumpjet receives the retarded flow velocity, slower than the free stream velocity. In order to generate a thrust, again this flow should be accelerated. However, if the pumpjet is properly designed, the accelerated flow velocity can nearly be that of the vehicle speed. If one looks at a similar control volume, from the inertial frame, the ejected flow out of the pumpjet has almost no absolute velocity and thus leaves hardly any jet wake after the vehicle passed. There exists much less wasted energy in the flow field after a vehicle with a pumpjet passes. This is the major reason why the pumpjet can produce such high propulsive efficiency such as 90% or higher if it is properly designed.

Quietness is a guaranteed aspect with the pumpjet, as can be seen from its configuration (see Figure 1-1); a long shroud completely surrounding the rotor helps prevent rotor noise from emitting into the outside flow field. Furthermore, this "internal" flow machine has better resistance characteristics against cavitation, resulting in quieter shallow water operation where propulsors are most susceptible to cavitation.

However, in order to achieve such a high standard of performance there are many penalties to be paid in reality. The first such penalty naturally stems from the pumpjet's utilizing the velocity-retarded wake flow. A typical meridional flow distribution at the inlet of pumpjet rotor is shown in Figure 1-2; the velocity at the hub is only 30% of the free stream velocity and rapidly increases to 75% at the shroud internal boundary. This large velocity gradient in the transverse direction is, of course, built up by the viscous boundary layer effect and is one of the key features causing difficulties in design, fabrication and eventually in achieving the pumpjet high performance.

When one designs an axial or a near axial pump, it is customary to distribute the blade loading from hub to tip in a forced vortex or a free vortex distribution method, such as shown in Figure 1-3. Such distribution methods are important in obtaining as uniform a discharge jet behind the rotor as possible to minimize the mixing loss. However, a serious problem arises in attempting to implement either forced vortex or free vortex loading distribution against the flow field having a large velocity gradient, as shown in Figure 1-2. Due to the lack of enough meridional flow velocity near the hub, the blade there should be designed to have extremely large incidence angle as well as large camber. It is for this reason that the pumpjet rotor designed to date has a distorted profile shape from hub to tip, see Figure 1-4. If this were a conventional propeller, the stagger angle would become smaller towards the hub and the camber would stay more or less constant. However, for the reason mentioned above, the pumpjet blade stagger angle first becomes smaller up to

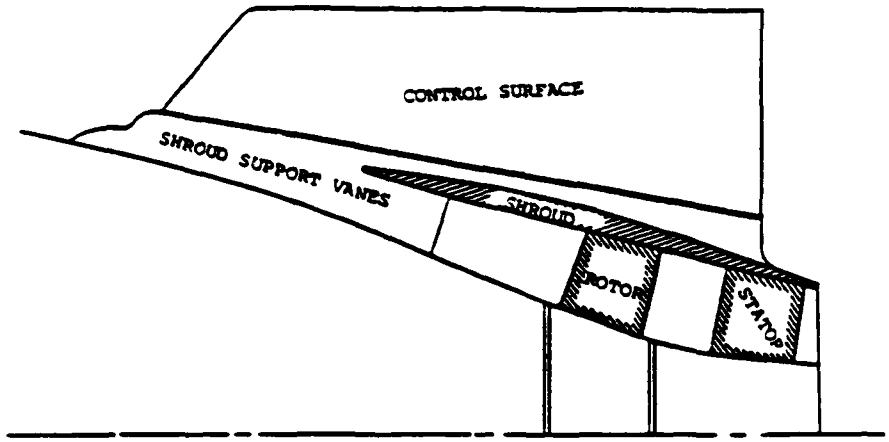


FIGURE 1-1. A TYPICAL PUMPJET BLADE AND SHROUD CONFIGURATION

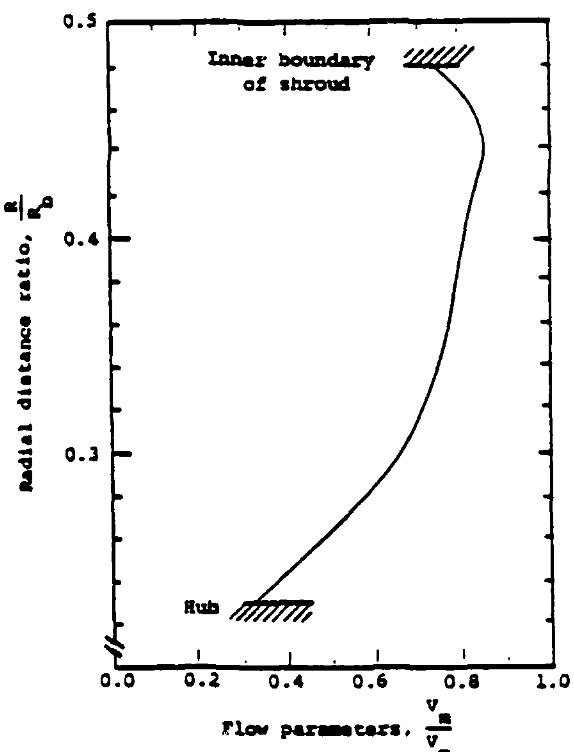


FIGURE 1-2. A TYPICAL MERIDIONAL FLOW VELOCITY (V_m) DISTRIBUTION FOR A PUMPJET WHERE V_∞ IS UPSTREAM FLOW VELOCITY

964-8-6
080888

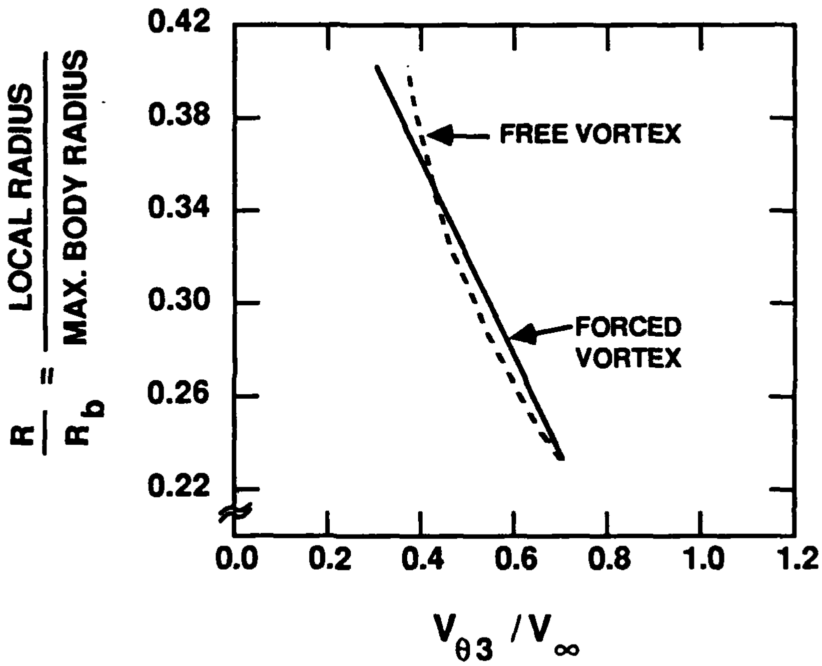
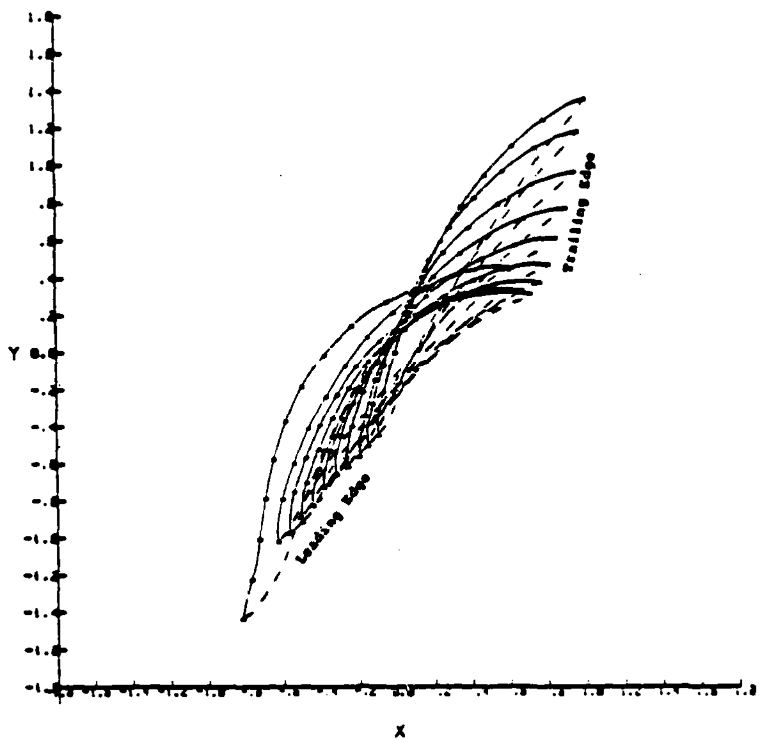
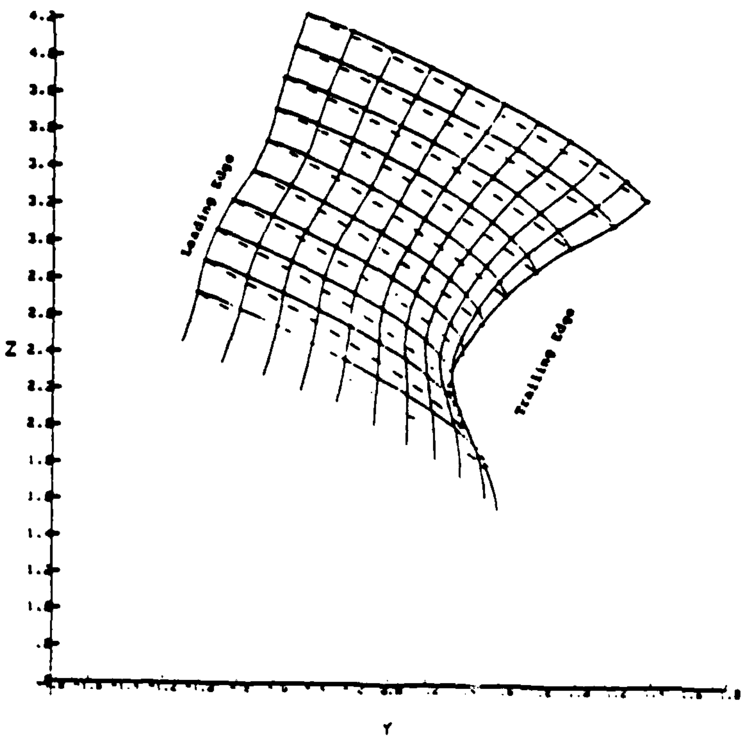


FIGURE 1-3. A TYPICAL LOAD DISTRIBUTION IN TERMS OF $V_{\theta 3}$ FOR PUMPJET ROTOR BLADE WHERE $V_{\theta 3}$ IS CIRCUMFERENTIAL COMPONENT OF THE TURNED FLOW VELOCITY



(A)



(B)

964-8-7
080888

FIGURE 1-4. TYPICAL PUMPJET ROTOR BLADE CONFIGURATION, (A) TOP VIEW AND (B) UPSTREAM VIEW

the midspan area but becomes larger toward the hub and thus the camber is designed to be substantially larger.

This unusual rotor blade setup causes various hydrodynamic problems. First of all, since a typical flow incidence angle near the hub should be surprisingly high (e.g., 30°), even a slight error in design may cause flow separation, possibly cavitation and then noise generation. Secondly, even if design is made properly, the same vulnerable situation is generated with a slight flow disturbance or blade deformation due to fabrication inaccuracy.* A recent study at Tetra Tech (see the report by Furuya, et al. (1984)) indicated that some blade deformation, particularly near the hub, could cause an increase of the power coefficient, C_p, by as much as 7 percent. Furthermore, there exists a profound discrepancy between water tunnel test results and actual sea runs. What causes such a discrepancy has not been clarified to date. It is conceivable that 1) a small trim angle (such as 1 ~ 2°) existing at actual sea runs might have caused a change in boundary layer velocity profile, or 2) the boundary layer may be different between the water tunnel and unbounded flow environment so that the pumpjet performance is substantially affected. It should be noted that the utilization of the boundary layer is an advantage in obtaining the pumpjet's high efficiency on one hand but it is a disadvantage in causing many difficult problems on the other hand.

The turning capability of the underwater vehicle thrusted with a pumpjet is said to be inferior to that with, e.g., a counter-rotating propeller. The reason for such inferior turning capability seems also attributable to the utilization of the wake flow; when the vehicle turns, the boundary layer substantially changes. The pumpjet seems to lose a considerable

* Some pumpjet rotors are produced by investment casting process so that the fabrication accuracy cannot be expected to be high.

thrust capability due to the change of boundary layer velocity profile, resulting in a poor turning capability.

Another problem area in the pumpjet lies in the pumpjet design method. The only design method developed to date and used is a two-dimensional graphic method combined with experimental data of Bruce, et al. (1977) despite the fact that the pumpjet experiences a three-dimensional flow. Based on the momentum theorem applied to the cascade configuration, the blade sectional pressure increase Δp is given

$$\Delta p = K\, V_m \cdot \Delta V_\theta$$

where
Δp = local pressure increase through the rotor,
V_m = meridional velocity,
ΔV_θ = circumferential velocity, and
K = a constant determined by the cascade configuration. (1.1)

This two-dimensional momentum theory indicates that, in order to generate a certain pressure increase at a blade section, only the amount of total flow deflection in the circumferential direction (between the inlet and exit) counts, see Figure 1-5. In this method once the sectional blade leading edge and trailing edge angles are determined, then the rest of the blade section can be arbitrarily determined by connecting these predetermined leading and trailing edges, e.g., anglewise smoothly.

One of the obvious problems in this graphic method arises from the fact that the flow coming into the cascade blade cannot exactly follow the blade camber, but substantially deviates from it. What is required therefore is a camber correction, the amount of which

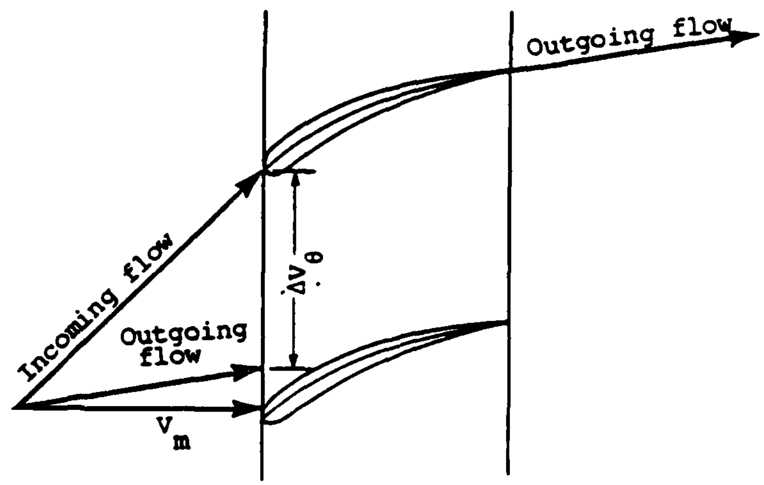


FIGURE 1-5. FLOW CONFIGURATION FOR CASCADE

depends upon the cascade geometry. Unfortunately, a typical pumpjet solidity* near the hub is larger than 2.0 and therefore the camber correction required there becomes as much as 5 times in terms of lift coefficient. It means that the camber graphically constructed should be deformed until the lift coefficient increases by 5 times that graphically obtained. This correction is made semi-empirically based on limited numbers of existing experimental data for cascade blade flows. In this sense, therefore, this graphical method is useless for the blade design near the hub and it can be said that the final design is almost entirely dependent upon these empirical data.

---

* Solidity is defined as a ratio of blade chord length to blade spacing measured normal to the axial direction and the high solidity means more blade packed cascade.

2.0 OBJECTIVES

The objectives of the work to be conducted under the GHR program are therefore:

1) to develop a more reliable and accurate pumpjet design method based on a three-dimensional pump or propeller design theory and then

2) to improve the pumpjet performance characteristics.

The characteristics to be improved include:

a) the susceptibility to flow disturbance and rotor's deformation due to fabrication inaccuracy,

b) the discrepancy problem between the water tunnel test results and high speed sea runs and

c) the poor turning capability.

3.0 METHODOLOGY SELECTION FOR A THREE-DIMENSIONAL PUMPJET DESIGN

3.1 REQUIREMENTS AND CANDIDATES

There exist several possible approaches which can incorporate three-dimensionality into pumpjet design procedure. However, the following aspects should be considered in selecting such a methodology:

1) Moderate three-dimensionality — A pumpjet is usually installed astern of the underwater vehicle hull where the hull shape has a negative slope of tapering shape. Although this provides three-dimensional flow characteristics, its three-dimensionality is rather mild, unlike that in radial pump cases.

2) Capability of determining detailed blade profile shapes as well as pressure distribution — In the previous two-dimensional graphic method Bruce, et al. (1974), the blade profile shape was graphically determined for meeting the head generation requirement. It is mainly for this reason that the method failed to check the possibility of flow separation after the blade was designed. A new method to be developed in this research work should be the one with which the detailed pressure distribution or velocity distribution on the blade can be determined.

3) Accurate loading determination supported by experiments — When the sectional loading is determined analytically in the course of designing a pumpjet, it is usually quite inaccurate since such loading substantially changes due to the effect of adjacent blades. It is therefore necessary for the new method to incorporate the cascade effect into design procedure, or to use an empirical approach to increase such accuracy.

With these features taken into consideration, the following three candidate methods are compared in Section 3.2:

Method I: Katsanis' Quasi-Three-Dimensional Method

Method II: Blade-Through Flow with Blade-to-Blade Flow Method

Method III: Singularity Distribution Method

for which simple explanation will be given in the following.

Method I: Katsanis' (1964) Quasi-Three-Dimensional Method

In this method it is first assumed that a mean stream surface from hub to shroud between blades is known in advance. On this stream surface a two-dimensional solution for the velocity and pressure distributions is obtained. Then, an approximate calculation of the blade surface velocities is made. This method is based on an equation for the velocity gradient along an arbitrary quasi-orthogonal rather than the normal to the streamline. Since the solution is obtained to this quasi-orthogonal line, in this method, an iteration procedure needed in the previous orthogonal-line methods can be eliminated and a solution can be obtained in a single computer run.

Method II: Blade-Through Flow with Blade-to-Blade Flow Method

The blade-through flow is first obtained by, e.g., Streamline Curvature Method (SCM). Once the stream surface is found, it is mapped to a two-dimensional plane so that the blades cut through by the stream surface become a row of blades, i.e., cascade on a plane. On this cascade configuration, the blade-to-blade flow will be solved. Difficulty in doing this lies in the fact that the governing equation is not a Laplace equation any more on this two-dimensional plane, but a Poisson equation due to the deviation of stream surface from a perfect cylinder. In order to account for such deviation of stream surface from a perfect cylinder. In order to account for such deviation into the two-dimensional flow, appropriate source/sink and vortices should be distributed over the entire flow field. This, in turn, results in the change of blade camber shape. The design procedure relies on an iteration scheme.

Method III: Singularity Distribution Method

The method is similar to that used in design of conventional propellers, see, e.g., the work by Kerwin and Leopold (1964). The differences in velocities between the pressure and suction sides of a rotor blade can be represented by distributions of singularities such as source/sink and vortex. The strengths of such singularities are determined by satisfying the boundary conditions on the blade surface as well as those at infinity. The methodology is described in the paper of Kerwin and Leopold (1964) in detail.

The disadvantage of the method lies in the computational complexity and instability. Furthermore, this type of method is suitable for design of devices used in the open field, but not so for those used in the internal flow since it does not take advantage of confined flow configuration available for the latter case.

3.2 COMPARISON AND SELECTION

Table 3-1 provides qualitative evaluation on three candidate methods described in the previous section over various hydrodynamic, numerical and design aspects. As seen from this table, a combination of blade-through method with blade-to-blade flow seems to have an advantage over the other two methods. Particularly, the method has the capability of determining detailed blade profile shape as well as loading and velocity/pressure distribution with accuracy verified by existing cascade experimental data. It is for this reason that the blade-through flow with blade-to-blade flow method has been selected as a basic concept for developing a three-dimensional pumpjet design method.

Similar methods already exist for design of quasi-axial pumps and compressors. However, those methods have many inadequate features in their design procedure. Furthermore, it is assumed in these design methods that the incoming flow is more or less uniform, unlike the pumpjet where the rotor should be designed for highly retarded velocity distribution due to viscous boundary layer on the hull. The following section describes the blade-through flow (BT) with blade-to-blade flow (BTB) method with various aspects of modifications and improvement necessary for developing the pumpjet design method.

TABLE 3-1. QUALITATIVE COMPARISONS FOR THE CANDIDATE METHODS AS A THREE-DIMENSIONAL PUMPJET DESIGN METHOD

FEATURE \ METHOD	METHOD I KATSANIS METHOD	METHOD II BLADE-THROUGH/ BLADE-TO-BLADE FLOW	METHOD III SINGULARITY DISTRIBUTION METHOD
THREE-DIMENSIONALITY	0	Δ	0
DETERMINATION OF DETAILED BLADE PROFILE SHAPE	X	0	0
ACCURACY IN SECTIONAL BLADE LOADING	Δ	0	Δ
CALCULATION OF PRESSURE/ VELOCITY DISTRIBUTION	Δ	0	Δ
SIMPLICITY IN NUMERICAL COMPUTATIONS	Δ	Δ	X
DESIGN CAPABILITY	Δ	0	0
OVERALL EVALUATION	Δ	0	Δ

0 = EXCELLENT
Δ = GOOD
X = POOR

4.0 SELECTED DESIGN METHOD – BLADE-THROUGH/BLADE-TO-BLADE METHOD

Design of a pumpjet for an underwater vehicle requires preliminary information on the vehicle including its geometry and hydrodynamic drag coefficient. Furthermore, most importantly, the velocity profile at an upstream reference section should be obtained either analytically or experimentally. Any error in the velocity profile would result in a pumpjet of lower efficiency or failure of the pumpjet meeting the specifications at the design point. In the present study, it is assumed that this velocity profile is given at a goal speed or at the corresponding Reynolds number.

The first step for design of a pumpjet (see Figure 4-1) is to determine the shroud intake diameter. From the viewpoint of cavitation, the maximum and minimum shroud diameter to prevent cavitation must exist. If it is too large, the rotor blade tip speed becomes too high so that cavitation occurs. On the other hand, if it is too small, the rotation speed must be increased to generate the required head so that the chance of cavitation inception also increases. Another aspect of determining the shroud diameter stems from consideration of the overall propulsive efficiency. The equation for global momentum balance should be able to determine an efficiency-optimum shroud diameter for the given velocity profile and vehicle drag. The detailed mathematical formulation and sample calculations will be given in Section 4.1.

Once the shroud diameter is determined, the streamline will be calculated by using the streamline curvature method (SCM). In this calculation, the loading distribution on the rotor and blade thickness must be assumed in advance. One of the major concerns in using the existing streamline curvature method lies in the fact that SCM may only be used for relatively uniform incoming flow, but may generate a substantial error for a thick wake

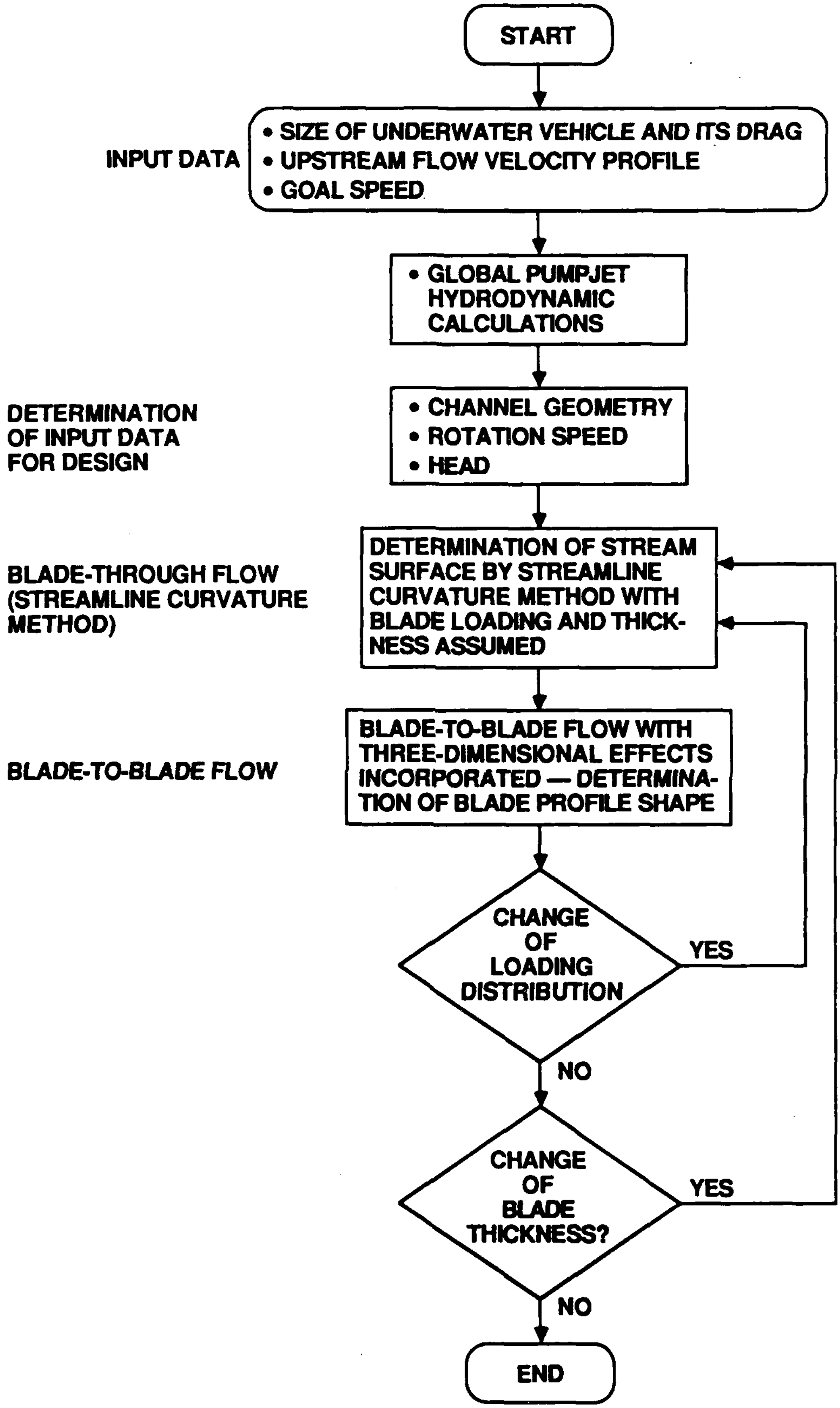


FIGURE 4-1. FLOW CHART OF THE SELECTED PUMPJET DESIGN METHOD

flow, i.e., highly retarded flow due to the viscous boundary layer on the vehicle hull. Detailed mathematical formulation and sample calculations are presented in Section 4.2. Also included are discussions regarding the problems of application of conventional SCM to the thick wake flow.

The next step of the design method is to map the stream tube or surface calculated by SCM onto a plane so that the rotor blades are mapped into cascade configuration. If the stream surface is totally cylindrical shape, the governing equation to be used for the cascade analysis will be a Laplace equation. Unfortunately, the stream surface is of three-dimensional cone shape in general for the tail cone section of the underwater vehicle. The field governing equation now becomes a Poisson equation, for which the results of powerful potential theory analysis are no more applicable. A method of correcting the effect of the Poisson equation on the potential theory results is introduced to modify the blade profile shape obtained in the potential theory. In choosing the blade profile shape, the experimental data are used to ensure that there is no chance of flow separation due to overloading on the blade. Furthermore, based on the calculated velocity along the blade, the cavitation inception is checked. If there exists a chance of either flow separation or cavitation, the loading distribution from hub to tip should be changed. If such a change is made, and/or thickness of blades is changed, the streamline curvature method should be used again to determine the new location of streamline or stream surface. This iterative Section 4.3 describes the technical approach to be used for the blade-to-blade flow analysis.

4.1 PUMPJET GLOBAL HYDRODYNAMICS

It is a well-known fact that the pumpjet utilizes the tailcone low-energy, boundary layer flow in order to achieve its high efficiency. It means that the optimum* pumpjet design depends entirely upon the incoming flow velocity profile.

Figure 4-2 shows a schematic diagram of an underwater vehicle tail cone/pumpjet flow. (1) and (7) in Figure 4-2 are considered to be the upstream and downstream reference stations, respectively, where it is assumed that the freestream static pressure exists, whereas (2) and (3) are the rotor inlet and exit stations.

4.1.1 Calculation of Thrust Force

The thrust force, T, due to the pumpjet work can be determined by applying the momentum equation to the control volume enclosed by stations (1), (7) and the stagnation streamline (see Figure 4-2).

$$T = \int_{r_{H7}}^{r_{T7}} d\dot{m}_7 \cdot V_7(r)\cos\theta_7 - \int_{r_{H1}}^{r_{T1}} d\dot{m}_1 \cdot V_1(r)\cos\theta_1 \qquad (4.1.1\text{-}1)$$

$$\text{where} \quad \left.\begin{aligned} d\dot{m}_7 &= \rho 2\pi r V_7(r)\cos\theta_7\, dr \\ d\dot{m}_1 &= \rho 2\pi r V_1(r)\cos\theta_1\, dr \end{aligned}\right\} \qquad (4.1.1\text{-}2)$$

$V_1(r), V_7(r)$ = meridional flow velocities at (1) and (7), respectively.

θ_1, θ_7 = meridional flow angles at (1) and (7), respectively.

---

*Note: By "optimum" pumpjet design it means that of proving the maximum propulsion efficiency.

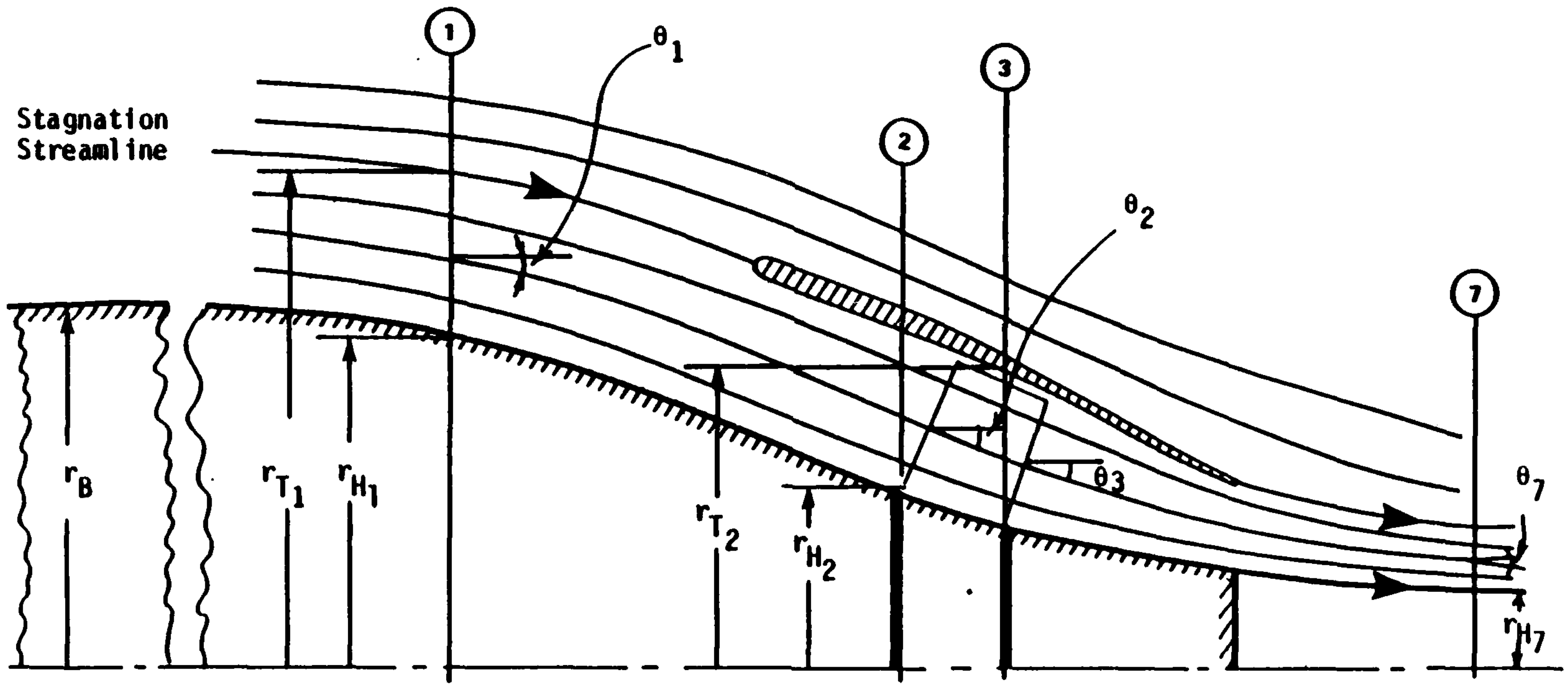


FIGURE 4-2. SCHEMATIC DIAGRAM OF PUMPJET FLOW

964-8-10
080888

Therefore

$$T = \int_{r_{H7}}^{r_{T7}} \rho 2\pi r V_7^2(r)\cos^2\theta_7\,dr - \int_{r_{H1}}^{r_{T1}} \rho 2\pi r V_1^2(r)\cos^2\theta_1\,dr \qquad (4.1.1\text{-}3)$$

Defining the following quantities,

$$\frac{\bar{\bar{V}}_7}{V_\infty} = \frac{\displaystyle\int_{r_{H7}/r_B}^{r_{T7}/r_B} \left(\frac{V_7(r)}{V_\infty}\right)^2 \frac{r}{r_B}\cdot\cos\theta_7\,d\left(\frac{r}{r_B}\right)}{\displaystyle\int_{r_{H7}/r_B}^{r_{T7}/r_B} \frac{V_7(r)}{V_\infty}\frac{r}{r_B}\cos\theta_7\cdot d\left(\frac{r}{r_B}\right)} \qquad (4.1.1\text{-}4)$$

$$\frac{\bar{\bar{V}}_1}{V_\infty} = \frac{\displaystyle\int_{r_{H1}/r_B}^{r_{T1}/r_B} \left(\frac{V_1(r)}{V_\infty}\right)^2 \frac{r}{r_B}\cos\theta_1\,d\left(\frac{r}{r_B}\right)}{\displaystyle\int_{r_{H1}/r_B}^{r_{T1}/r_B} \frac{V_1(r)}{V_\infty}\frac{r}{r_B}\cos\theta_1\,d\left(\frac{r}{r_B}\right)} \qquad (4.1.1\text{-}5)$$

and normalizing T, Eqn. (4.1.1-3) becomes

$$C_T = \frac{T}{\frac{1}{2}\rho V_\infty^2\cdot A_B}$$

$$= \frac{\displaystyle\int_{r_{H7}}^{r_{T7}} \rho 2\pi r V_7^2(r)\cos^2\theta_7\,dr - \int_{r_{H1}}^{r_{T1}} \rho 2\pi r V_1^2(r)\cos^2\theta_1\,dr}{\frac{1}{2}\rho V_\infty^2\,\pi r_B^2}$$

$$= 4\left[\int_{r_{H7}/r_B}^{r_{T7}/r_B} \left(\frac{V_7(r)}{V_\infty}\right)^2\left(\frac{r}{r_B}\right)\cos^2\theta_7\,d\left(\frac{r}{r_B}\right) - \int_{r_{H1}/r_B}^{r_{T1}/r_B} \left(\frac{V_1(r)}{V_\infty}\right)^2\left(\frac{r}{r_B}\right)\cos^2\theta_1\,d\left(\frac{r}{r_B}\right)\right]$$

$$= 2\left(\frac{\bar{\bar{V}}_7}{V_\infty}\cos\theta_7 - \frac{\bar{\bar{V}}_1}{V_\infty}\cos\theta_1\right)C_m \qquad (4.1.1\text{-}6)$$

where the mass conservation equation below has been used;

$$C_m = C_{m1} = C_{m7} \tag{4.1.1-7}$$

$$\dot{m}_1 = \int_{r_{H1}}^{r_{T1}} \rho 2\pi r V_1 \ (r) \cos\theta_1 \ dr \tag{4.1.1-8}$$

$$\dot{m}_7 = \int_{r_{H7}}^{r_{T7}} \rho 2\pi r V_7 \ (r) \cos\theta_7 \ dr \tag{4.1.1-9}$$

$$C_{m1} = \frac{\dot{m}_1}{\rho V_\infty \pi r_B^{\ 2}} = 2\int_{r_{H1}/r_B}^{r_{T1}/r_B} \frac{V_1(r)}{V_\infty} \cdot \frac{r}{r_B} \cos\theta_1 \ d\left(\frac{r}{r_B}\right) \tag{4.1.1-10}$$

$$C_{m7} = \frac{\dot{m}_7}{\rho V_\infty \pi r_B^{\ 2}} = 2\int_{r_{H7}/r_B}^{r_{T7}/r_B} \frac{V_7(r)}{V_\infty} \frac{r}{r_B} \cos\theta_7 \cdot d\left(\frac{r}{r_B}\right) \tag{4.1.1-11}$$

Let's define

$$\frac{\overline{\overline{\Delta V_a}}}{V_\infty} = \frac{\overline{\overline{V_7 \cos\theta_7}}}{V_\infty} - \frac{\overline{\overline{V_1 \cos\theta_1}}}{V_\infty} , \tag{4.1.1-12}$$

then we obtain, from (4.1.1-6)

$$C_T = 2 \cdot \left(\frac{\overline{\overline{\Delta V_a}}}{V_\infty}\right) \cdot C_m \tag{4.1.1-13}$$

Note that

$$C_m = 2 \cdot \left(\frac{\overline{V}_1}{V_\infty}\right) \frac{A_1}{A_B} \tag{4.1.1-14}$$

where

$$\frac{\bar{V}_1}{V_\infty} = \frac{\int_{r_{H1}/r_B}^{r_{T1}/r_B} \frac{V_1(r)}{V_\infty} \left(\frac{r}{r_B}\right) \cos\theta_1 \, d\left(\frac{r}{r_B}\right)}{\int_{r_{H1}/r_B}^{r_{T1}/r_B} \left(\frac{r}{r_B}\right) \cos\theta_1 \, d\left(\frac{r}{r_B}\right)} \qquad (4.1.1\text{-}15)$$

4.1.2 Relationship Between Pump Head and Thrust

The hydraulic head of the pump $\tilde{H}_R$ is given

$$\frac{\tilde{H}_R}{V_\infty^2/2g} = \frac{2\int_{r_{H3}/r_B}^{r_{T3}/r_B} \frac{V_3}{V_\infty} \frac{U}{V_\infty} \frac{V_{\theta 3}}{V_\infty} \frac{r}{r_B} \frac{1}{\cos\theta_3} \, d\left(\frac{r}{r_B}\right)}{\int_{r_{H3}/r_B}^{r_{T3}/r_B} \frac{V_3}{V_\infty} \frac{r}{r_B} \frac{1}{\cos\theta_3} \, d\left(\frac{r}{r_B}\right)} \qquad (4.1.2\text{-}1)$$

where $V_3(r)$, $V_{\theta 3}(r)$ = meridional and circumferential velocities at station (3), respectively.

With the hydraulic efficiency η_R introduced, the actual head generated in the fluid is $\tilde{H}$,

$$\tilde{H} = \eta_R \cdot \tilde{H}_R \qquad (4.1.2\text{-}2)$$

where $\tilde{H}$ can be defined

$$\frac{\tilde{H}}{V_\infty^2/2g} = \left(\frac{\tilde{V}_7}{V_\infty}\right)^2 - \left(\frac{\tilde{V}_1}{V_\infty}\right)^2 + K_1 \left(\frac{\tilde{V}_1}{V_\infty}\right)^2 \qquad (4.1.2\text{-}3)$$

where

$$\left(\frac{\tilde{V}_7}{V_\infty}\right)^2 = \frac{\int_{r_{H7}/r_B}^{r_{T7}/r_B} \left(\frac{V_7(r)}{V_\infty}\right)^3 \frac{r}{r_B} \cos\theta_7 \, d\left(\frac{r}{r_B}\right)}{\int_{r_{H7}/r_B}^{r_{T7}/r_B} \left(\frac{V_7(r)}{V_\infty}\right) \frac{r}{r_B} \cos\theta_7 \, d\left(\frac{r}{r_B}\right)} \qquad (4.1.2\text{-}4)$$

$$\frac{\bar{V}_1}{V_\infty} = \frac{\int_{r_{H1}/r_B}^{r_{T1}/r_B} \frac{V_1(r)}{V_\infty}\left(\frac{r}{r_B}\right)\cos\theta_1\, d\left(\frac{r}{r_B}\right)}{\int_{r_{H1}/r_B}^{r_{T1}/r_B}\left(\frac{r}{r_B}\right)\cos\theta_1\, d\left(\frac{r}{r_B}\right)} \qquad (4.1.2\text{-}5)$$

K_1 = head loss coefficient between station (1) and (7) (but mostly inlet loss and see Appendix for determining K_1).

Since

$$\bar{\bar{\Delta V}}_m = \bar{\bar{V}}_7 - \bar{\bar{V}}_1 \qquad (4.1.2\text{-}6)$$

and $\Delta\bar{\bar{V}}_m$ can be approximated as $\Delta\bar{\bar{V}}_m \simeq \tilde{V}_7 - \tilde{V}_1$, Eqn. (4.1.2-3) becomes

$$\frac{H}{V_\infty^2/2g} = 2\,\frac{\bar{\bar{\Delta V}}_m}{V_\infty}\cdot\frac{\tilde{V}_1}{V_\infty} + \left(\frac{\bar{\bar{\Delta V}}_m}{V_\infty}\right)^2 + K_1\left(\frac{\tilde{V}_1}{V_\infty}\right)^2 . \qquad (4.1.2\text{-}7)$$

From Eqns. (4.1.1-12) and (4.1.2-7),

$$\frac{\bar{\bar{\Delta V}}_m}{V_\infty} = \frac{\frac{\bar{\bar{\Delta V}}_a}{V_\infty} + \frac{\bar{\bar{V}}_1}{V_\infty}\left(\cos\theta_1 - \cos\theta_7\right)}{\cos\theta_7} \qquad (4.1.2\text{-}8)$$

4.1.3 Power Calculation

The power to be used on the pumpjet rotor shaft can be calculated by integrating the product of the local head and the local mass flow rate over the entire duct flow,

$$P = \int \rho g \Delta Q H \ ,$$

or in terms of the power coefficient

$$C_P = \frac{P}{\frac{1}{2}\rho V_\infty^3 A_B} = \frac{\int \rho g \Delta Q H}{\frac{1}{2}\rho V_\infty^3 \pi r_B^2}$$

$$= \frac{\rho g \int_{r_{H3}}^{r_{T3}} \frac{V_{\theta 3} \cdot U}{g} \cdot V_3(r)\, 2\pi r \, \frac{1}{\cos\theta_3}\, dr}{\frac{1}{2}\rho V_\infty^3 \pi r_B^2}$$

or

$$C_P = 4\int_{r_{H3}/r_B}^{r_{T3}/r_B} \left(\frac{V_{\theta 3}}{V_\infty}\right)\left(\frac{U}{V_\infty}\right)\left(\frac{V_3(r)}{V_\infty}\right)\frac{r}{r_B}\,\frac{1}{\cos_{\theta 3}}\, d\left(\frac{r}{r_B}\right) . \qquad (4.1.3\text{-}1)$$

From (4.1.2-1),

$$C_P = 2\,\frac{\widetilde{H}_R}{V_\infty^2/2g} \cdot \int_{r_{H3}/r_B}^{r_{T3}/r_B} \frac{V_3}{V_\infty}\,\frac{r}{r_B}\,\frac{1}{\cos_{\theta 3}}\, d\left(\frac{r}{r_B}\right)$$

$$= \frac{\widetilde{H}_R}{V_\infty^2/2g} \cdot C_m \ ,$$

or

$$C_P = \frac{1}{\eta_R} \cdot \frac{\widetilde{H}}{V_\infty^2/2g} \cdot C_m \ . \qquad (4.1.3\text{-}2)$$

Therefore, the propulsive efficiency η_p can be calculated from the following definition:

$$\eta_P = \frac{C_T}{C_P} \qquad (4.1.3\text{-}3)$$

4.1.4 Procedure of Calculating Propulsive Efficiency with $V_1(r)$, etc. given and Sample Results

Now all tools for calculating the propulsive efficiency η_p with V_1 (r) given are provided. The principal equations to be used will be Eqns. (4.1.1-13), (4.1.2-6), (4.1.2-8), (4.1.3-2), and (4.1.3-3). A flow chart describing the calculation procedure is given in Figure 4-3.

Sample velocity profiles, $V_1(r)$ and $V_2(r)$, are shown in Figures 4-4 and 4-5. The flow angles, θ_1, θ_2, and θ_7 can be obtained from the drawings of a typical underwater vehicle tail cone profile. The head loss coefficient, K_1, can be calculated from the formula given in Appendix A with the velocity distributions and pressure distributions both at stations (1) and (2), which are also given in Figures 4-4 and 4-5.

Figures 4-6 to 4-7 show the results of calculations made for various parameters, including the shroud opening diameter, r_1/r_B, the exit jet flow angle θ_7, and flow velocity amplification factor. The design values of r_1/r_B and flow amplification factor are known, i.e.,

$$(r_1/r_B)_D = .93.$$

Flow Amplification Factor = 1

but that of θ_7 is not known except for the fact that the average geometric angle of the shroud and tail cone angle at exit is about 11°. Thereoetically, however, the jet coming out from the shroud exit with 11° should align itself in the direction of the body axis, indicating that θ_7 could be zero. The present calculations were therefore made for θ_7 = 0°, 5°, 8° and 11°.

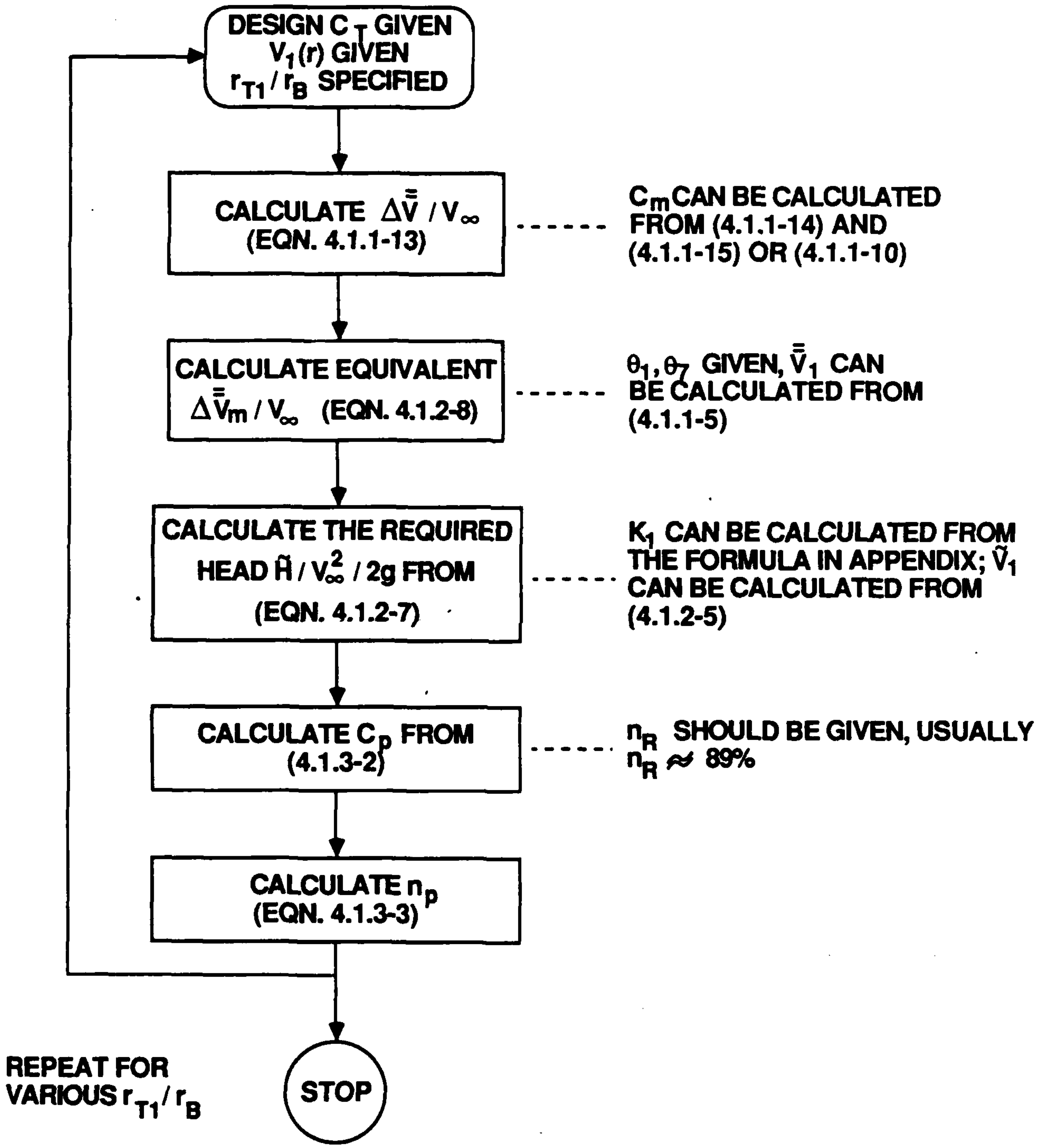


FIGURE 4-3. FLOW CHART FOR CALCULATION OF THE PUMPJET EFFICIENCY

964-8-12
080888

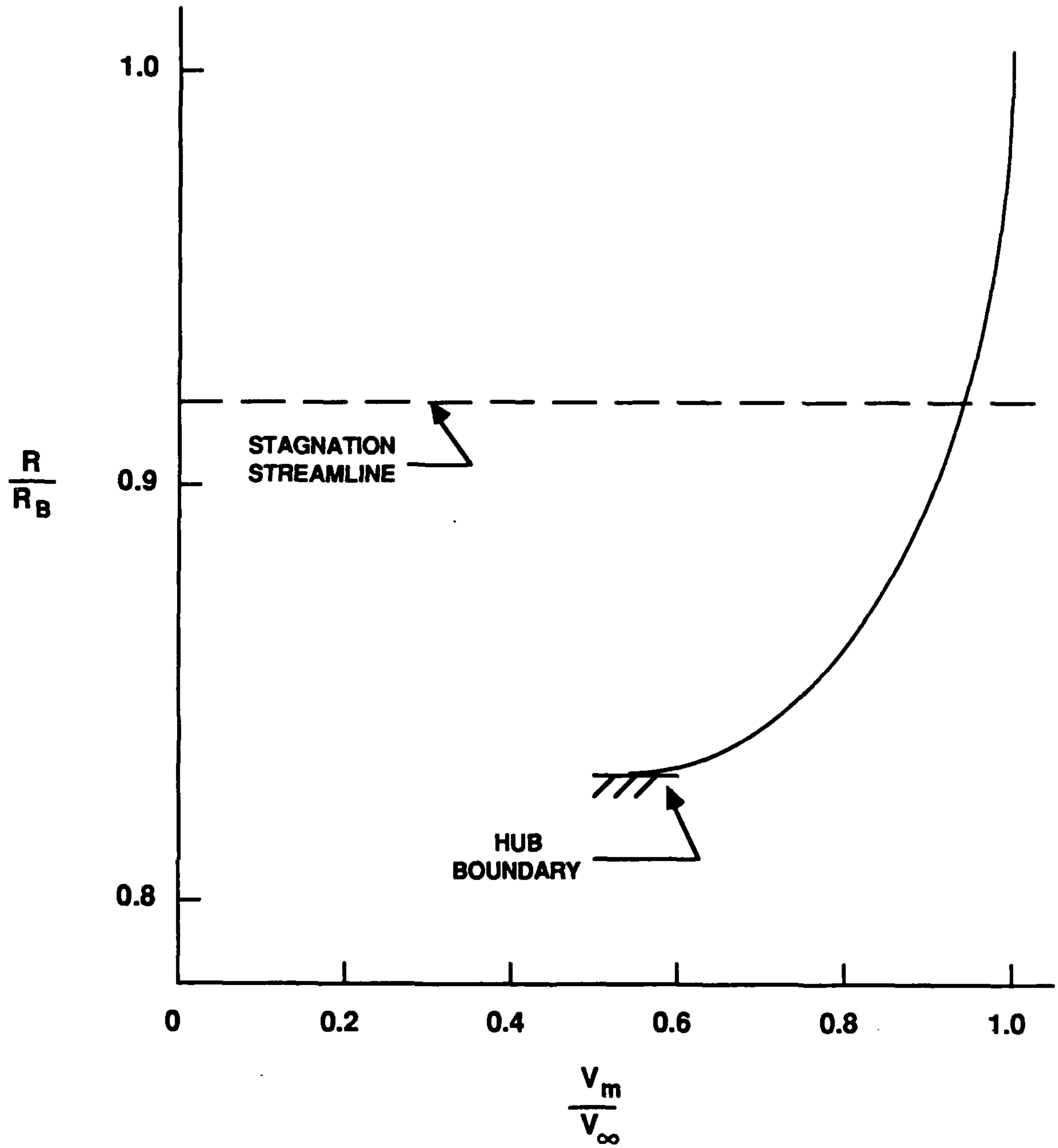


FIGURE 4-4. MERIDIONAL VELOCITY AT STATION (1) OF FIGURE 4-2

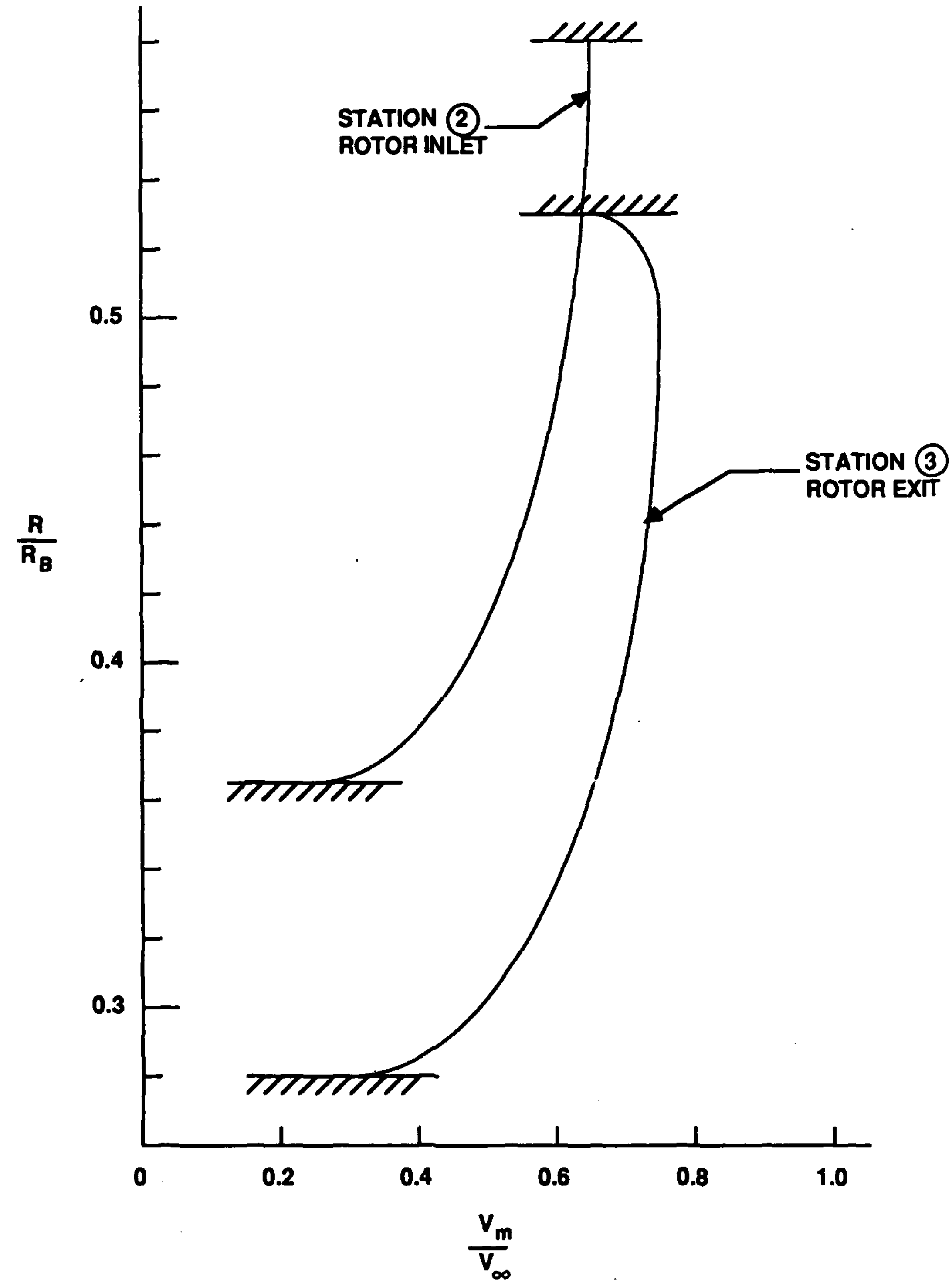


FIGURE 4-5. MERIDIONAL VELOCITY DISTRIBUTIONS AT STATIONS ② AND ③ OF FIGURE 4-2

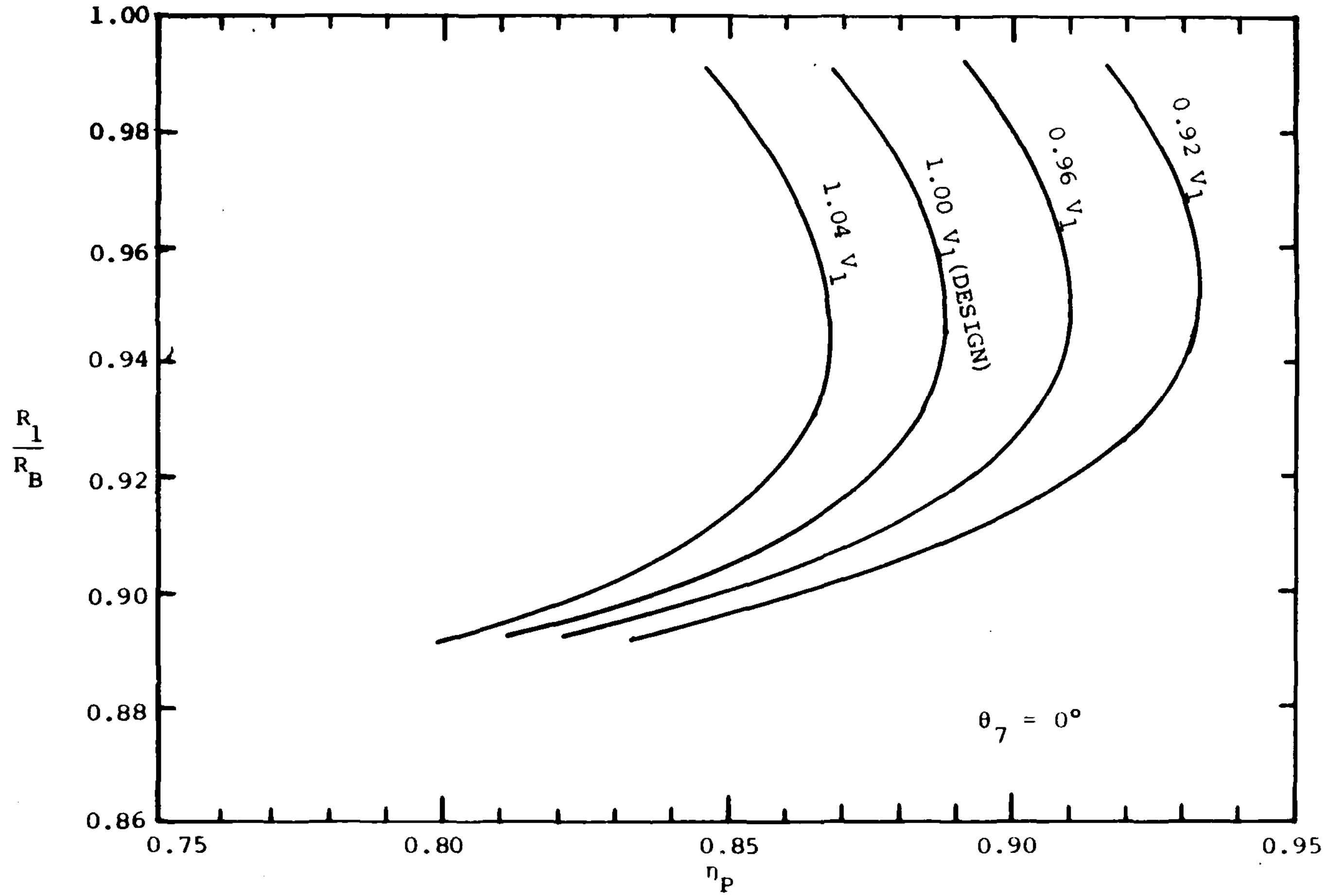


FIGURE 4-6. CALCULATED PROPULSIVE EFFICIENCY AS FUNCTIONS OF R_1 / R_B (STAGNATION STREAMLINE RADIUS) WITH THE INCOMING FLOW VELOCITY AMPLIFICATION FACTOR, $\theta_7 = 0°$

964-8-15
080888

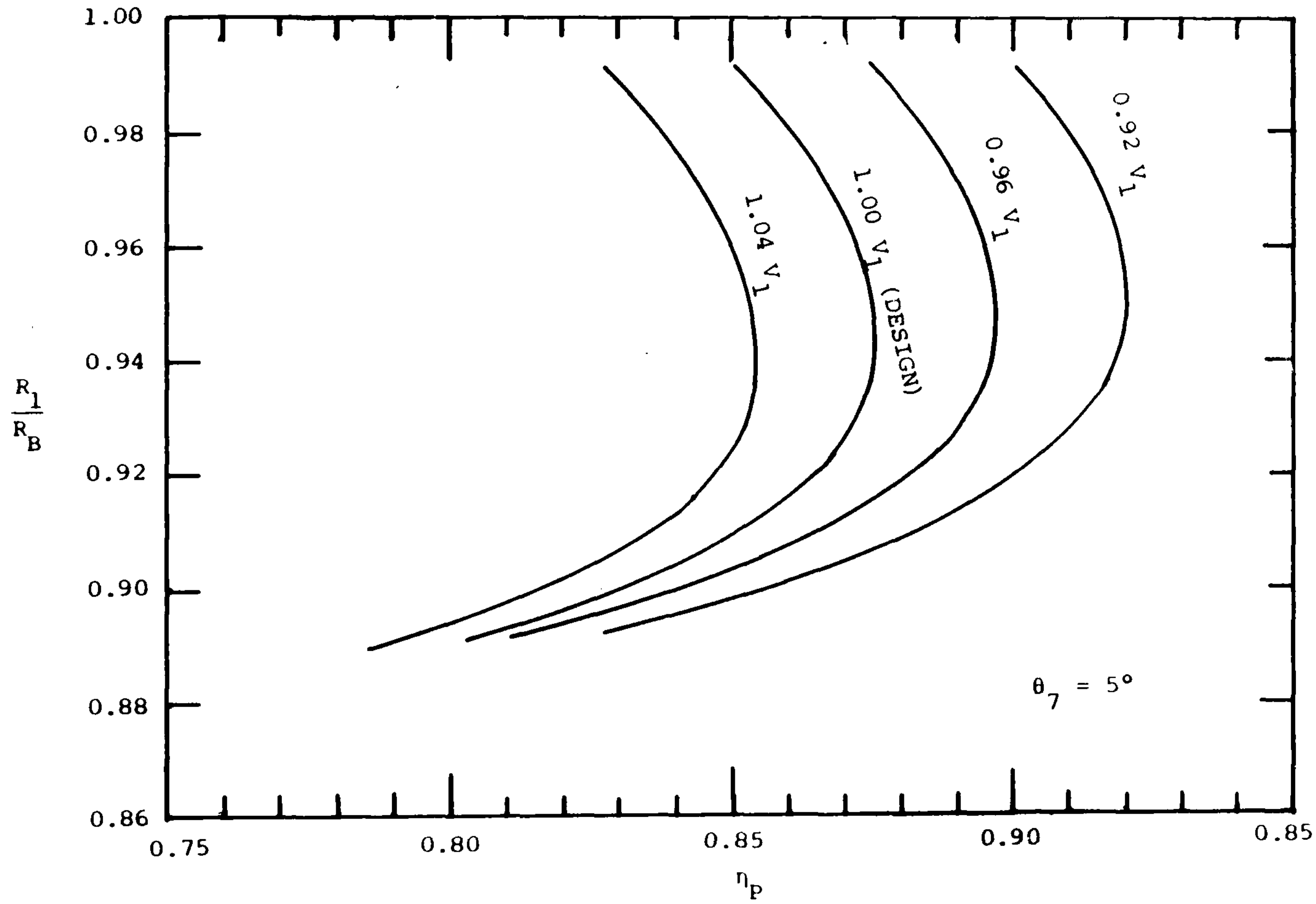


FIGURE 4-7. THE SAME AS FIGURE 4-6, EXCEPT FOR $\theta_7 = 5°$

964-8-16
080888

The shroud opening, r_1/r_B, was also varied in the present analysis in order to determine the optimum shroud opening radius in terms of efficiency. Also changed was the incoming flow velocity amplitude to investigate a possibility of pumpjet efficiency improvement in combination with the tail cone flow pattern change.

The pump hydraulic efficiency η_R for this type of pump, used in the analysis, is about 89°, which is the measured value by many pump makers.

Figures 4-6 to 4-8 show the calculated propulsive efficiencies as a function of the shroud opening r_1/r_B with the flow amplification factor as a parameter for θ_7 = 0°, 5° and 11°, respectively. As can be seen from these figures, the efficiency curve has the maximum value at an r_1/r_B value specific for the conditions used.

Figure 4-6 shows that the efficiency at the design condition should be 88.4% when θ_7 = 0° is assumed. The design shroud opening, r_1/r_B = .93, is slightly on the smaller side than that for the maximum efficiency. The maximum efficiency of 88.8% can be obtained at a slightly larger shroud radius, i.e., r_1/r_B = .93, is slightly on the smaller side than that for the maximum efficiency. The maximum efficiency of 88.8% can be obtained at a slightly larger shroud radius, i.e., r_1/r_B = .95. Also seen from Figure 4-6 is the fact that the smaller the incoming flow velocity, the larger the maximum propulsive efficiency. This indicates that, if the incoming velocity amplitude at the tail cone area can be reduced by some means, the propulsive efficiency of the pumpjet is substantially increased.

It should also be remembered that the actual efficiency achieved is 76.9%, much lower than any of the values calculated here.

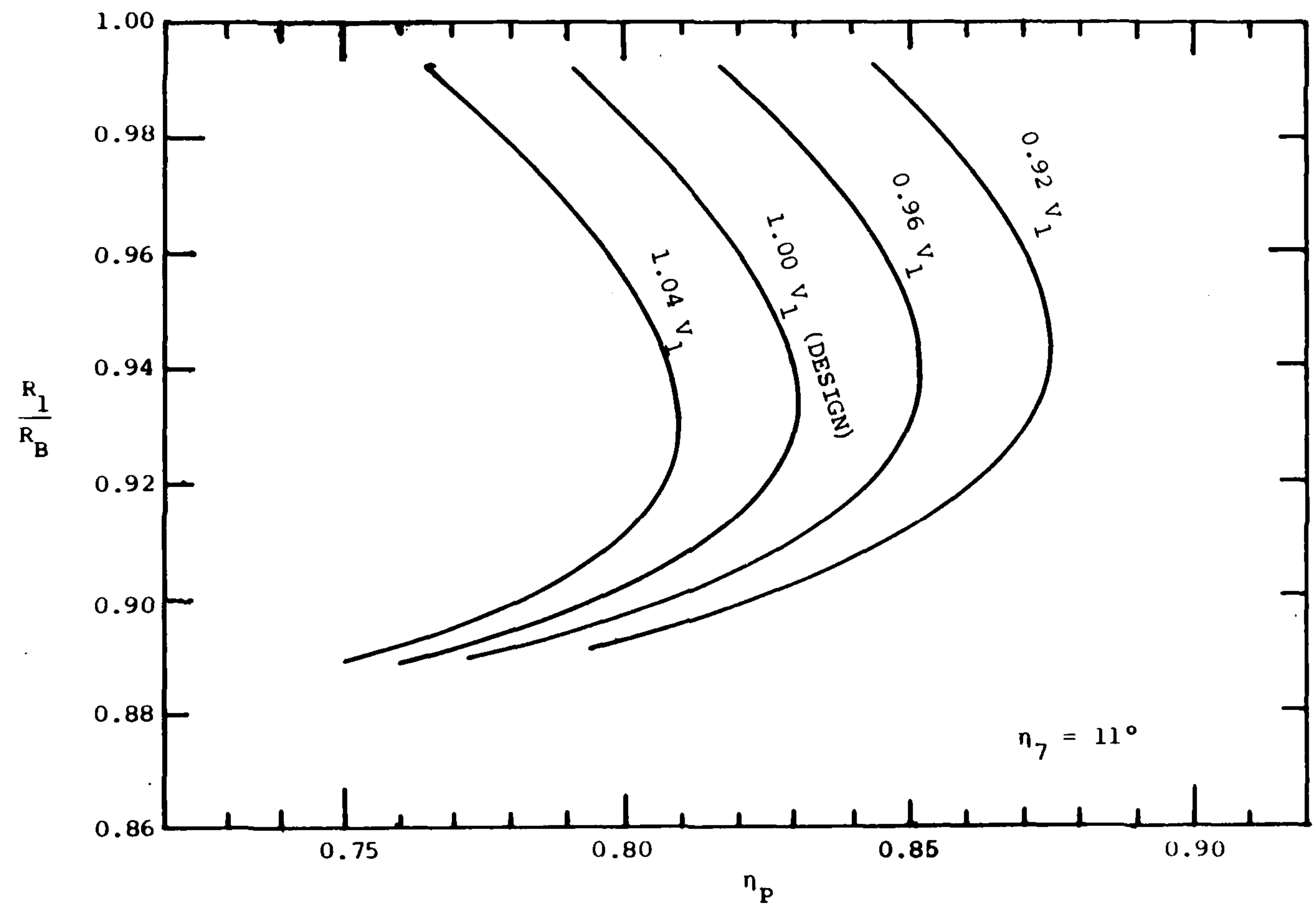


FIGURE 4-8. THE SAME AS FIGURE 4-6, EXCEPT FOR $\theta_7 = 11°$

964-8-17
080888

As θ_7 increases, the efficiency curves shift to the lower efficiency side, see Figures 4-6 to 4-8. This is naturally expected since the jet thrusting force is not effectively utilized as θ_7 increases.

It may also be coincidental, as seen in Figure 4-8, that if $\theta_7 = 11°$ is used as obtained from the pumpjet exit geometry, the current shroud opening, $r_1/r_B = .93$, is the optimum selection for providing the maximum efficiency, 82.9%, smaller than that of the original design. It should be pointed out that if $\theta_7 = 11°$ is the true exit jet flow angle, an increase in the shroud radius for alleviating the flow separation problem may cost a substantial efficiency loss (see Figure 4-8). On the other hand, if $\theta_7 = 0°$ is the true value, a moderate increase (e.g., 2%) in the shroud radius will increase the efficiency in addition to the efficiency gained due to the suppression of the flow separation. However, again the penalty exists when the amount of shroud radius increase exceeds more than four percent.

Figure 4-9 summarizes the present calculations in terms of the maximum propulsive efficiency with θ_7 and the flow amplification factor as parameters. It is shown that a substantial efficiency improvement may be achieved by:

1) reducing the incoming flow velocity amplitude by modifying the tail cone profile shape and thus changing the boundary layer flow,

2) choosing the optimum shroud opening radius depending on the flow conditions (see also Figures 4-6 to 4-8),

3) redirecting the jet flow at the shroud exit as close to the body axis as possible, if the jet flow of the current design is not.

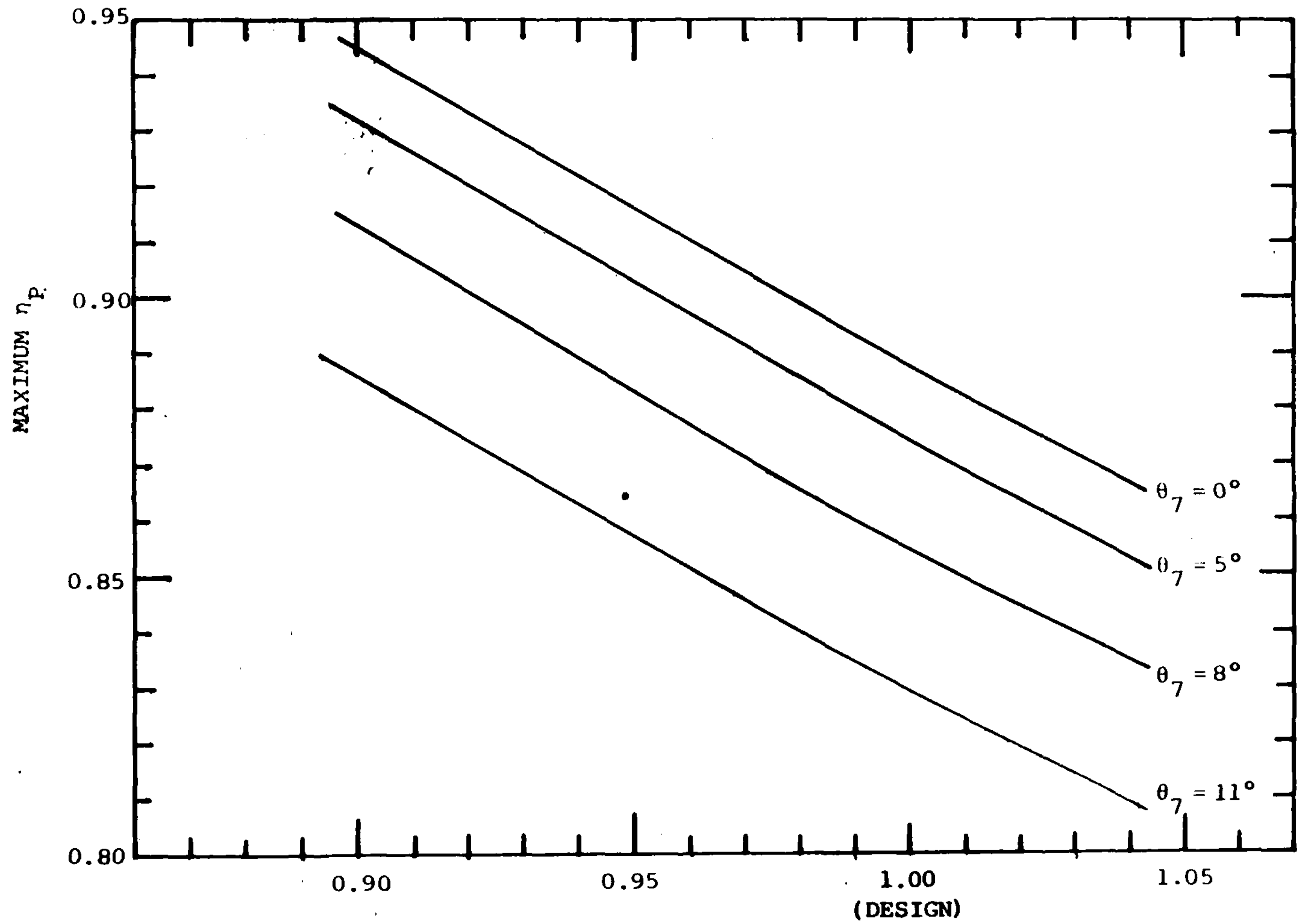


FIGURE 4-9. MAXIMUM PROPULSIVE EFFICIENCY AS A FUNCTION OF THE INCOMING FLOW VELOCITY AMPLIFICATION FACTOR WITH θ_7 AS A PARAMETER

964-8-18
080888

4.2 BLADE-THROUGH FLOW ANALYSIS – STREAMLINE CURVATURE METHOD (SCM)

4.2.1 Mathematical Formulation

From the definition of entropy, S, in the second law of thermodynamics, the following relationship is obtained for a reversible transformation, i.e.,

$$T\nabla S = \nabla Q \tag{4.2.1-1}$$

where T is the temperature and ∇Q is the amount of heat the system under consideration receives. On the other hand, the first law of thermodynamics says

$$\nabla E = \nabla Q + \nabla W \tag{4.2.1-2}$$

where E is the internal energy and W is the work performed on the fluid. Since $\nabla W = -p\nabla v$, (4.2.1-2) becomes

$$\begin{aligned} \nabla E &= \nabla Q - p\nabla v \\ &= \nabla Q - p\nabla\left(\frac{1}{\rho}\right) \end{aligned} \tag{4.2.1-3}$$

where v is the specific volume of fluid and, in terms of fluid density, ρ, $v = 1/\rho$. The definition of enthalpy, H, is given by

$$H = \frac{1}{2}u^2 + E + \frac{p}{\rho} + \psi \tag{4.2.1-4}$$

where u is the amplitude of low velocity and ψ is the potential energy. Gradient of H yields

$$\nabla H = \nabla\left(\frac{1}{2}u^2 + \psi\right) + \nabla E + \frac{1}{\rho}\nabla p + p\nabla\left(\frac{1}{\rho}\right)$$

From the above equation and Eqn. (4.2.1-3),

$$\nabla H = T\nabla S + \nabla\left(\frac{1}{2}u^2 + \psi\right) + \frac{1}{\rho}\nabla p \qquad (4.2.1\text{-}5)$$

The steady-state momentum theorem gives

$$\rho \underline{u} \cdot \nabla \underline{u} = -\nabla p - \rho\nabla\psi \qquad (4.2.1\text{-}6)$$

where an assumption has been made that $-\nabla\psi = \underline{F}$, where $\underline{F}$ is an external force. By using a vector identity, $\underline{u} \times (\nabla \times \underline{u}) = 1/2\ \nabla\ u^2 - \underline{u} \cdot \nabla\underline{u}$, Eqn. (4.2.1-6) is now written

$$\underline{u} \times \left(\nabla \times \underline{u}\right) = \frac{1}{2}\nabla u^2 + \nabla\psi + \frac{1}{\rho}\nabla p$$

$$\text{or}\quad \underline{u} \times 2\underline{\omega} = \nabla\left(\frac{u^2}{2} + \psi\right) + \frac{1}{\rho}\nabla p \qquad (4.2.1\text{-}7)$$

where $\underline{\omega} = \nabla \times \underline{u}$. Substituting Eqn. (4.2.1-5) into (4.2.1-7) gives

$$\underline{u} \times 2\underline{\omega} = \nabla H - T\nabla S \quad , \qquad (4.2.1\text{-}8)$$

a relation first found by Crocco (1937), which will be used to derive the formula used for SCM hereafter.

By using the cylindrical coordinate system (r, θ, z), the velocity components of $\underline{u}$ are defined by

$$\underline{u} = (u_r, u_\theta, u_z) \quad . \qquad (4.2.1.9)$$

Thus, the components of vortex term $\underline{\omega}$ are written

$$\text{(a)} \quad 2\underline{\omega}_r = \left(\nabla \times \underline{u}\right)_r = \frac{1}{r}\left(\frac{\partial u_z}{\partial \theta} - \frac{\partial r u_\theta}{\partial z}\right)$$

$$\text{(b)} \quad 2\underline{\omega}_\theta = \left(\nabla \times \underline{u}\right)_\theta = \frac{\partial u_r}{\partial z} - \frac{\partial u_z}{\partial r}$$

$$\text{(c)} \quad 2\underline{\omega}_z = \left(\nabla \times \underline{u}\right)_z = \frac{1}{r}\left(\frac{\partial r u_\theta}{\partial r} - \frac{\partial u_r}{\partial \theta}\right) \qquad (4.2.1\text{-}10)$$

By introducing a direction m, defined by, (see also Fig. 4-10)

$$\text{(a)} \quad dm : dr : dz = u_m : u_r : u_z$$

$$\text{(b)} \quad u_m^2 = u_r^2 + u_z^2$$

$$\text{(c)} \quad \tan\phi = u_r / u_z$$

$$\text{(d)} \quad u_r = u_m \sin\phi$$

$$\text{(e)} \quad u_z = u_m \cos\phi \quad , \qquad (4.2.1\text{-}11)$$

it becomes evident that the m-direction is the "meridional direction" or on the projection of streamline in the r-z plane. The directional derivative with respect to m then becomes

$$\text{(a)} \quad u_m \frac{\partial}{\partial m} = u_r \frac{\partial}{\partial r} + u_z \frac{\partial}{\partial z}$$

$$\text{(b)} \quad \frac{\partial}{\partial m} = \frac{\partial r}{\partial m}\frac{\partial}{\partial r} + \frac{\partial z}{\partial m}\frac{\partial}{\partial z}$$

$$= \sin\phi \frac{\partial}{\partial r} + \cos\phi \frac{\partial}{\partial z} \qquad (4.2.1\text{-}12)$$

where (4.2.1-11) has been used.

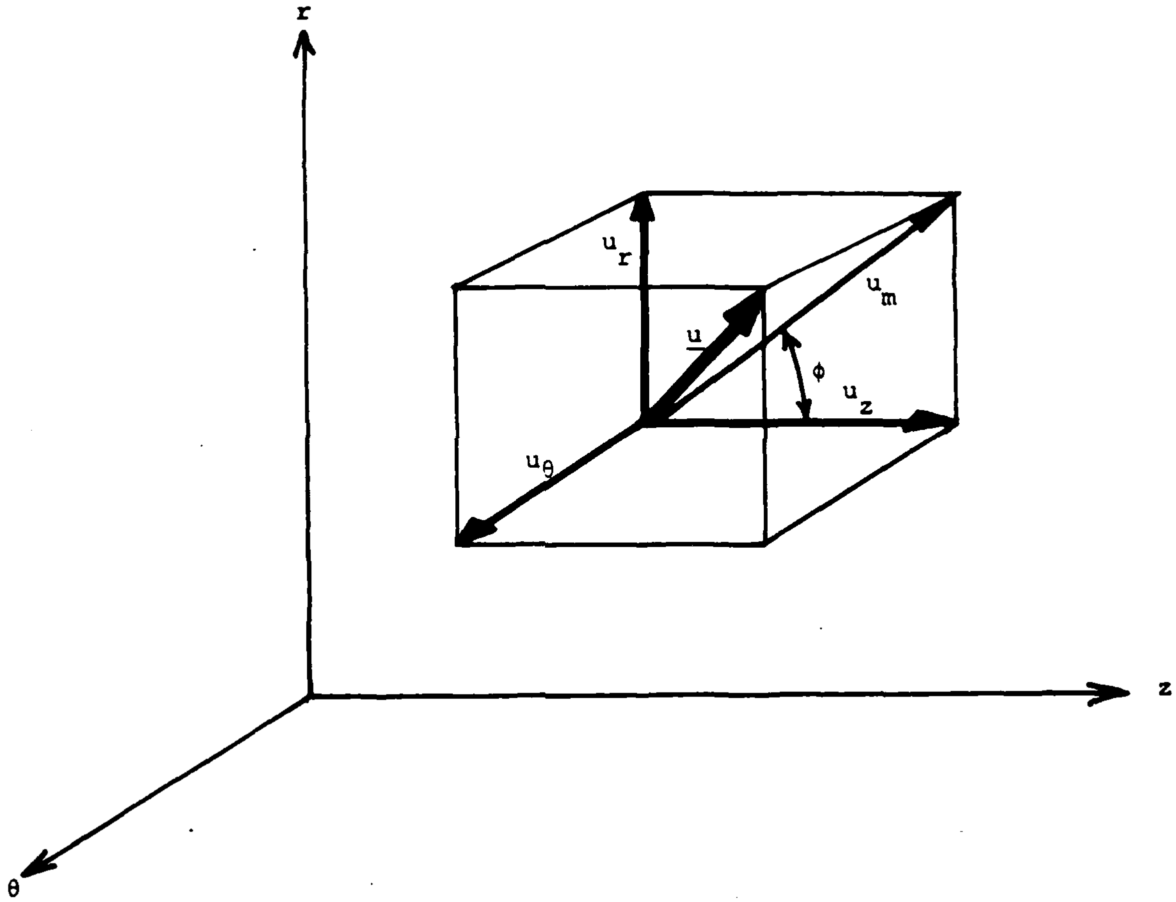


FIGURE 4-10. VELOCITY DIAGRAM

964-8-19
080888

Furthermore, the θ-component of the Crocco equation (4.2.1-8) gives

$$\frac{u_z}{r}\left(\frac{\partial u_z}{\partial \theta} - \frac{\partial r u_\theta}{\partial z}\right) - \frac{u_r}{r}\left(\frac{\partial r u_\theta}{\partial r} - \frac{\partial u_r}{\partial \theta}\right)$$

$$= \frac{1}{r}\left(\frac{\partial H}{\partial \theta} - T\frac{\partial S}{\partial \theta}\right) \qquad (4.2.1\text{-}13)$$

Under the assumption of axisymmetricity,

$$\frac{\partial H}{\partial \theta} = 0$$

$$T\frac{\partial S}{\partial \theta} = 0 \ ,$$

so that Eqn. (4.2.1-13) becomes

$$u_r \frac{\partial u_r}{\partial \theta} + u_z \frac{\partial u_z}{\partial \theta} = u_r \frac{\partial r u_r}{\partial r} + u_z \frac{\partial r u_\theta}{\partial z}$$

or $$\frac{1}{2} \cdot \frac{\partial}{\partial \theta}\left(u_r^2 + u_z^2\right) = u_m \frac{\partial r u_\theta}{\partial m}$$

Using (4.2.1-11b),

$$\frac{\partial u_m}{\partial \theta} = \frac{\partial r u_\theta}{\partial m} \qquad (4.2.1\text{-}14)$$

It is now ready to perform a coordinate transform of Eqn. (4.2.1-10) by using Eqns. (4.2.1-11), (4.2.1-12) and (4.2.1-14). The result is

$$\text{(a)} \quad 2\omega_r = (\nabla x \underline{u})_r = \frac{\tan\phi}{r}\left[\frac{\partial r u_\theta}{\partial r} - \sin\phi \frac{\partial r u_\theta}{\partial m} - u_m \cos\phi \frac{\partial \phi}{\partial \theta}\right]$$

$$\text{(b)}\quad 2\omega_\theta = (\nabla\times\underline{u})_\theta = \frac{1}{u_m\cos\phi}\left[u_m^2\left(\frac{\sin\phi}{u_m}\frac{\partial u_m}{\partial m} - \frac{\cos\phi}{r_m}\right) - \frac{1}{2}\frac{\partial u_m^2}{\partial r}\right]$$

$$\text{(c)}\quad 2\omega_z = (\nabla\times\underline{u})_z = \frac{1}{r}\left[\frac{\partial r u_\theta}{\partial r} - \sin\phi\frac{\partial r u_\theta}{\partial m} - u_m\cos\phi\frac{\partial \phi}{\partial \theta}\right] \qquad (4.2.1\text{-}15)$$

where r_m is the radius of curvature of the meridional streamline projection, defined by

$$\frac{1}{r_m} = -\frac{\partial\phi}{\partial m} \qquad (4.2.1\text{-}16)$$

Further application of axisymmetricity to Eqn. (4.2.1-15) yields

$$\text{(a)}\quad 2\omega_r = (\nabla\times\underline{u})_r = \frac{\tan\phi}{r}\left(\frac{\partial r u_\theta}{\partial r}\right)$$

$$\text{(b)}\quad 2\omega_\theta = (\nabla\times\underline{u})_\theta = \frac{1}{u_m\cos\phi}\left[u_m^2\left(\frac{\sin\phi}{u_m}\frac{\partial u_m}{\partial m} - \frac{\cos\phi}{r_m}\right) - \frac{1}{2}\frac{\partial u_m^2}{\partial r}\right]$$

$$\text{(c)}\quad 2\omega_z = \frac{1}{r}\left(\frac{\partial r u_\theta}{\partial r}\right) \qquad (4.2.1\text{-}17)$$

where the following relations

$$\frac{\partial\phi}{\partial\theta} = 0$$

$$\frac{\partial r u_\theta}{\partial m} = \frac{\partial u_m}{\partial\theta} = 0$$

have been applied.

The right-hand side of Eqn. (4.2.1-8) becomes ∇H under the assumption of adiabatic process for the fluid to go through the pump channel.

Now, it is ready to write the r-component of Eqn. (4.2.1-8) for the meridional flow velocity and the final form is shown after some rearrangements:

$$\frac{\partial u_m^2}{\partial r} + 2\left(-\frac{\sin\phi}{u_m}\frac{\partial u_m}{\partial m} + \frac{\cos\phi}{r_m}\right)u_m^2 = 2\left(\frac{\partial H}{\partial r} - \frac{u_\theta}{r}\frac{\partial r u_\theta}{\partial r}\right) \qquad (4.2.1\text{-}18)$$

Eqn. (4.2.1-18) can be written as

$$\frac{\partial u_m^2}{\partial r} + P(r)\,u_m^2 = T(r) \qquad (4.2.1\text{-}19)$$

where

$$P(r) = 2\left(-\frac{\sin\phi}{u_m}\frac{\partial u_m}{\partial m} + \frac{\cos\phi}{r_m}\right)$$

$$T(r) = 2\left(\frac{\partial H}{\partial r} - \frac{u_\theta}{r}\frac{\partial r u_\theta}{\partial r}\right) \qquad (4.2.1\text{-}20)$$

In Eqns. (4.2.1-18) – (4.2.1-20) the first term of P(r), i.e., $\sin\phi/u_m \bullet \partial m_m/\partial m$, will provide some difficulty in numerical computations since it is related to the derivatives with respect to "m". The basic philosophy of the streamline curvature method is to express the meridional velocity in terms of "r" and "r-derivatives" so that u_m can be solved in the direction of r only. This feature will be of advantage in numerical computations since the derivatives with respect to "m" are not needed and thus the m-directional control points do not have to be taken in fine increments. Fortunately, $\sin\phi/u_m \bullet \partial u_m/\partial m$ can be expressed in terms of r by using the continuity equation.

$$\frac{\partial u_z}{\partial z} + \frac{1}{r}\frac{\partial r u_r}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} = 0 \ .$$

Again, applying the axisymmetric assumption and Eqns. (4.2.1-11), (4.2.1-12) and (4.2.1-16), the following relation is obtained

$$\frac{\sin\phi}{u_m} \cdot \frac{\partial u_m}{\partial m} = -\left[\frac{\sin^2\phi}{r}\left(1 + \frac{r}{r_m \cos\phi}\right) + \tan\phi \cdot \frac{\partial \phi}{\partial r}\right] \qquad (4.2.1\text{-}21)$$

The basic equation for the streamline curvature method, i.e., Eqn. (4.2.1-18), is expressed in terms of "r" except for the radius of curvature, r_m, so that it can be readily solved numerically. The only problem remaining is that $u_m(r)$ cannot be uniquely determined. This problem can be resolved by applying the mass conservation equation

$$2\pi \int_{r_h}^{r_s} K_b \cdot \rho r \ u_m(r) \ \cos\phi_q \ dr = \dot{G} \qquad (4.2.1\text{-}22)$$

where K_b is the blockage factor due to the blade displacement thickness as well as that of the boundary layer, and ϕ_q is the angle between the line of integration (called "q-line" hereafter) and the line normal to the streamline. With this, the mathematical formulation for the streamline curvature method (SCM) is completed. In what follows, the numerical solution method for SCM will be described in detail.

4.2.2 Solution Method

Since Eqns. (4.2.1-18) and (4.2.1-22) are highly nonlinear for u_m, only an iterative procedure depending on numerical analysis is a possible solution method. First of all, it is assumed that the distribution of upstream flow velocity is known as function of r. Figure 4-11 shows a sample flow configuration on an underwater vehicle tail cone where the

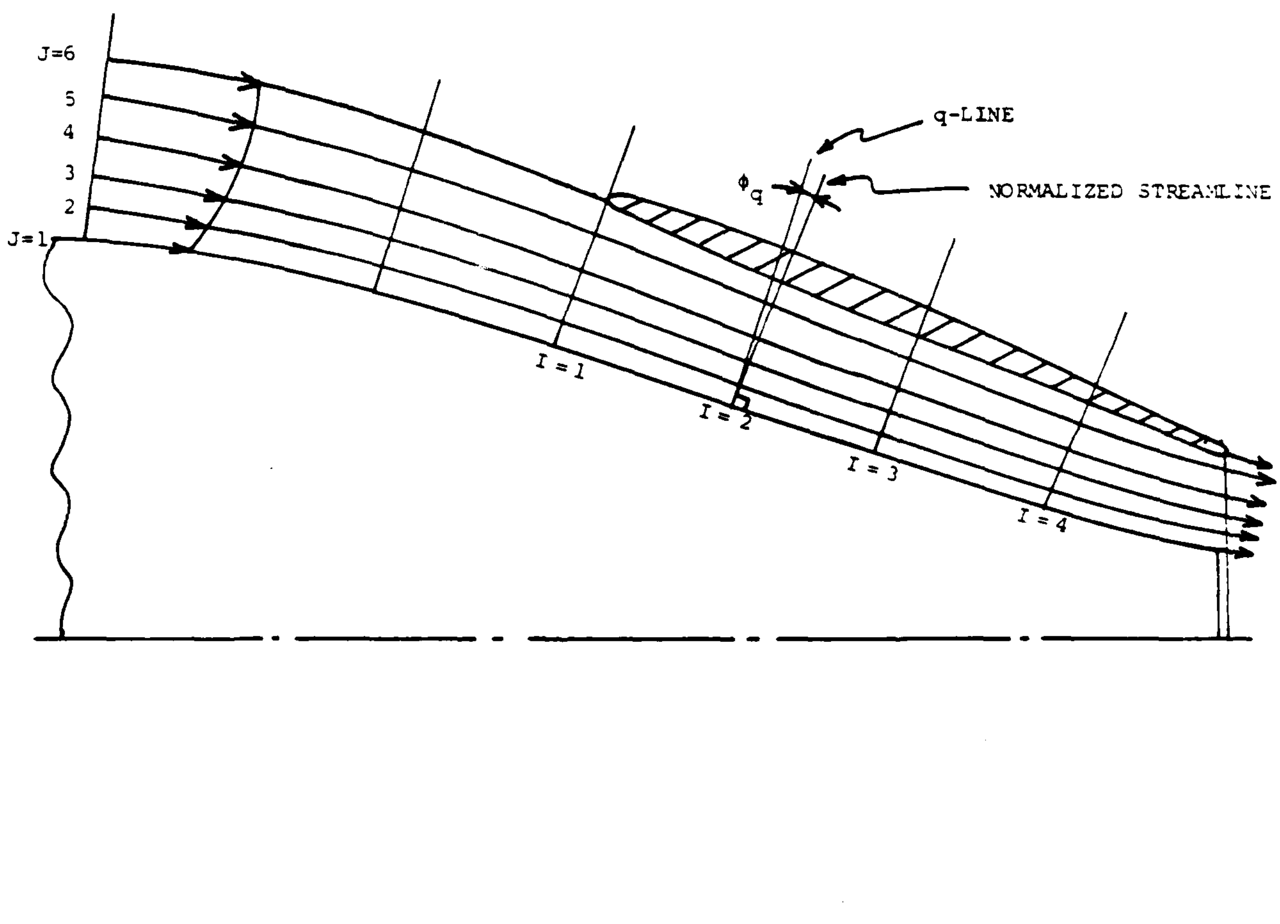


FIGURE 4-11. A SCHEMATIC FLOW DIAGRAM USED FOR NUMERICAL COMPUTATIONS ON STREAMLINE CURVATURE METHOD

964-8-20
080888

upstream location in this case is identified by station 1 (I = 1). The upstream flow velocity can either be uniform or nonuniform*. Station 1 is then divided into a finite number of control points including the hub and the inside wall of the shroud. In Figure 4-11, a total of 6 control points (J = 6) are used. By using the mass conservation equation (4.2.1-22), the local mass flow rate ($g_{J,J+1}$) between each two adjacent control points J and J+1 is calculated. The total mass flow rate, $\dot{G}_1$, is just the summation of local flow rates.

$$\dot{G}_1 = \sum \dot{g}_{1,J,J+1} \qquad (4.2.2\text{-}1)$$

where $$\dot{g}_{1,J,J+1} = 2\pi \int_{r_j}^{r_{j+1}} K_b \cdot \rho r\, u_{m_1}(r) \cos\phi_q\, dr \qquad (4.2.2\text{-}2)$$

For Stations I = 2,3,4..., the initial control points and initial u_m velocity profile are also needed for the iteration procedure. For the case of handling a uniform upstream flow velocity, the selection of control points, J = 1 ~ 6, and determination of initial flow velocity may be done in the same manner as that for the first Station, I = 1, since the constant velocity distribution can be assumed. However, the case of nonuniform flow velocity distribution will require a little care for selection of control points and determination of initial flow velocity distribution. Among many possible ways, it has been decided herein that the velocity distribution is assumed to have a similarity nature as that of the upstream at I = 1, i.e.,

* Note: However, the upstream flow, which is severely retarded or highly nonuniform due to, e.g., the viscous effect, may present problems of accuracy, which will be discussed in Section 4.2.3.

$$u_{mI}(r) = k_I u_{m1}(r)$$

$$r = r_{H1} + \frac{r - r_{H1}}{r_{S1} - r_{H1}} \cdot (r_{S1} - r_{H1})$$

$$; I + 2,3.... \qquad (4.2.2\text{-}3)$$

where

r_{S1}, r_{S1} = radius of shroud internal wall at Station 1 and I (≥ 2), respectively

r_{H1}, r_{H1} = radius of hub at Station 1 and I (≥ 2), respectively

k_I = an arbitrary constant, dependent of Station I, to be determined later.

An arbitrary constant k_I is used to adjust the total flow rate at Station $I \geq 2$ becomes $\dot{G}_1$ when $u_{mI}(r)$ is substituted into Eqn. (4.2.1-22). Once k_I is properly determined, the control points J = 2,3... can be determined by using Eqn. (4.2.2-2). In order to carry out the above computations, ϕ should be known in advance. If the control points at every control station are known, ϕ can be calculated by connecting these points for each streamline by using, e.g., cubic spline method. However, at the first iteration, even these points are not yet known. Therefore, ϕ should be determined by guess. One possible way is to interpolate linearly ϕ for $2 \leq J \leq 5$ from ϕ at J = 1 and ϕ at J = 6, i.e., the hub wall angle and shroud internal wall angle, respectively.

It is now ready to calculate a new set of u_m's or $\partial u_m/\partial r$ at I = 2,3,4... by using Eqn. (4.2.1-18). Since actual calculations are made on $\partial u_m/\partial r$, an integration constant should be determined to uniquely determine u_m itself. This constant can be readily determined by applying the mass conservation equation (4.2.1-22), the control points (J = 2,3...) at each $I \geq 2$ should be shifted according to Eqn. (4.2.2-2). Integral limits, r_J's, which determine the control points on the q-line for J = 2,3... are determined one by one starting from the hub in such a way that the mass flow rate $\dot{g}_{1,J,J+1}$ remains the same in each stream sheet as that for I = 1. This iteration process must be repeated until convergence for u_m's as well as the location of control points (or streamlines) is obtained.

4.2.3 Potential Problems of SCM for Highly Nonuniform Velocity Profile Due to Viscosity

The present streamline curvature method (SCM) to be used for determining the meridional flow streamlines is based on the momentum equation and mass conservation equation with viscous effects totally ignored. Therefore, if the upstream flow is the one fully retarded due to the viscous effect, i.e., boundary layer flow, so that the velocity distribution is highly nonlinear, the application of the present SCM may create substantial inaccuracy in determining the location and velocity of streamlines. This point is clearly understood by investigating the momentum equation (4.2.1-19); P(r) and T(r) are only dependent upon r except for r_m which is a function of curvature of streamline. It means that u_m is a weak function of the axial-direction coordinate so that u_m at a certain station is almost entirely determined by the inner and outer wall curvature. No matter how strongly the incoming flow velocity is retarded, the flow velocity will become more or less uniform before the flow travels too far downstream because the curvature effect (r_m) cannot last too long.

The above discussions seem to suggest that the development of streamline curvature method (SCM) with viscous effect incorporated may be in order, particularly for handling the highly viscous flow near the tail cone area of underwater vehicle.

The momentum equation for such a flow should be of the form

$$\underline{u} \times 2\underline{\omega} = \nabla H - T\nabla S - \upsilon \nabla^2 \underline{u} \qquad (4.2.3\text{-}1)$$

instead of Eqn. (4.2.1-8).

One additional term will make the problem extremely complex and this problem was handled in the FY-86 GHR program.

4.2.4 Numerical Results

The streamline curvature method described in Sections 4.2.1 and 4.2.2 was used to calculate the streamlines for a typical underwater vehicle tail cone area with a shroud. In the present case the upstream flow velocity was assumed to be uniform. A total of 14 q-lines (I = 14) were used with 5 control points (J = 5) at each q-line, see Figure 4-12. The solid lines are the initial guess for the streamlines whereas the dashed lines are the converged solution for the final streamlines. It is seen from this figure that these two sets of lines match well to each other except for the area behind the middle chord of rotor. It means that the initial guess used here was very accurate until the flow passes the rotor and stator. Due to the initial accurate guess for streamlines, computer time was minimal.

4.3 BLADE-TO-BLADE FLOW

4.3.1 Transformation

Under the assumption that an axisymmetric stream surface exists in a rotating machine, from the conservation equation of circulation, i.e., $\underline{\nabla} \times \underline{w} + 2\underline{\omega} = \underline{0}$, the following relation is obtained for the relative flow,

$$\frac{\partial w_m}{\partial \theta} - \frac{\partial (r w_\theta)}{\partial m} = 2\omega r \frac{\partial r}{\partial m} \qquad (4.3\text{-}1)$$

where w_m and w_θ are relative flow velocities in the direction of m and θ, see Figure 4-13. The continuity equation for the same stream surface is also written

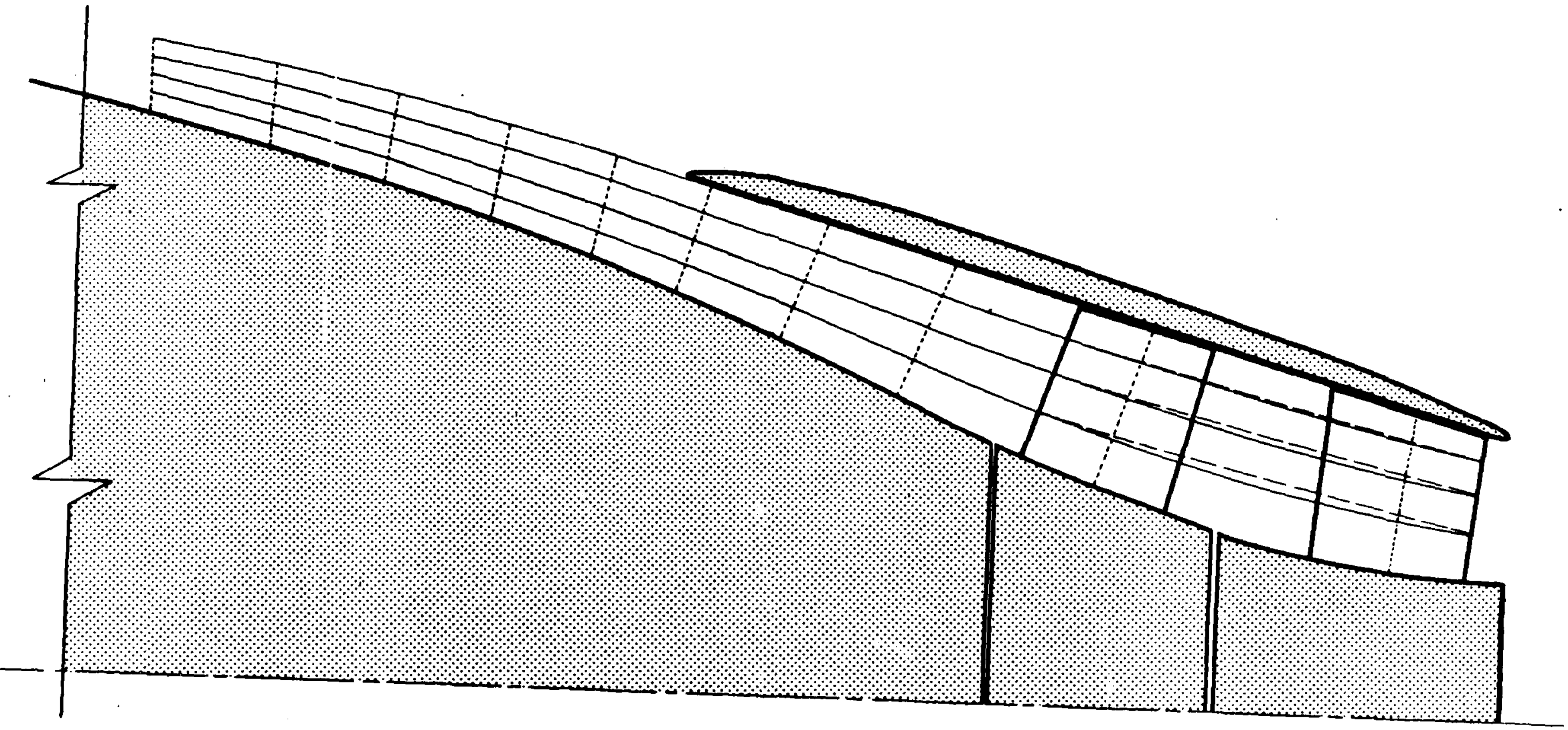


FIGURE 4-12. CALCULATED RESULTS OF STREAMLINE CURVATURE METHOD FOR A TYPICAL UNDERWATER VEHICLE TAIL CONE WITH SHROUD WHERE THE SOLID LINES ARE OF THE INITIAL GUESS AND DASHED LINES ARE THE CONVERGED SOLUTION (THE DOTTED LINES ARE Q-LINES USED FOR THE PRESENT COMPUTATION)

964-8-21
080888

$$\frac{\partial\left(b\rho w_\theta\right)}{\partial\theta} + \frac{\partial\left(b\rho r w_m\right)}{\partial m} = 0 \qquad (4.3\text{-}2)$$

where b is the thickness of stream surface.

Then, a stream function ψ can be defined by

$$w_\theta = \frac{1}{b\rho}\frac{\partial\psi}{\partial m}, \quad w_m = -\frac{1}{b\rho}\frac{\partial\psi}{r\partial\theta} \qquad (4.3\text{-}3)$$

Substitution of w_θ and w_m in Eqn. (4.3-3) into Eqn. (4.3-1) yields

$$\frac{\partial^2\psi}{r^2\partial\theta^2} + \frac{\partial^2\psi}{\partial m^2} + \left(\frac{1}{r}\frac{\partial r}{\partial m} - \frac{1}{b}\frac{\partial b}{\partial m}\right)\frac{\partial\psi}{\partial m} = -2b\rho\omega\,\sin\lambda \qquad (4.3\text{-}4)$$

where λ is the angle of the line tangent to the stream surface at the point of interest made with the axis of rotation, see Figure 4-13.

This three-dimensional axisymmetric stream surface can be mapped onto a two-dimensional plane, (X, Y), see Figure 4-14, by the following mapping functions

$$\frac{dX}{dm} = \frac{r_o}{r}, \quad \frac{dY}{d\theta} = -r_o\,, \qquad (4.3\text{-}5)$$

where r_o is an arbitrary constant which is used for the purpose of scaling between the physical coordinate space and mapped plane (X, Y). The governing equation (4.3-4) can now be written in the (X, Y) coordinate system by using Eqn. (4.3-5)

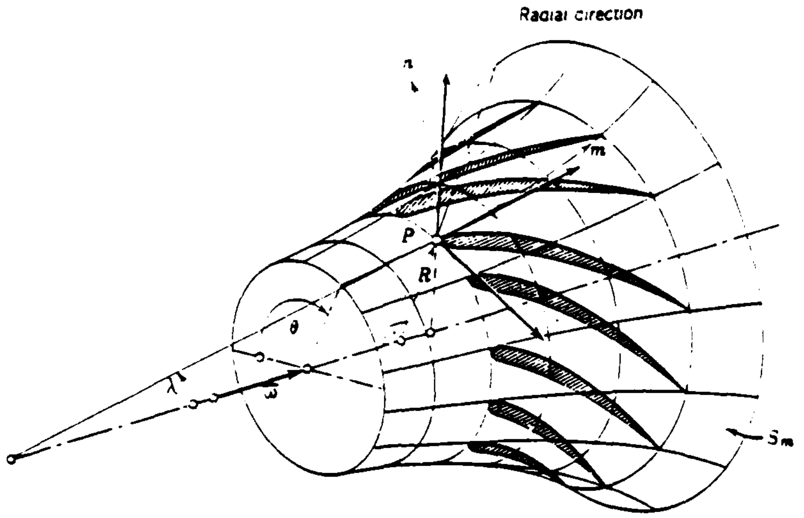


FIGURE 4-13. AXISYMMETRIC STREAM SURFACE

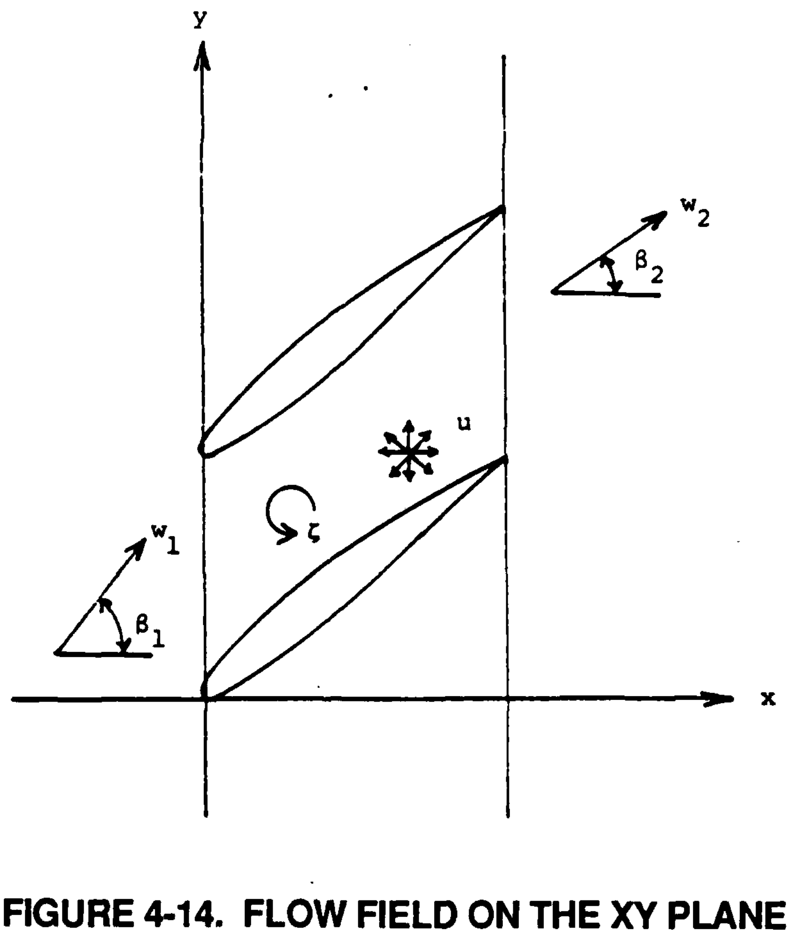


FIGURE 4-14. FLOW FIELD ON THE XY PLANE

964-8-22
080888

$$\nabla^2 \psi = -2b\rho\omega \left(\frac{r}{r_o}\right)^2 \sin\lambda$$

$$+ \frac{1}{b\rho} \left\{ \frac{\partial(b\rho)}{\partial X} \frac{\partial \psi}{\partial X} + \frac{\partial(b\rho)}{\partial Y} \frac{\partial \psi}{\partial Y} \right\} . \qquad (4.3\text{-}6)$$

Also, the relative velocities in the X- and Y-directions are given

$$\text{(a)} \quad w_X = \frac{1}{b\rho} \frac{\partial \psi}{\partial Y} = \frac{r}{r_o} w_m$$

$$\text{(b)} \quad w_Y = -\frac{1}{b\rho} \frac{\partial \psi}{\partial X} = -\frac{r}{r_o} w_\theta . \qquad (4.3\text{-}7)$$

As seen from Eqn. 4.3-6, the governing equation for the (X, Y) plane is now a Poisson equation instead of the Laplace equation, which exists only for a flow on the perfectly cylindrical stream surface. Therefore, the results obtained from the two-dimensional linear cascade theory should be corrected according to the right-hand side term of Eqn. (4.3-6). It is readily understood that these right-hand side terms are satisfied by distributing the following vortices and sources on the entire (X, Y) plane

$$\text{(a)} \quad \zeta = (\nabla \times \underline{w})_{X,Y} = 2\omega \left(\frac{r}{r_o}\right)^2 \sin\lambda$$

$$\text{(b)} \quad \mu = (\nabla \cdot \underline{w})_{X,Y} = -\frac{1}{(b\rho)^2} \left\{ \frac{\partial(b\rho)}{\partial X} \frac{\partial \psi}{\partial Y} - \frac{\partial(b\rho)}{\partial Y} \frac{\partial \psi}{\partial X} \right\} . \qquad (4.3\text{-}8)$$

By adding the induced velocities calculated from ζ and μ, the blade profile shape or equivalently the camber obtained in the conventional two-dimensional analysis will be corrected. It should be noted that the first term on the right-hand side of Eqn. (4.3-6) arises from non-zero λ, i.e., the stream surface is not parallel to the axis of rotation, whereas the second group of terms is due to the non-uniform thickness of stream surface or tube. Needless to say, if $\lambda = 0$ and bρ is constant, Eqn. (4.3-6) becomes a Laplace equation and thus a two-dimensional linear cascade theory holds.

A method similar to the present one was developed by Inoue and his colleague (e.g., the paper by Inoue, et al. (1980)). In this paper there exist a few major drawbacks, some of which could potentially lead to a substantial error in the final design. First of all, since they use a two-dimensional linearized cascade theory, the error becomes significant for high solidity and high stagger angle area, i.e., near the hub, although they introduce experimental data in a later step of the analysis. Secondly, their velocity triangle used for determining the incoming flow angle to the blade is in error of the first order since they did not take into consideration the effect of non-cylindrical and variable thickness stream surface. Finally, due to the use of the linearized cascade theory, they failed to obtain the velocity distribution so that a boundary layer analysis and cavitation inception analysis are not possible.

4.3.2 Linearized Cascade Theory

A two-dimensional cascade theory is implemented here to be used for calculating the lift coefficient for the three-dimensional flow which will be discussed in Section 4.4. The boundary condition (Eqn. 4.3.2-16) has to be modified, as will be discussed in Section 4.4, when this cascade theory is applied in the quasi three-dimensional flow.

The lift coefficient is determined for any given cascade geometry which is specified by the solidity (blade chord-gap ratio, c/s) and the stagger angle, λ (Figure 4-15). Symbols used in Mellor (1959) were followed; use L_i and C_{Li} in denoting the ideal lift force and lift coefficient when the drag is zero. Then (see also Weinig, 1964)

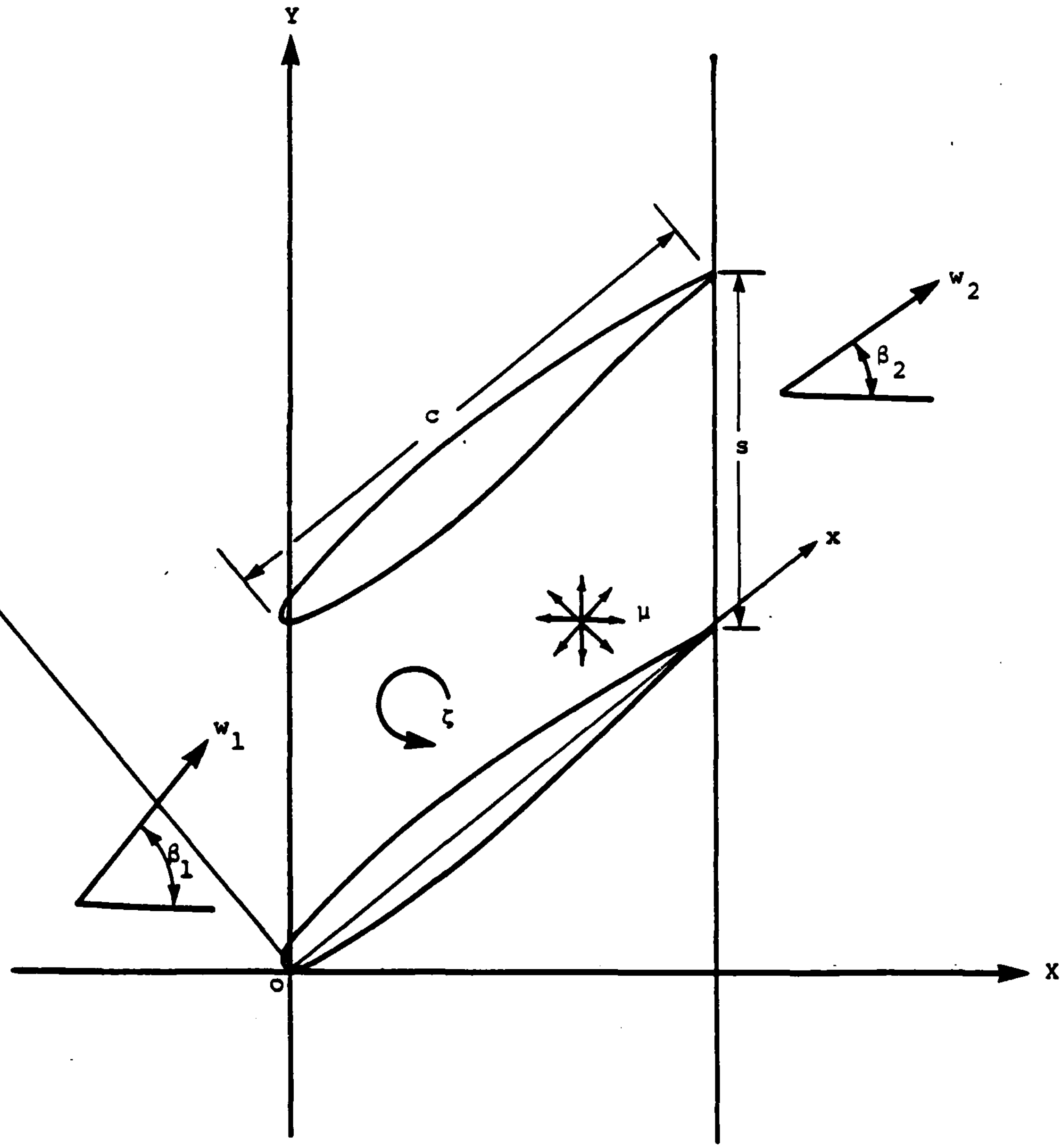


FIGURE 4-15. DEFINITION DIAGRAM

964-8-23
080888

$$L_i = (s/c)\,\rho W_m \Delta V_\theta \tag{4.3.2-1}$$

$$\text{and} \quad C_{L_i} = (L_i/c) \,/\, \left(\frac{1}{2}\rho W_m^2\right)$$

$$= 2(s/c)\,(\Delta V_\theta / W_m) \tag{4.3.2-2}$$

where s is the blade pitch, c is the chord length, ρ is the fluid density, W_m is the mean relative velocity, and ΔV_θ is the difference between the peripheral velocity at the exit and that at the inlet.

Replacing the product $s\Delta V_\theta$ by the line integral $\oint \bar{v} \cdot dr$ on a closed path comprising two streamlines s distance apart and joined by two lines parallel to the θ-direction, we have (Weinig, 1964)

$$C_{L_i} = 2\Gamma/(cW_m) \tag{4.3.2-3}$$

$$\text{where} \quad \Gamma = \oint \bar{v} \cdot dr \tag{4.3.2-4}$$

is the circulation around a profile (Wislicenus, 1965).

The camber is assumed to be sufficiently small so that the chord length is substantially equal to the distance measured along the camber line. Then, the circulation around a thin wing profile is given by (Abbott and von Doenhoff, 1959)

$$\Gamma = \int_0^c \gamma \, dx \tag{4.3.2-5}$$

where γ is the difference in velocity between the suction and pressure surfaces, which is also the strength of the vortex sheet comprising the blade camber line (von Karman and Burgers, 1963). Therefore, equation (4.3.2-3) is expressed by

$$C_{L_i} = 2\int_0^1 \left(\gamma/W_m\right) d(x/c) \qquad (4.3.2\text{-}6)$$

The cambered blade is built up by superimposing vortices on the camber line and a distribution of sources and sinks on the camber line to account for the profile thickness effects. The distribution of source (sink), q, is (Mellor, 1959)

$$q = W_m\, dy_t / dx + d(uy_t) / dx \qquad (4.3.2\text{-}7)$$

where the thickness of blade is denoted by y_t and the induced chord-wise velocity, u, is considered constant along the y-direction within the profile. The second term can be shown to be negligible (Mellor, 1959) and so we have

$$q/W = dy_t / dx$$

$$= (t/c)\, f_t\, (x/c) \qquad (4.3.2\text{-}8)$$

where

$$f_t'(x) = \partial f_t / \partial x \qquad (4.3.2\text{-}9)$$

and the thickness function f_t is defined by

$$y_t/C = (t/c)\, f_t\, (x/c) \qquad (4.3.2\text{-}10)$$

where t is the maximum thickness of the blade.

The camber function f_c is defined by

$$y_c/C = C_b\, f_c\, (x/c) \qquad (4.3.2\text{-}11)$$

where y_c denotes the camber distribution and C_b is defined by

$$C_b = 2\int_0^{\pi} (dy_c/dx) \cos\theta \, d\theta \tag{4.3.2-12}$$

in which

$$\cos\theta = 1 - 2\,x/c \tag{4.3.2-13}$$

A blade is approached by a mean velocity W_m at a mean angle α_m. To satisfy the condition that the normal velocity vanishes at the boundary, the flow velocity, together with the induced velocity, should be tangent to the surfaces. Neglecting the thickness effect, the boundary condition at x_o becomes

$$\begin{aligned}(W_m \sin\alpha_m + v_o) / (W_m \cos\alpha_m + u_o) &= (dy_c / dx)_o \\ &= C_b\, f_c'(x_o/c)\end{aligned} \tag{4.3.2-14}$$

where u_o and v_o denote respectively the x- and y-components of the induced velocity at x_o on the 0^{th} blade with x measured along the chord from the leading edge.

To find the lift coefficient by equation (4.3.2-6), we assume that γ/W_m may be represented by a trignometric series as (Abbott and von Doenhoff, 1959)

$$\gamma/W_m = 2A_o\,(1 + \cos\theta)/\sin\theta + 4\sum_{n=1}^{\infty} A_n \sin n\theta \tag{4.3.2-15}$$

which is zero at the trailing edge of $\theta = \pi$ so that the Kutta condition is satisfied. This distribution of vortices, together with the distribution of sources/sinks shown in equation (4.3.2-8), may be used to obtain the componenets of induced velocity, u_o and v_o. The components of induced velocity are then substituted into equation (4.3.2-14). With the aid of (von Karman and Burgers, 1963; Milne-Thomson, 1966)

$$\int_{o} \left[\cos n\theta / (\cos \theta - \cos \theta_o)\right] d\theta = \pi \sin n\theta_o / \theta_o \quad (n = 0, 1, 2, \ldots) \qquad (4.3.2\text{-}16)$$

the following equation is obtained:

$$\sum_{n=0}^{\infty} A_n g_n = \sin\alpha_m - C_b f_c'(\theta_o) \cos\alpha_m + C_b \sum_{n=0}^{\infty} A_n h_n - \frac{t}{c} \left(C_b B - T\right) \qquad (4.3.2\text{-}17)$$

where g_n, h_n, B, and T are defined in Mellor (1959) except that $f_t'(\theta_o)$ should be replaced by $f_t'(\theta)$ in defining T. A_n *are the Fourier coefficients to be evaluated.*

For a cascade of certain solidity, incident angle, and stagger angle, equation (4.3.2-17) is used to calculate N coefficients, A_n, based on the camber and thickness at N locations along the chord. This study follows the solution method of Mellor (1959) which greatly reduces the calculation labor when a set of solutions as functions of the solidity and stagger angle is desired.

Equation (4.3.2-17) is multiplied by cos $k\theta_o$ and integrated from 0 to π to obtain an equation for numerical integration.

Having the double Fourier integral functions calculated, the cascade coefficients, A_n, were computed. Then the lift coefficient was obtained from

$$C_{L_i} = 2\pi(A_0 + A_1) \tag{4.3.2-18}$$

which is derived by inserting equation (4.3.2-15) to equation (4.3.2-6).

4.3.3 Data Analysis

The NACA 65-series experimental data given by Herrig, et al. (1951) are used to do the multiple regression analysis in the present study. The data include the cascade lift coefficient and the design angle of attack. The design angle of attack is a function of solidity and blade camber (Figure 4-16). The lift coefficient is a function of stagger angle, solidity, blade camber, and angle of attack. Among the lift coefficient values, only those associated with the design angle of attack are used in the data analysis.

Based on 28 data points in Figure 4-16, the fourth order polynomial equation of the design angle of attack, α_d, obtained by the multiple regression analysis is

$$\begin{aligned} \alpha_d = & -1.78681\ \sigma^4 + 0.51975\ \sigma^3 C_b + 0.02078\ \sigma^2 C_b^2 \\ & -0.06408\ \sigma C_b^3 + 0.01716\ C_b^4 + 7.68063\ \sigma^3 \\ & -2.50213\ \sigma^2 C_b + 0.30280\ \sigma C_b^2 + 0.01298\ C_b^3 \\ & -12.95580\ \sigma^2 + 6.54778\ \sigma C_b - 0.30496\ C_b^2 \\ & +12.87888\ \sigma + 2.78086\ C_b - 2.22656 \end{aligned} \tag{4.3-19}$$

where σ denotes solidity and C_b represents the camber. Figure 4-17 shows the correlation between the calculated value from Eqn. (4.3-19) and the experimental data. The mean residual is -0.00002, and the standard deviation is 0.04, and the maximum residual is 0.09 degrees which is smaller than the error bound of the original data.

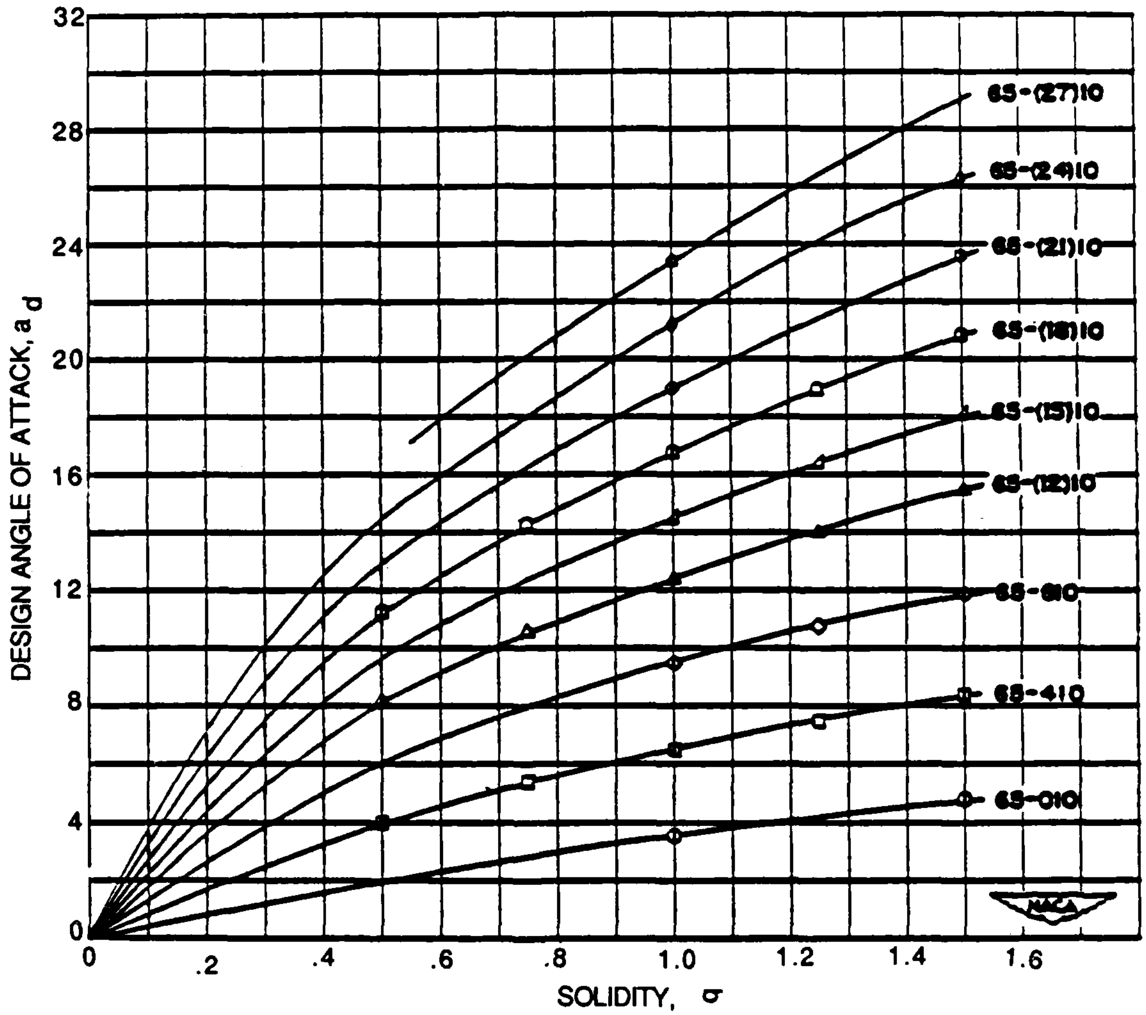


FIGURE 4-16. VARIATION OF DESIGN ANGLE OF ATTACK WITH SOLIDITY FOR THE SECTIONS TESTED

964-8-24
080888

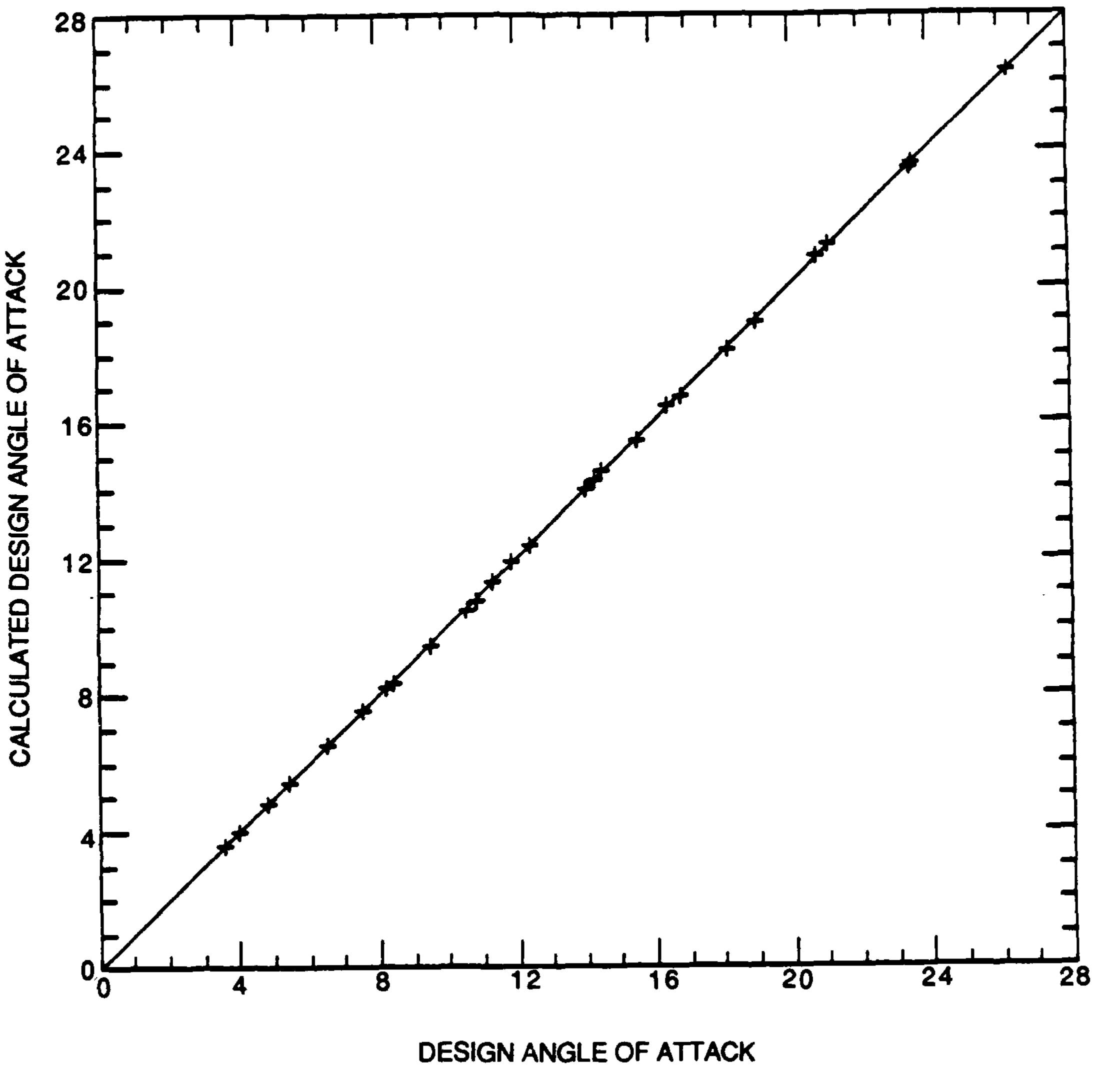


FIGURE 4-17. COMPARISON OF DESIGN ANGLE OF ATTACK OBTAINED FROM THE MULTIPLE REGRESSION ANALYSIS AND THOSE OBTAINED FROM THE LABORATORY

964-8-25
080888

Cascade lift coefficient at the design angle of attack has 79 data as a function of stagger angle, solidity, and camber in Herrig, et al. (1951). Two data at the falling limb, in the figure of lift coefficient vs. angle of attack, are removed from the sample. A total of 77 data is used in the multiple regression analysis. The resultant fourth order polynomial equation is

$$
\begin{aligned}
C_b = {} & 0.00000038594\ \beta^4 - 1.66604\ \sigma^4 - 2.74752\ C_l^4 \\
& -0.000050739\ \beta^3\sigma - 4.42872\ \sigma^3 C_l - 0.093633\ C_l^3\beta \\
& +0.0013187\ \beta^2\sigma^2 - 10.87314\ \sigma^2 C_l^2 - 0.0016033\ C_l^2\beta^2 \\
& -0.0014369\ \beta\sigma^3 - 9.83274\ \sigma C_l^3 - 0.000035308\ C_l\beta^3 \\
& -0.000020018\ \beta^3 + 8.26201\ \sigma^3 + 20.91537\ C_l^3 \\
& +0.0040173\ \beta^2\sigma + 26.19466\ \sigma^2 C_l + 0.33569\ C_l^2\beta \\
& -0.11946\ \beta\sigma^2 + 41.23668\ \sigma C_l^2 + 0.0059786\ C_l\beta^2 \\
& -0.00080978\ \beta^2 - 14.15108\ \sigma^2 - 45.84094\ C_l^2 \\
& -0.010265\ \beta\sigma - 47.03403\ \sigma C_l - 0.39504\ C_l\beta \\
& +0.015917\ \beta + 13.59117\ \sigma + 36.29869\ C_l \\
& -5.23047
\end{aligned}
\qquad (4.3\text{-}20)
$$

The results calculated from this equation are shown as solid lines in Figures 4-18 to 4-21 for different relative flow angle at the blade inlet. Also shown in these figures are discrete data at different conditions. The results from the regression analysis fit well with the data.

4.4 THREE-DIMENSIONAL ANALYSIS – BLADE-TO-BLADE FLOW

This section illustrates the theory and procedure to solve the blade-to-blade flow on each stream surface in an axisymmetric three-dimensional flow environment.

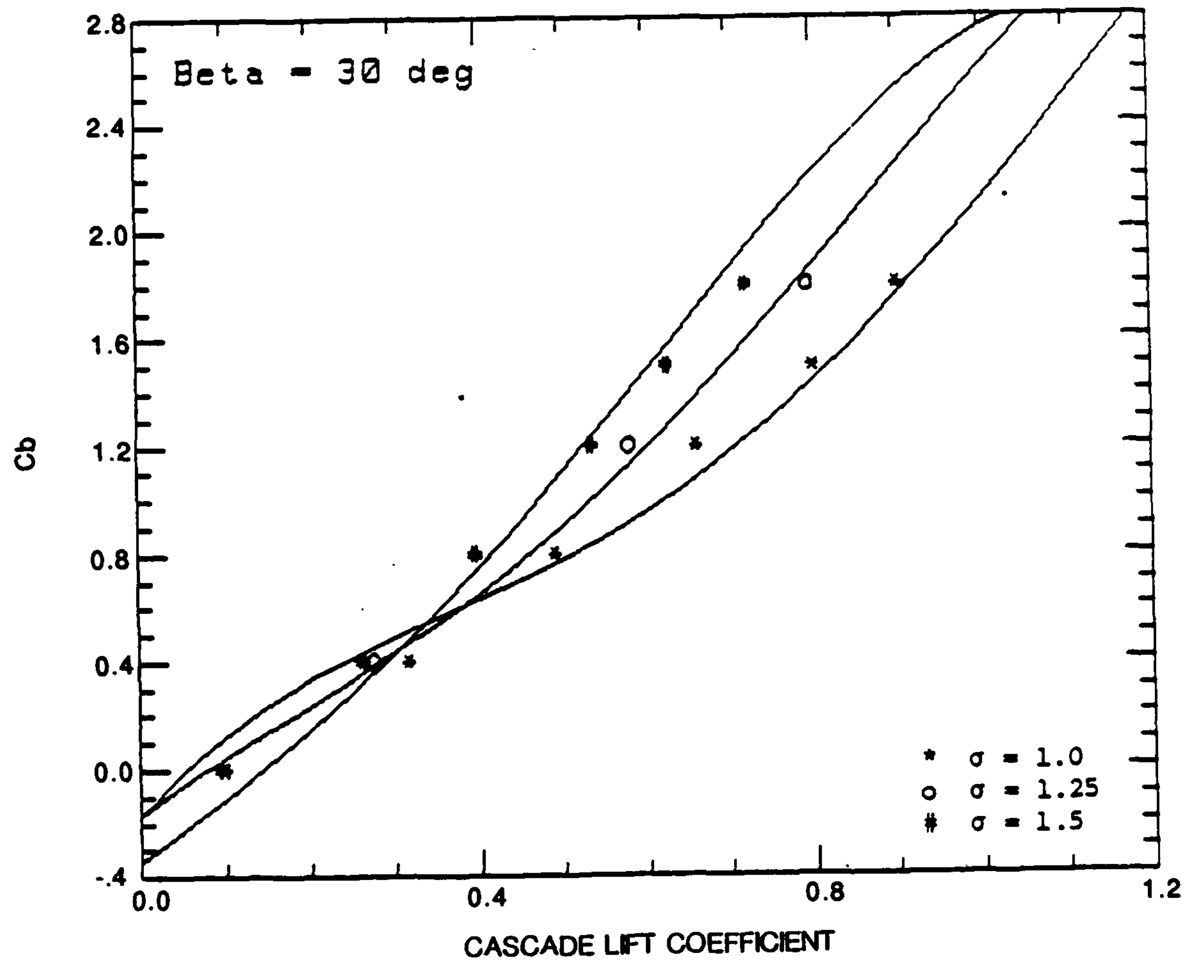


FIGURE 4-18. CAMBER AS A FUNCTION OF CASCADE LIFT COEFFICIENT AND SOLIDITY OBTAINED FROM REGRESSION ANALYSIS DATA (SOLID LINE) COMPARED WITH ORIGINAL DATA (DISCRETE DATA POINT), FOR $\beta_1 = 30°$

964-8-26
080888

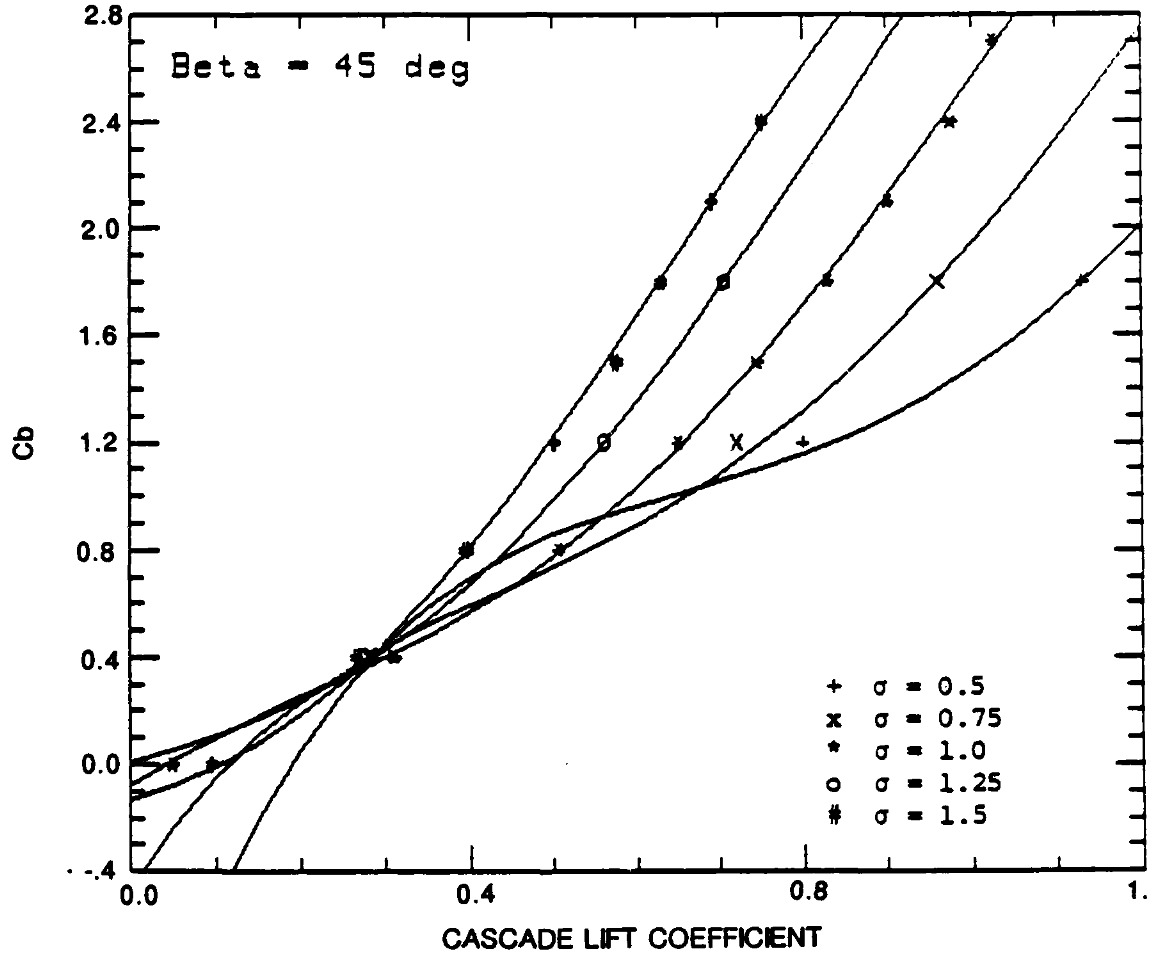


FIGURE 4-19. CAMBER AS A FUNCTION OF CASCADE LIFT COEFFICIENT AND SOLIDITY OBTAINED FROM REGRESSION ANALYSIS DATA (SOLID LINE) COMPARED WITH ORIGINAL DATA (DISCRETE DATA POINT), FOR $\beta_1 = 45°$

964-8-27
080888

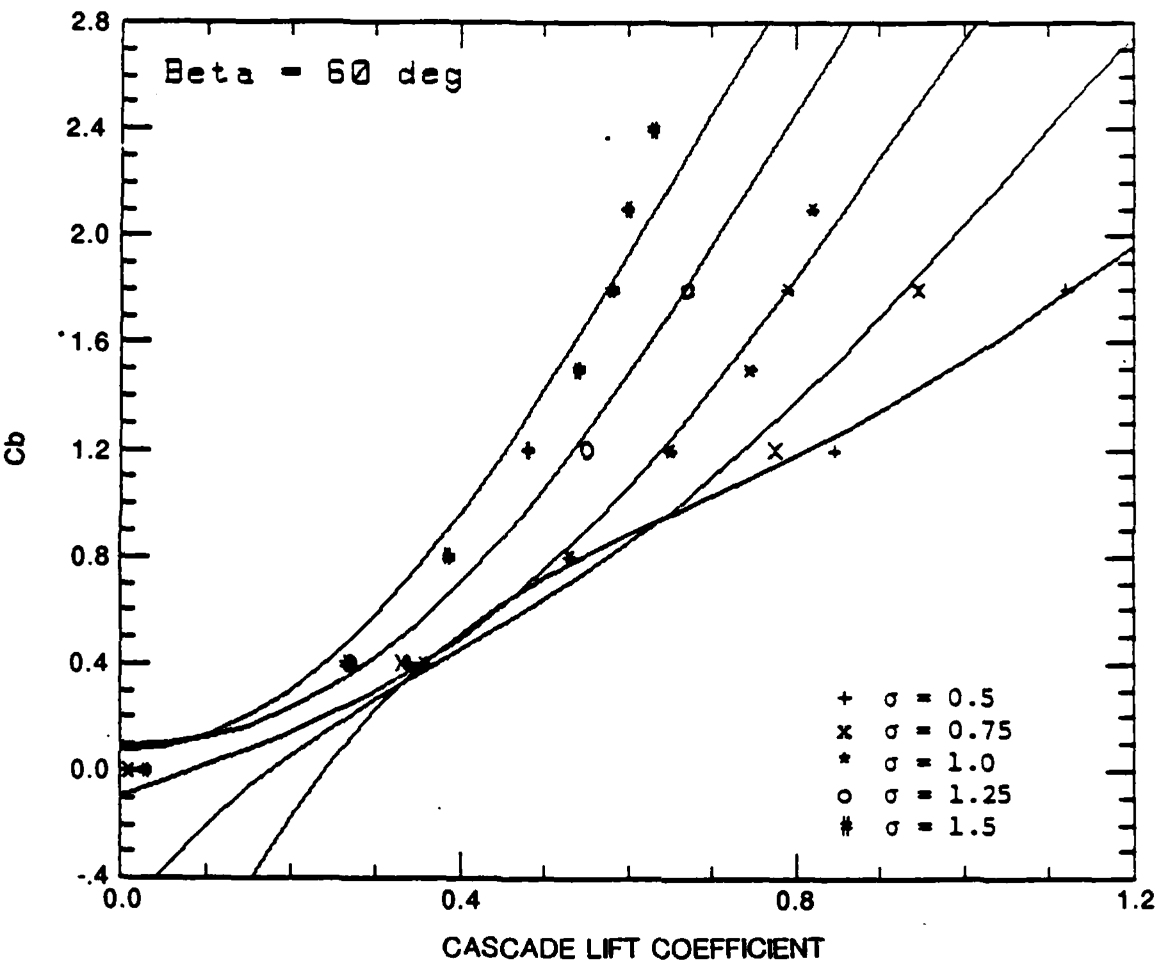


FIGURE 4-20. CAMBER AS A FUNCTION OF CASCADE LIFT COEFFICIENT AND SOLIDITY OBTAINED FROM REGRESSION ANALYSIS DATA (SOLID LINE) COMPARED WITH ORIGINAL DATA (DISCRETE DATA POINT), FOR $\beta_1 = 60°$

964-8-28
080888

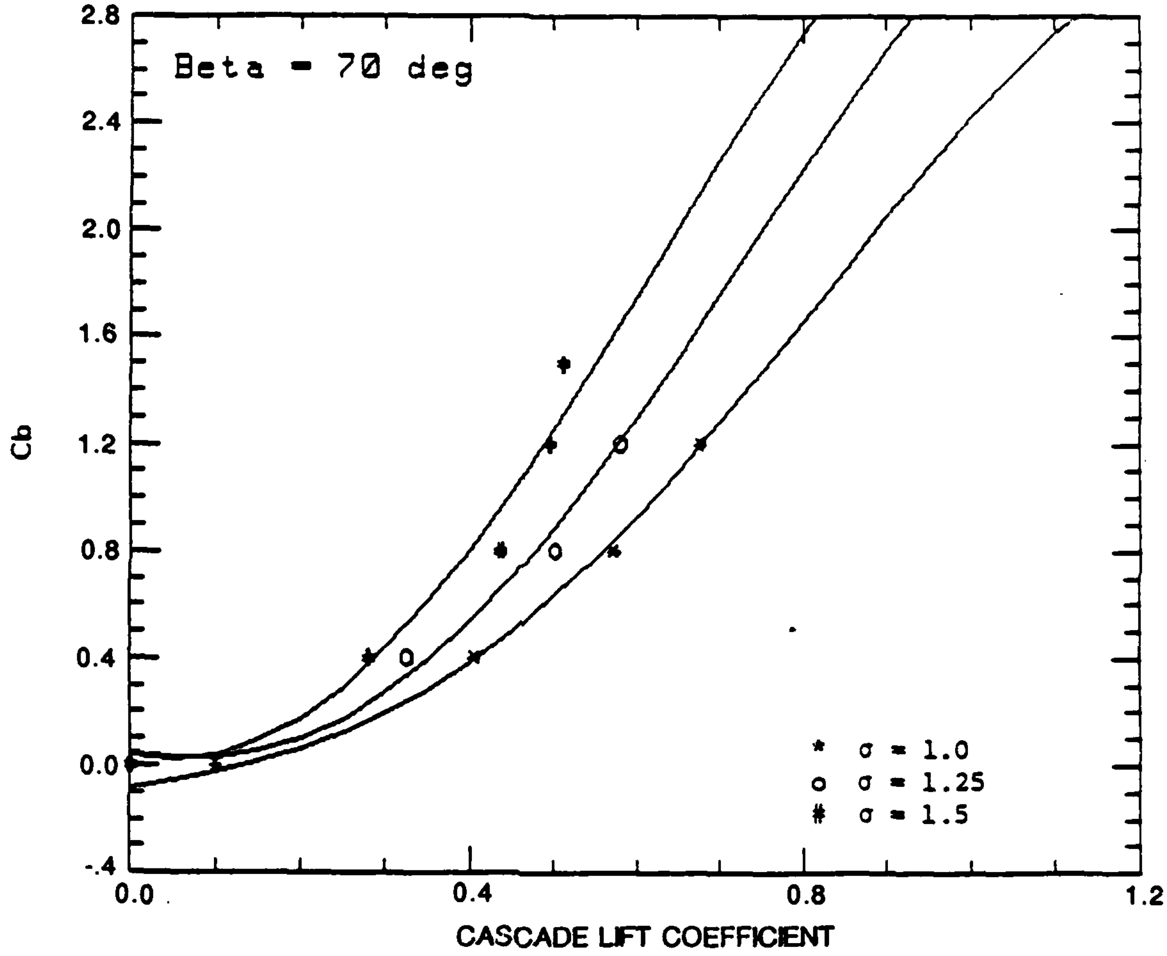


FIGURE 4-21. CAMBER AS A FUNCTION OF CASCADE LIFT COEFFICIENT AND SOLIDITY OBTAINED FROM REGRESSION ANALYSIS DATA (SOLID LINE) COMPARED WITH ORIGINAL DATA (DISCRETE DATA POINT), FOR $\beta_1 = 70°$

964-8-29
080888

4.4.1 Differential Equations

Under the assumption that an axisymmetric stream surface exists in a rotating machine, from the conservation equation of steady circulation, i.e., $\nabla \times \underline{w} + 2\underline{\omega} = \underline{0}$, the following relation is obtained for the relative flow,

$$\frac{\partial w_m}{\partial \theta} - \frac{\partial (r w_\theta)}{\partial m} = 2\omega r \frac{\partial r}{\partial m} = 2\omega r \sin\lambda' \qquad (4.4\text{-}1)$$

where w_m and w_θ are relative flow velocities in the directions of m and θ, r measures the radial distance, and λ' is the angle of the line tangent to the stream surface at the point of interest made with the axis of rotation (Figure 4-22). The continuity equation for the same stream surface is also written

$$\frac{\partial (b\rho w_\theta)}{\partial \theta} + \frac{\partial (b\rho r w_m)}{\partial m} = 0 \qquad (4.4\text{-}2)$$

where b is the thickness of stream surface and ρ is the fluid density.

Then, a stream function ψ can be defined by

$$w_\theta = \frac{1}{b\rho}\frac{\partial \psi}{\partial m}, \quad w_m = -\frac{1}{b\rho}\frac{\partial \psi}{r\partial \theta} \qquad (4.4\text{-}3)$$

Substitution of w_θ and w_m in Eqn. (4.4.3) into Eqn. (4.4.1) yields

$$\frac{\partial^2 \psi}{r^2 \partial \theta^2} + \frac{\partial^2 \psi}{\partial m^2} + \frac{1}{r}\frac{\partial r}{\partial m} - \frac{1}{b}\frac{\partial b}{\partial m}\frac{\partial \psi}{\partial m} = -2b\rho\omega \sin\lambda' \qquad (4.4\text{-}4)$$

4.4.2 Transformation

Consider a Cartesian coordinate system (X, Y) with the origin 0 at the leading edge of a blade and the X-axis in the axial direction (Figure 4-15), the three-dimensional axisymmetric stream surface given by Eqn. (4.4-4) can be mapped onto this two-dimensional X-Y plane by the following mapping functions

$$\frac{dX}{dm} = \frac{r_o}{r}, \quad \frac{dY}{d\theta} = -r_o \, , \tag{4.4-5}$$

where r_o is an arbitrary constant which is used for the purpose of scaling between the physical coordinate space and mapped plane (X, Y). By using Eqn. (4.4-5), the governing equation (4.4-4) can now be written in the (X, Y) coordinate system as

$$\nabla^2 \psi = -2b\rho\omega \left(\frac{r}{r_o}\right)^2 \sin\lambda + \frac{1}{b\rho}\left\{\frac{\partial(b\rho)}{\partial X}\frac{\partial\psi}{\partial X} + \frac{\partial(b\rho)}{\partial Y}\frac{\partial\psi}{\partial Y}\right\} \tag{4.4-6}$$

Also, the relative velocities in the X- and Y-directions are given by

$$\text{(a)} \quad w_X = \frac{1}{b\rho}\frac{\partial\psi}{\partial Y} = \frac{r}{r_o} w_m = \frac{r}{r_o} c_m$$

$$\text{(b)} \quad w_Y = -\frac{1}{b\rho}\frac{\partial\psi}{\partial X} = -\frac{r}{r_o} w_\theta = -\frac{r}{r_o}(c_\theta - u) \tag{4.4-7}$$

where c_m and c_θ are absolute flow velocities along m and θ directions, respectively.

On the mapped plane, the relative flow angles at the inlet and exit of a blade, β_1 and β_2, respectively, are obtained from the equivalent velocity diagram to be

$$\tan \beta_1 = \frac{w_{Y1}}{w_{X\infty}} \tag{4.4-8}$$

$$\tan \beta_2 = \frac{w_{Y2}}{w_{X\infty}} \tag{4.4-9}$$

where the subscripts 1 and 2 denote the condition at the inlet and exit, respectively, of a blade, and $w_{X\infty}$ is the mean value of w_{X1} and w_{X2}.

It is important to note that the three-dimensional flow configuration is now transformed to the two-dimensional X-Y plane where all the benefits of the 2-D cascade theory may be used. Only the problem of using such a transformation is that there exits no 2-D cascade flow pattern as shown in Figure 4-22. In order to make the 2-D cascade theory useful for the present analayis, an equivalent flow diagram should be generated by violating the continuity equation as shown in Figure 4-22, by chain lines. The flow stagger angles β_1 and β_2 are defined by this equivalent flow diagram and used throughout the 2-D cascade analysis.

4.4.3 Effects of Streamline Inclination and Meridian Velocity Variation

As seen from Eqn. 4.4-6, the governing equation for the (X, Y) plane is now a Poisson equation instead of the Laplace equation which exists only for a flow on a perfectly cylindrical stream surface with uniform velocity distribution. Therefore, the results obtained from the two-dimensional linear cascade theory should be corrected according to the right-hand side term of Eqn. (4.4-6). It is readily understood that these right-hand side terms are satisfied by distributing the following vortices, ζ, and sources, μ, on the entire (X, Y) plane

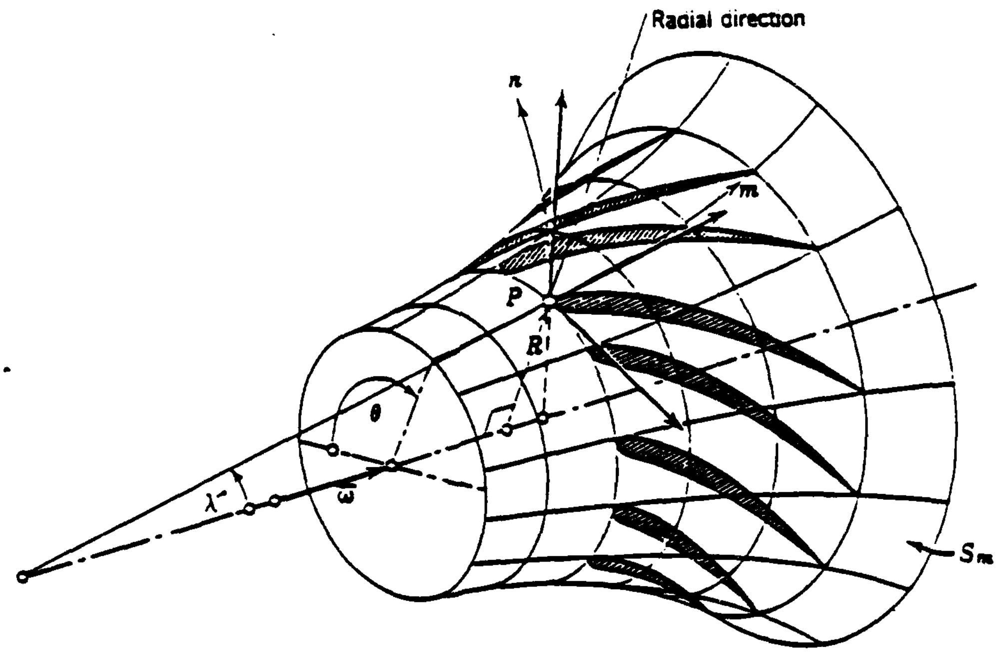


FIGURE 4-22. AXISYMMETRIC STREAM SURFACE

964-8-30
080888

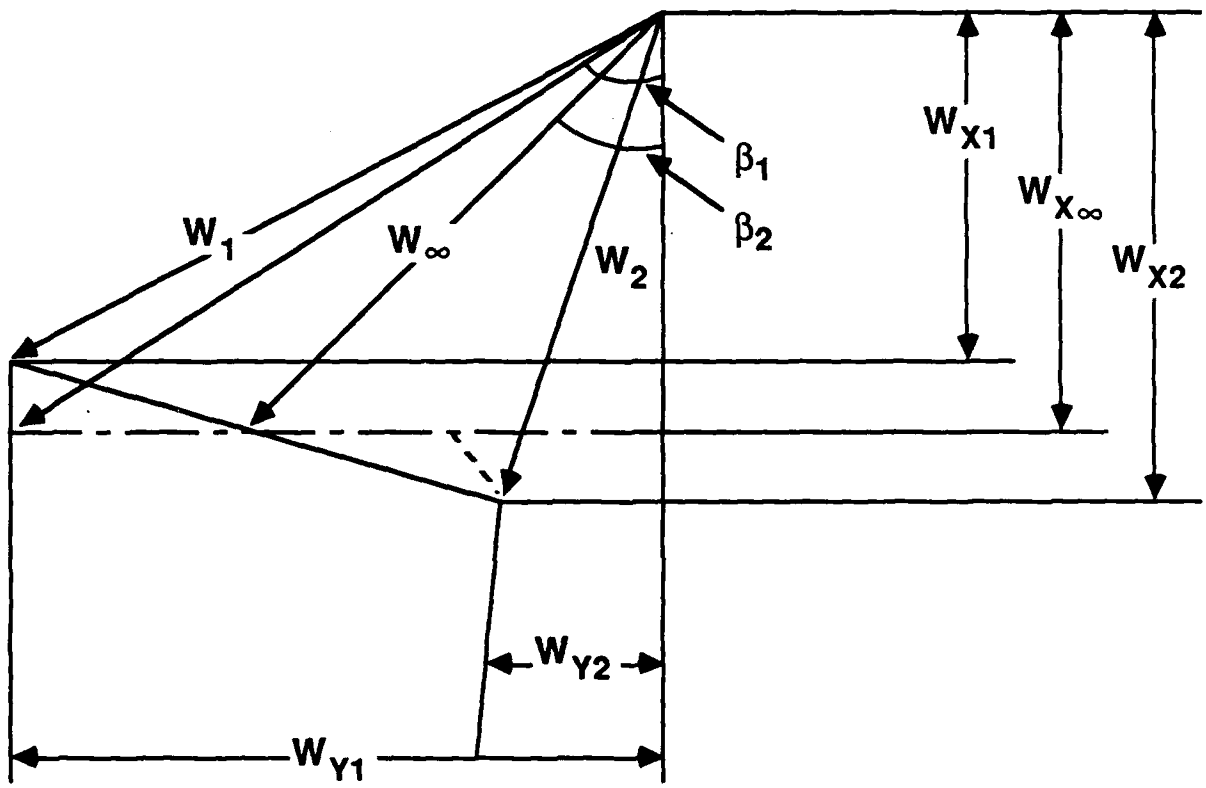


FIGURE 4-22a

$$\text{(a)} \quad \zeta = (\nabla \times \underline{w})_{X,Y} = 2\omega\left(\frac{r}{r_o}\right)^2 \sin\lambda'$$

$$\text{(b)} \quad \mu = (\nabla \cdot \underline{w})_{X,Y} = -\frac{1}{(b\rho)^2}\left\{\frac{\partial(b\rho)}{\partial X}\frac{\partial\psi}{\partial Y} - \frac{\partial(b\rho)}{\partial Y}\frac{\partial\psi}{\partial X}\right\} . \qquad (4.4\text{-}10)$$

By adding the induced velocities calculated from ζ and μ, the blade profile shape or equivalently the camber obtained in the conventional two-dimensional analysis will be corrected. It should be noted that the first term on the right-hand side of Eqn. (4.4-6) arises from non-zero λ', i.e., the stream surface is not parallel to the axis of rotation, whereas the second group of terms is due to the non-uniform thickness of stream surface or tube caused by the variation of meridional velocity. Needless to say, if $\lambda' = 0$ and $b\rho$ is constant, Eqn. (4.4-6) becomes a Laplace equation and thus a two-dimensional linear cascade theory holds.

A method similar to the present one was developed by Inoue and his colleague. In their study (e.g., Inoue, et al., 1980), there exist a few major drawbacks, some of which could potentially lead to a substantial error in the final design. First of all, since they use a two-dimensional linearized cascade theory, the error becomes significant for high solidity and high stagger angle area, i.e., near the hub, although they introduce experimental data in a later step of the analysis. Secondly, their velocity triangle used for determining the incoming flow angle to the blade is in error of the first order since they did not take into consideration the effect of non-cylindrical and variable thickness stream surface. Finally, due to the use of the linearized cascade theory, they failed to obtain the velocity distribution and therefore a boundary layer analysis and cavitation inception analysis are not possible.

With these aspects in mind, effort has been made in the current GHR project to improve the accuracy of the linear cascade theory as well as to avoid the singular behavior of velocity at the leading edge of blade. Detailed discussions on the loading correction and leading edge correction have been presented in the FY '85 Report.

4.4.4 Induced Velocities

If the inclination of stream surface is small such as that in an axial-flow case, an approximation solution of the velocity induced by the distributed vortices (Eqn. 4.4-10a) are obtained by a replaced average vorticity $\bar{\zeta}$ (Inoue, et al., 1979):

$$\bar{\zeta} = \frac{1}{c\cos\lambda}\int_{0}^{c\cos\lambda} \zeta dX$$

$$= \frac{u_o}{c\cos\lambda}\,\frac{r_2^{\,2} - r_1^{\,2}}{r_o^{\,2}} \qquad (4.4\text{-}11)$$

where c is the chord length, u_o is the speed of blade at the reference radius r_o, and subscripts 1 and 2 indicate the inlet and exit, respectively, of the blade.

Similarly, when the variation of axial velocity is small, the distribution of sources (Eqn. 4.4-10b) is replaced by a uniform distribution:

$$\bar{\mu} = \frac{\bar{w}_{X2} - \bar{w}_{X1}}{c\cos\lambda}$$

$$= \frac{\Delta w_X}{c\cos\lambda} \qquad (4.4\text{-}12)$$

where $\bar{w}$ denotes the average relative velocity.

Consider another Cartesian coordinate system (x, y) with the origin at 0 and the x-axis in the chordwise direction which has a stagger angle λ relative to the axial direction (Figure 4-23). The mean flow velocity along the chord direction is

$$\begin{aligned} w_{x\infty} &= w_{X\infty} \cos\lambda + w_{Y\infty} \sin\lambda \\ &= w_{X\infty} \cos\lambda \, (1 + \tan\beta_\infty \tan\lambda) \quad . \end{aligned} \tag{4.4-13}$$

The induced velocities, relative to $w_{x\infty}$, due to the uniform distribution of vortices $\bar{\zeta}$, and sources, $\bar{\mu}$, are

$$\frac{v_{\zeta x}}{w_{x\infty}} = \chi\left(\frac{x}{c} - \frac{y}{c}\tan\lambda\right)\frac{\tan\lambda}{1 + \tan\lambda\tan\beta_\infty} \tag{4.4-14}$$

$$\frac{v_{\zeta y}}{w_{x\infty}} = \chi\left(\frac{x}{c} - \frac{y}{c}\tan\lambda\right)\frac{1}{1 + \tan\lambda\tan\beta_\infty} \tag{4.4-15}$$

$$\frac{v_{\mu x}}{w_{x\infty}} = \xi\left(\frac{x}{c} - \frac{y}{c}\tan\lambda\right)\frac{1}{1 + \tan\lambda\tan\beta_\infty} \tag{4.4-16}$$

$$\frac{v_{\mu y}}{w_{x\infty}} = -\xi\left(\frac{x}{c} - \frac{y}{c}\tan\lambda\right)\frac{\tan\lambda}{1 + \tan\lambda\tan\beta_\infty} \tag{4.4-17}$$

where

$$\begin{aligned} \chi &= \frac{\bar{\zeta}\, c \cos\lambda}{w_{X\infty}} \\ &= \frac{1}{\Phi}\,\frac{r_2^{\,2} - r_1^{\,2}}{r_o^{\,2}} \end{aligned} \tag{4.4-18}$$

and

$$\xi = \frac{\bar{\mu}\, c \cos\lambda}{w_{X\infty}}$$

$$= \frac{1}{\Phi} \frac{r_2 \overline{w}_{m2} - r_i \overline{w}_{mt}}{r_o u_o} \qquad (4.4\text{-}19)$$

are streamline inclination parameter and axial velocity variation parameter, respectively, and

$$\Phi = \frac{r_1 \overline{w}_{m1} - r_2 \overline{w}_{m2}}{2 r_o u_o}$$

$$= \frac{w_{X\infty}}{u_o} \qquad (4.4\text{-}20)$$

is a local flow coefficient, with subscript m denoting the meridional component of the velocity.

Eqns. (4.4-14) and (4.4-15) are good if the streamline inclination is small. Eqns. (4-4-16) and (4.4-17) are obtained by ignoring the blockage *effect of blade thickness. In the* following discussion, the blockage effect is considered and the induced velocity from the distributed vortices given by Eqn. (4.4-10a) are obtained by solving the Poisson equation

$$\nabla^2 \psi = -2b\rho\omega \left(\frac{r}{r_o}\right)^2 \sin\lambda \qquad (4.4\text{-}21)$$

to give the solution

$$v_{\zeta Y} = u_o \left[\left(\frac{r}{r_o}\right)^2 - \frac{r_1^2 + r_2^2}{2 r_o^2}\right] \qquad (4.4\text{-}22)$$

where u_o is a reference velocity.

The induced velocity due to the distribution of sources given by Eqn. (4.4-10b) is approximated by (Inoue, et al., 1980)

$$v_{\mu X} = \frac{1}{K_b}\left(\frac{b_1 \rho_1}{b\rho}\right) - 1 \; w_{X1} - \frac{1}{2}\left(w_{X2} - w_{X1}\right) \qquad (4.4\text{-}23)$$

with the blockage factor of blade thickness, K_b, put into the consideration.

By decomposing both $v_{\zeta Y}$ and $v_{\mu X}$ into the x and y directions, Eqns. (4.4-22) and (4.4.23), together with (4.4-13), become

$$\frac{v_{\zeta x}}{w_{x\infty}} = \frac{u_o}{w_{x\infty}}\left[\left(\frac{r}{r_o}\right)^2 - \frac{r_1^{\,2} + r_2^{\,2}}{2\,r_o^{\,2}}\right]\frac{\tan\lambda}{1 + \tan\beta_\infty \tan\lambda} \qquad (4.4\text{-}24)$$

$$\frac{v_{\zeta y}}{w_{x\infty}} = \frac{v_{\zeta x}}{w_{x\infty}} \,/\, \tan\lambda \qquad (4.4\text{-}25)$$

$$\frac{v_{\mu x}}{w_{x\infty}} = \left[\frac{1}{K_b}\left(\frac{b_1 \rho_1}{b\rho} - 1\right)\frac{w_{X1}}{w_{X\infty}} - \frac{w_{X2} - w_{X1}}{2 w_{X\infty}}\right]\frac{1}{1 + \tan\beta_\infty \tan\lambda} \qquad (4.4\text{-}26)$$

and

$$\frac{v_{\mu y}}{w_{x\infty}} = \frac{v_{\mu x}}{w_{x\infty}} \tan\lambda \;. \qquad (4.4\text{-}27)$$

4.4.5 Boundary Condition

The flow approaches the blade by a velocity $v_{x\infty}$ at an angle β_∞ relative to the axis of symmetry. This velocity, together with flow velocities induced by distributed vortices and sources, should satisfy the following condition of flow tangency:

$$\frac{dy_c}{dx} = \frac{w_{y\infty} + w_{Iy} + v_{\zeta y} + v_{\mu y}}{w_{x\infty} + w_{Ix} + v_{\zeta x} + v_{\mu x}} \qquad (4.4\text{-}28)$$

where y_c denotes the y coordinate of the camber line, and w_{lx} and w_{ly}, respectively, are the x and y components of velocities induced by bound vortices and sources along the chord.

4.4.6 Flow Skewness in Diagonal Contracting Channel

The major goal of the proposed theory is to make corrections of flow stream tube nonuniformness and diagonal flow effects on the two-dimensional axial cascade flow. This is performed within the framework of perturbation method by uniformly distributing vortices, $\bar{\gamma}$, and source/sink, $\bar{\zeta}$, in cascade row, as described in Section 4.4-4. The blade profile shape will be properly modified in order to take $\bar{\gamma}$ and $\bar{\zeta}$ into account. Due to the distribution of uniform vortices and source/sink, the flow skewness problem exists, i.e., the first order effect on the flow incidence angle is generated by such singularity distribution.

As shown in Figure 4-23, the constant vortices, $\bar{\zeta}$, *are distributed in a strip over the entire* cascade row. The complex potential W for a point vortex $\bar{\zeta}$ placed at z' is expressed

$$W = \frac{i\bar{\zeta}}{2\pi} \ln (z-z') \tag{4.4-29}$$

and the induced velocities are obtained by taking the deviations of W

$$\dot{u}_\zeta - i\dot{v}_\zeta = \frac{dW}{d\zeta} = \frac{i\bar{\zeta}}{2\pi} \frac{x-x' + iy'}{(x-x')^2 + y'^2} . \tag{4.4-30}$$

The induced velocities due to the uniform distribution of vortices over a strip of cascade region are obtained by integration

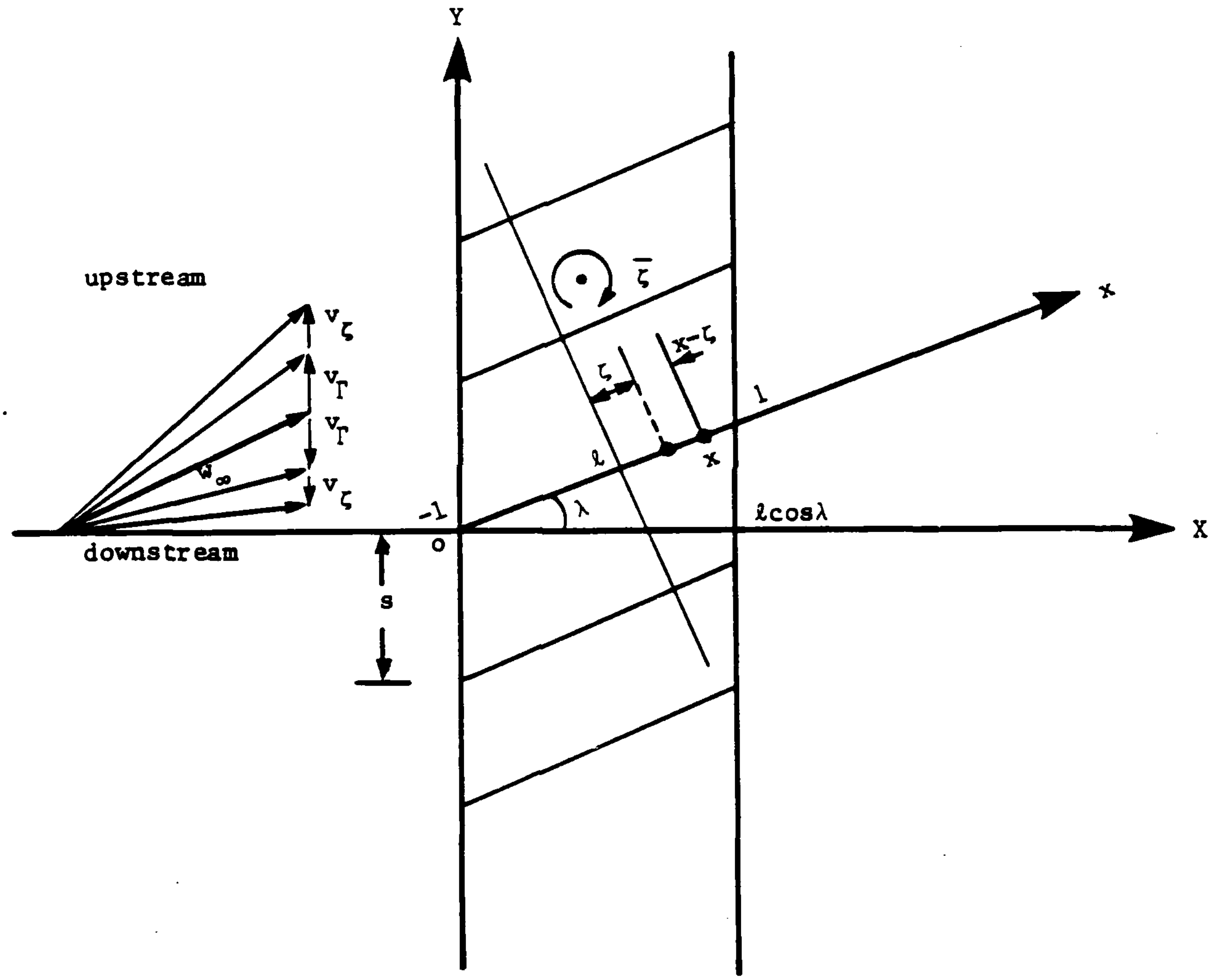


FIGURE 4-23. BLADE SETTING IN THE MAPPED PLANE

$$v_\zeta = \int_0^{l\cos\lambda} \int_{-\infty}^{\infty} \dot{v}_\zeta \, dx' \, dy'$$

$$= \int_0^{l\cos\lambda} \int_{-\infty}^{\infty} \frac{-\bar{\zeta}}{2\pi} \cdot \frac{x-x'}{(x-x')^2 + y'^2} \, dx' \, dy'$$

$$= \frac{-\bar{\zeta}}{\pi} \int_0^{l\cos\lambda} \tan^{-1} \frac{y'}{x-x'} \int_0^{\infty} dx'$$

As $x \rightarrow -\infty$,

$$v_\zeta = \frac{-\bar{\zeta}}{\pi} \int_0^{l\cos\lambda} \left(-\frac{\pi}{2} - 0\right) dx' = \frac{\bar{\zeta}}{2} \, l\cos\lambda \, , \tag{4.4-31}$$

whereas $x \rightarrow +\infty$

$$v_\zeta = -\frac{\bar{\zeta}}{2} \, l\cos\lambda \, . \tag{4.4-32}$$

Therefore, by distributing $\bar{\zeta}$ uniformly as described above, the upstream and downstream flows are equally deviated from the geometric mean flow w_∞ (see Figure 4-23 again).

On the other hand, a more rigorous mathematical model by Mani and Acosta (1968) will lead to a totally different conclusion. Since this point has been ignored by many researchers, let us review herein the details of Mani and Acosta's work. In their work, the channel contraction was chosen to be of an exponential shape as shown in Figure 4-24. By satisfying the proper boundary conditions, the velocity component in the y-direction due to a point vortex of unit strength at the origin is obtained (see the report of Mani (1966)),

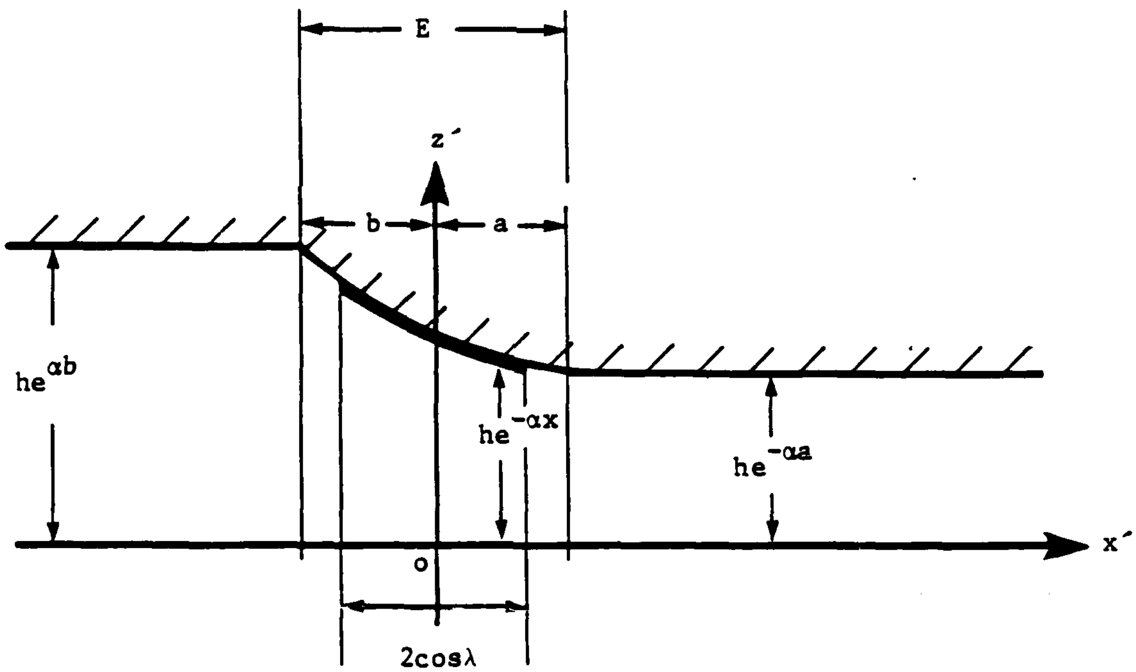


FIGURE 4-24. FLOW CONFIGURATION IN CONTRACTION CHANNEL

$$v' = \begin{cases} \dfrac{1}{2\pi}\left[\dfrac{x'}{x'^2 + y'^2}\left(1 - \dfrac{1}{2}\alpha b\right) + \dfrac{\alpha}{8}\ \ln \dfrac{(2a - x')^2 + y'^2}{x'^2 + y'^2}\right] \\ \qquad + 0\ (\alpha^2) \qquad \text{for } x' \le -b & (4.4\text{-}33) \\ \dfrac{1}{2\pi}\left[\dfrac{x'}{x'^2 + y'^2}\left(1 + \dfrac{1}{2}\alpha a\right) + \dfrac{\alpha}{8}\ \ln \dfrac{(x' + 2b)^2 + y'^2}{x'^2 + y'^2}\right] \\ \qquad + 0\ (\alpha^2) \qquad \text{for } x' \ge a\ . & (4.4\text{-}34) \end{cases}$$

Therefore, the induced velocity due to a distribution of the vortices over a chord length in the cascade configuration is obtained

$$v = \int_{-1}^{1} \gamma(\xi) \sum_{n=-\infty}^{\infty} v'\ d\xi \qquad (4.4\text{-}35)$$

with $x' = (x-\xi)\cos\lambda$

$y' = -ns + (x-\xi)\sin\lambda$

$a = \dfrac{E}{2} - \xi\cos\lambda$

$b = \dfrac{E}{2} + \xi\cos\lambda\ ,$

$$v = \begin{cases} \int_{-1}^{1} \gamma(\xi) \frac{1}{2\pi} \sum_{n=-\infty}^{n=\infty} \left[\left(1 - \frac{1}{2}\alpha b\right) \frac{(x-\xi)\cos\lambda}{(x-\xi)^2 + n^2 s^2 - 2ns(x-\xi)\sin\lambda} \right. \\ \left. + \frac{\alpha}{8} \ln \frac{\left[E - (x+\xi)\cos\lambda\right]^2 + n^2 s^2 - 2ns(x-\xi)\sin\lambda + (s-\xi)^2 \sin^2\lambda}{(x-\xi)^2 + n^2 s^2 - 2ns(x-\xi)\sin\lambda} \right] d\xi \\ \text{as } x \to -\infty \qquad (4.4\text{-}36) \\ \int_{-1}^{1} \gamma(\xi) \frac{1}{2\pi} \sum_{n=-\infty}^{n=\infty} \left[\left(1 - \frac{1}{2}\alpha a\right) \frac{(x-\xi)\cos\lambda}{(x-\xi)^2 + n^2 s^2 - 2ns(x-\xi)\sin\lambda} \right. \\ \left. + l\frac{\alpha}{8} n \frac{\left[(x+\xi)\cos\lambda + E\right]^2 + n^2 s^2 - 2ns(x-\xi)\sin\lambda + (x-\xi)^2 \sin^2\lambda}{(x-\xi)^2 + n^2 s^2 - 2ns(x-\xi)\sin\lambda} \right] d \\ \text{as } x \to +\infty \qquad (4.4\text{-}37) \end{cases}$$

Let's simplify the first term of the above integration by defining

$$P = \sum_{n=-\infty}^{n=\infty} \frac{(x-\xi)\cos\lambda}{(x-\xi)^2 + n^2 s^2 - 2ns\,(x-\xi)\sin\lambda} .$$

Then

$$P = \sum_{n=-\infty}^{\infty} \frac{(x-\xi)\cos\lambda}{\left[(x-\xi) - ns\sin\lambda\right]^2 + (ns\cos\lambda)^2}$$

$$= (x-\xi)\cos\lambda \cdot \sum_{n=-\infty}^{\infty} \frac{1}{(x-\xi) + ins\, e^{i\lambda}} \cdot \frac{1}{(x-\xi) - ins\, e^{-i\lambda}}$$

$$= (x-\xi)\cos\lambda \sum_{n=-\infty}^{n=\infty} \frac{1}{\frac{x-\xi}{e^{i\lambda}} + ins} \cdot \frac{1}{\frac{x-\xi}{e^{-i\lambda}} - ins}$$

$$= \frac{\pi}{2s} \sum_{n=-\infty}^{n=\infty} \left[\frac{1}{\frac{\pi}{e^{i\lambda}} \frac{x-\xi}{s} + in\pi} + \frac{1}{\frac{\pi}{e^{-i\lambda}} \frac{x-\xi}{s} - in\pi} \right]$$

From the identity (see Mellor (1959)),

$$\sum_{n=-\infty}^{n=\infty} \frac{1}{\frac{\pi}{e^{-i\lambda}} \frac{x-\xi}{s} - in\pi} = \coth\left(\pi \frac{x-\xi}{s} e^{i\lambda}\right) ,$$

then,

$$P = \frac{\pi}{2s} \left[-\coth\left(-\pi \frac{x-\xi}{s} e^{-i\lambda}\right) + \coth\left(\pi \frac{x-\xi}{s} e^{i\lambda}\right) \right] \tag{4.4-38}$$

Therefore,

$$P = \begin{cases} -\frac{\pi}{s} & \text{as } x \rightarrow -\infty \\ \frac{\pi}{s} & \text{as } x \rightarrow +\infty \ . \end{cases} \tag{4.4-39}$$

The second terms of (4.4-36) and (4.4-37) are defined, respectively

$$Q = \begin{cases} \sum_{n=-\infty}^{\infty} \ln \frac{\left[E - (x+\xi)\cos\lambda\right]^2 + n^2 s^2 - 2ns\,(x-\xi)\sin\lambda + (x-\xi)^2 \sin^2\lambda}{(x-\xi)^2 + n^2 s^2 - 2ns\,(x-\xi)\sin\lambda} & \text{for } x \leq -1 \\ \sum_{n=-\infty}^{\infty} \ln \frac{\left[(x+\xi)\cos\lambda + E\right]^2 + n^2 s^2 - 2ns\,(x-\xi)\sin\lambda + (x-\xi)^2 \sin^2\lambda}{(x-\xi)^2 + n^2 s^2 - 2ns\,(X-\xi)\sin\lambda} & \text{for } x \geq 1 \end{cases}$$

Then,

$$Q = \begin{cases} \sum_{n=-\infty}^{\infty} \ln\left[1 + \frac{E^2 - 2E\,(x-\xi)\cos\lambda + 4\xi x\cos^2\lambda}{(x-\xi)^2 + n^2s^2 - 2ns\,(x-\xi)\sin\lambda}\right] & , x \le -1 \\ \sum_{n=-\infty}^{\infty} \ln\left[1 + \frac{E^2 + 2E\,(x+\xi)\cos\lambda + 4\xi x\cos^2\lambda}{(x-\xi)^2 + n^2s^2 - 2ns\,(x-\xi)\sin\lambda}\right] & , x \ge 1 \end{cases}$$

or

$$Q = \begin{cases} \sum_{n=-\infty}^{\infty} \frac{E^2 - 2E\,(x+\xi)\cos\lambda + 4\xi x\cos^2\lambda}{(x-\xi)^2 + n^2s^2 - 2ns\,(x-\xi)\sin\lambda}, & \text{as } x \to -\infty \\ \sum_{n=-\infty}^{\infty} \frac{E^2 + 2E\,(x+\xi)\cos\lambda + 4\xi x\cos^2\lambda}{(x-\xi)^2 + n^2s^2 - 2ns\,(x-\xi)\sin\lambda}, & \text{as } x \to +\infty \end{cases} .$$

From (A-11),

$$Q = \begin{cases} -\frac{\pi}{s}\,\frac{E^2 - 2E\,(x+\xi)\cos\lambda + 4\xi x\cos^2\lambda}{(x-\xi)\cos\lambda}, & \text{as } x \to -\infty \\ \frac{\pi}{s}\,\frac{E^2 + 2E\,(x+\xi)\cos\lambda + 4\xi x\cos^2\lambda}{(x-\xi)\cos\lambda}, & \text{as } x \to +\infty \end{cases} ,$$

or

$$Q = \begin{cases} -\frac{\pi}{s}\,(-2E + 4\xi\cos\lambda), & \text{as } x \to -\infty \\ \frac{\pi}{s}\,(2E + 4\xi\cos\lambda), & \text{as } x \to +\infty \end{cases} .$$

Therefore,

$$v = \begin{cases} \int_{-1}^{1} \gamma(\xi) \frac{1}{2\pi} \left[\left(1 - \frac{1}{2} \alpha b\right) \left(-\frac{\pi}{s}\right) + \frac{\alpha}{8} \left(-\frac{\pi}{s}\right) (-2E + 4\xi\cos\lambda) \right] d\xi \\ \qquad , \text{ as } x \to -\infty \\ \int_{-1}^{1} \gamma(\xi) \frac{1}{2\pi} \left[\left(1 + \frac{1}{2} \alpha a\right) \left(\frac{\pi}{s}\right) + \frac{\alpha}{8} \left(\frac{\pi}{s}\right) (2E + 4\xi\cos\lambda) \right] d\xi \\ \qquad , \text{ as } x \to +\infty \end{cases}$$

Assuming a symmetric loading (i.e., $\int_{-1}^{1} \gamma(\xi)\xi d\xi = 0$),

$$v = \begin{cases} \frac{\Gamma}{2s} \left[1 - \frac{1}{2} \alpha \left(b + \frac{E}{2} \right) \right], & \text{as } x \to -\infty \\ \frac{-\Gamma}{2s} \left[1 + \frac{1}{2} \alpha \left(a + \frac{E}{2} \right) \right], & \text{as } x \to +\infty \end{cases}$$

where $\Gamma = \int_{-1}^{1} \gamma(\xi) d\xi$ and $\gamma(\xi)$ is positive in the clockwise direction. If $E = 2a = 2b$,

$$v = \begin{cases} \frac{\Gamma}{2s} \left(1 - \frac{\alpha}{2} E \right), & \text{as } x \to -\infty \\ \frac{-\Gamma}{2s} \left(1 + \frac{\alpha}{2} E \right), & \text{as } x \to +\infty \ . \end{cases} \qquad (4.4\text{-}40)$$

Rewriting this induced velocity in two parts, by v_Γ and v_α,

$$v = \begin{cases} v_\Gamma - v_\alpha , & \text{as } x \to -\infty \\ -v_\Gamma - v_\alpha , & \text{as } x \to +\infty \end{cases}$$

where

$$v_\Gamma = \frac{\Gamma}{2s}, \quad v_\alpha = \frac{\Gamma}{2s} \cdot \frac{\alpha}{2} E \ .$$

The velocity diagram is now written in Figure 4-25. It is clearly seen that there exits an error of first order of α if w_∞ is chosen as the reference velocity. Instead w'_∞ should be chosen as the reference velocity so that the blade setting will be changed. This error is substantial since the three-dimensional, diagonal flow correction handled under the

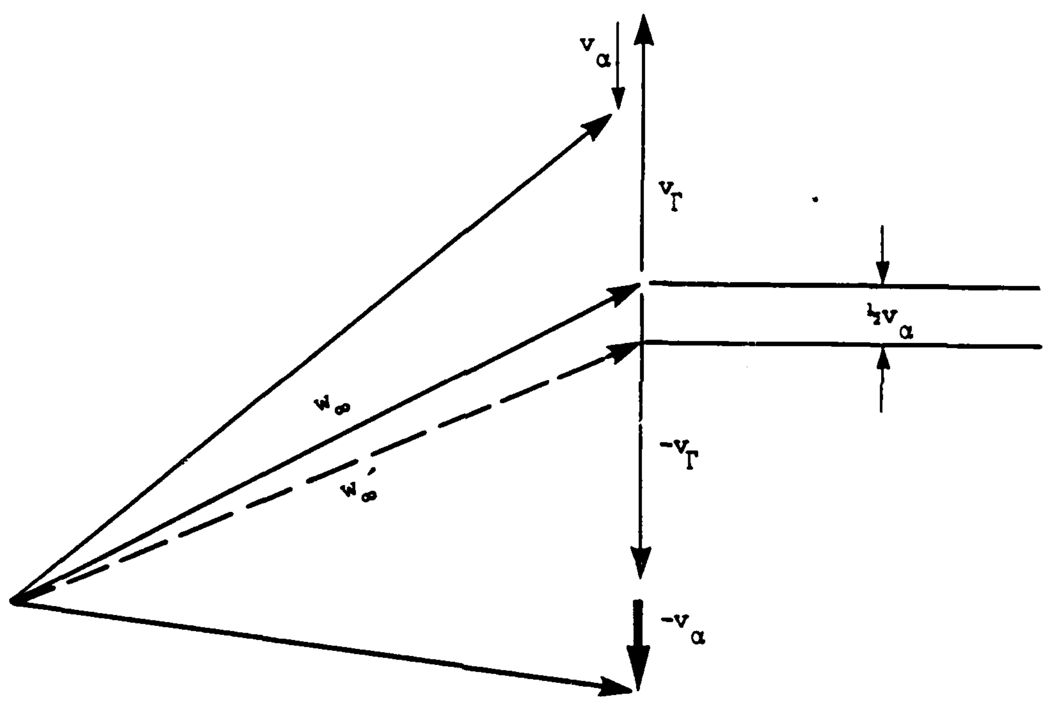


FIGURE 4-25. FLOW SKEWNESS ON THE VELOCITY DIAGRAM DUE TO VORTEX DISTRIBUTION

current method is order of magnitude α. This fact indicates that not only the blade camber profile needs correction as discussed earlier but also the blade setting should be changed in order to take such 3-D correction into account correctly. Inoue, et al. (1979, 1980) did not see this point in their series of papers.

There exist two possible ways of implementing this idea: 1) by assessing the flow channel contraction or expansion in terms of equivalent α, Equation 4.4-40 will be used, and 2) a more accurate calculation will be made by using the distributed vortices over the cascade strip. The method of actually incorporating this flow "skewness" into the current design theory will be one of the major tasks in the FY '87 GHR project.

4.4.7 Secondary Flow Correction

The secondary flow theory commonly used for axial flow is applicable to the current diagonal flow. The strength of vortex at the exit of blade channel due to the secondary flow, ζ_{SF}, is expressed

$$\zeta_{SF} = \frac{N}{2\pi r_2 \cos\beta_2} \left(\zeta_I \, w_1 \frac{b_1}{b_2} \oint \frac{ds}{w} - \frac{1}{\cos\varepsilon} \frac{d\Gamma}{dq} \right) \tag{4.4-41}$$

where

N = number of blades,

ζ_I = the component of vortex normal to the direction of relative flow on the rotating surface,

w_1 = relative incoming velocity

s = line element along the blade on the rotating surface

w = relative flow velocity on the blade surface

Γ = circulation of blade

q = q-line used for the streamline curvature method

ε = angle of a tangent line of meridional line made with q-line

b_1, b_2 = widths of flow at inlet and exit, respectively

β_2 = flow angle at exit relative to chord line.

By distributing the above secondary vortex at the exit of blade channel and satisfying the boundary condition on the hub and casing, the induced flow effect on C_θ (called ΔC_θ hereafter), will be calculated by solving the corresponding Poisson's equation. In the current design problem, this ΔC_θ will be taken into consideration for determining the blade profile shape. More detailed numerical analysis will be developed in future work.

4.5 DESIGN PROCEDURE

The design procedure presented here is based on the assumption that the shape and steady velocity of underwater vehicle is given. The whole procedure can be repeated for a modified shape or velocity. At the beginning, some parameters, such as the location of the rotor, the number of blades, and rotation speed, have to be chosen based on experiences. The shroud opening should be such that a peak efficiency is reached.

Then (Figure 4-2), the stream curvature method (SCM) is used to determine the streamlines for a through-flow in a meridional plane. As discussed in Section 4.2, the method is based on solving the momentum equation along quasi-orthogonal lines (q-lines) and the conservation of mass is kept along the meridional direction.

After streamlines are determined, average stream surfaces are taken to be the revolution of streamlines about the axis of rotation. Then a program, DSN3, is used to do the remaining design procedure. Figure 4-27 depicts the macro view of the procedure while Figures 4-26 through 4-30 show details of each substep based on the theory presented in Sections 4.1 to 4.4.

Figure 4-26 shows the input data required in the general background. Input data related to each individual cross-section are to be read in subroutine INP2 as shown in Figure 4-27. Some input data are obtained from the meridional flow solutions computed in the SCM program.

Figure 4-30 shows the subroutine to calculate the flow velocities in the mapped (X, Y) plane as discussed in Section 4.4.2. The subscripts 1 and 2 denote conditions at inlet and exit, respectively, of a blade.

One basic concept in the present design procedure is that the blade camber design based on cascade experimental data is adjusted, by considering the effects of streamline inclination and nonuniform velocity distribution, such that both the final flow turning angle and the incident angle relative to the axis of rotation are the same as those obtained in the two-dimensional flow without the three-dimensional factors.

The camber and turning angle for the case of uniform, parallel flow are evaluated based on the required lift coefficient and solidity (Figure 4-31). The method relies on the experimental data presented in Section 4.3.3.

In the presentation of three-dimensional effects, which are to be considered in the boundary condition as discussed in Section 4.4, the camber and stagger angle are adjusted until the turning angle in the mapped two-dimensional plane is the same as that with original camber in the uniform, parallel flow (Figure 4-32). A linearized cascade theory (Section 4.3.2), together with proper boundary condition which has the three-dimensional

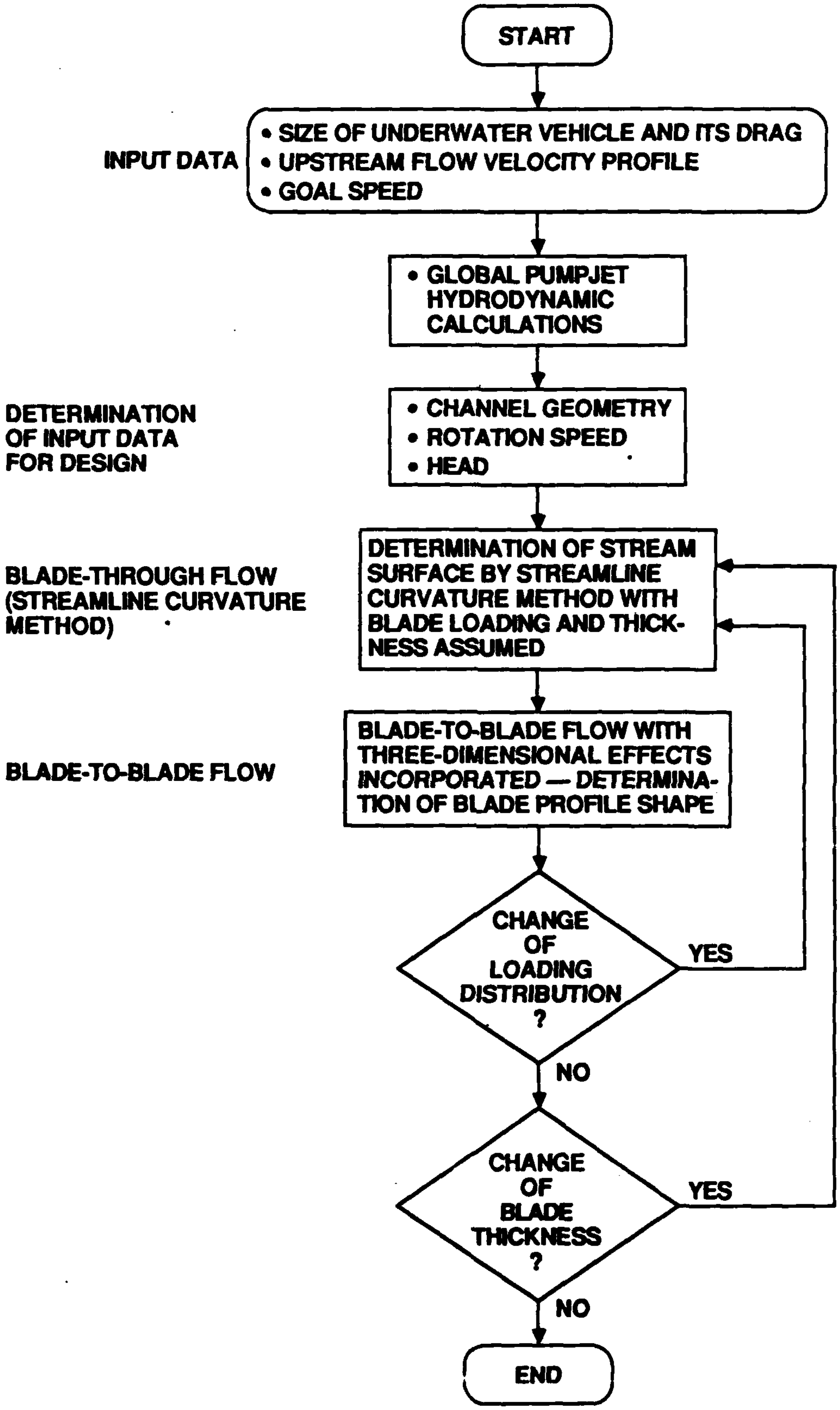


FIGURE 4-26. FLOW CHART OF THE SELECTED PUMPJET DESIGN METHOD

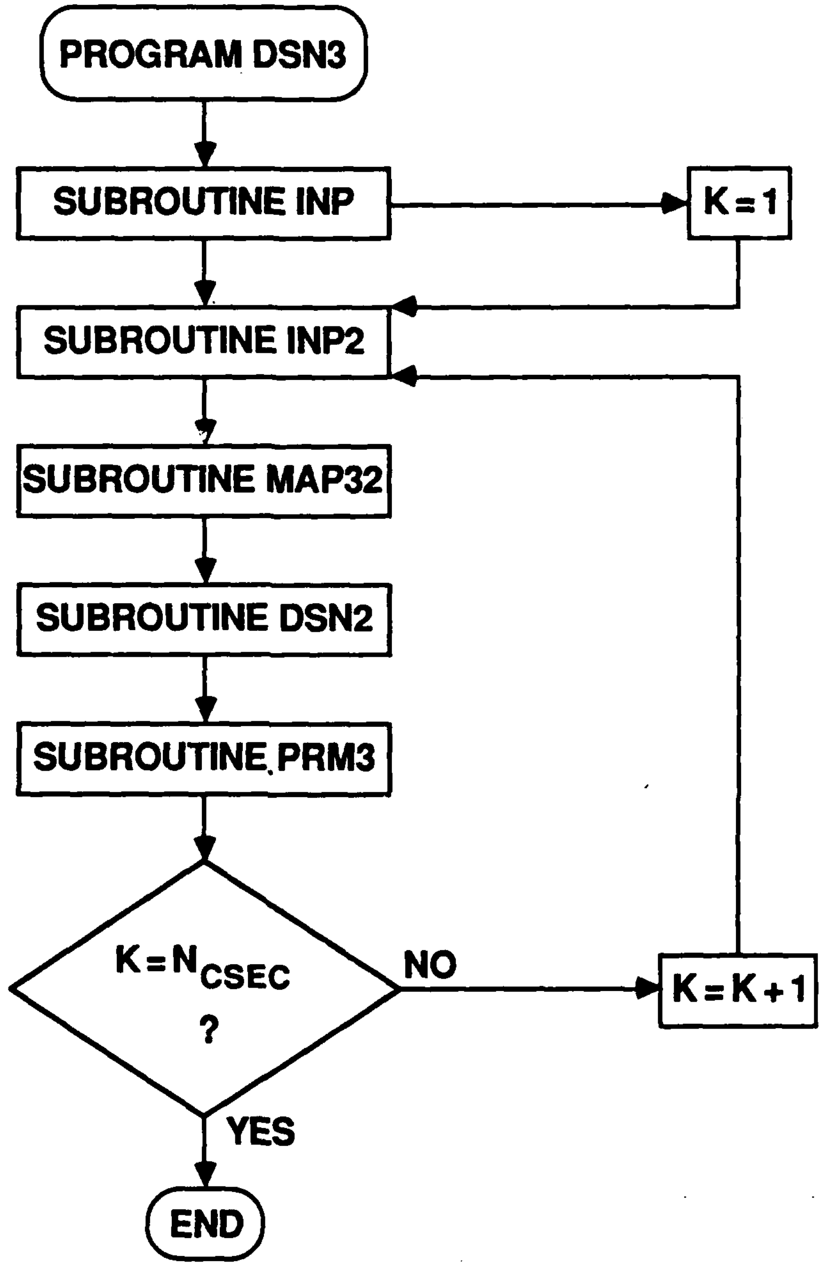


FIGURE 4-27. GENERAL FLOW CHART TO DESIGN BLADE IN A FLOW OF THREE-DIMENSIONAL CHARACTER

SUBROUTINE INP

↓

INPUT

ROTATIONAL SPEED	RPM
CONVERGENCE CRITERIA FOR $\Delta\phi_{eq}$, $\Delta\beta_1$ AND $\Delta\sigma$	REFERENCE RADIUS VARIOUS CONVERGENCE FACTORS
MAX. ITERATION NUMBER	M_{max}
NO. OF BLADES	N_{blade}
NO. OF CROSS-SECTIONS ALONG THE RADIUS DIRECTION	N_{csec}
NO. OF SEGMENTS ALONG A CHORD	N_{seg}
REF. RADIUS	r_o
NO. OF STATIONS ALONG A CHORD	N_{sta}
CODE TO DETERMINE THE WAY OF OBTAINING SECTION DATA	I_{nterp}

↓

RETURN

FIGURE 4-28. FLOW CHART FOR SUBROUTINE INP

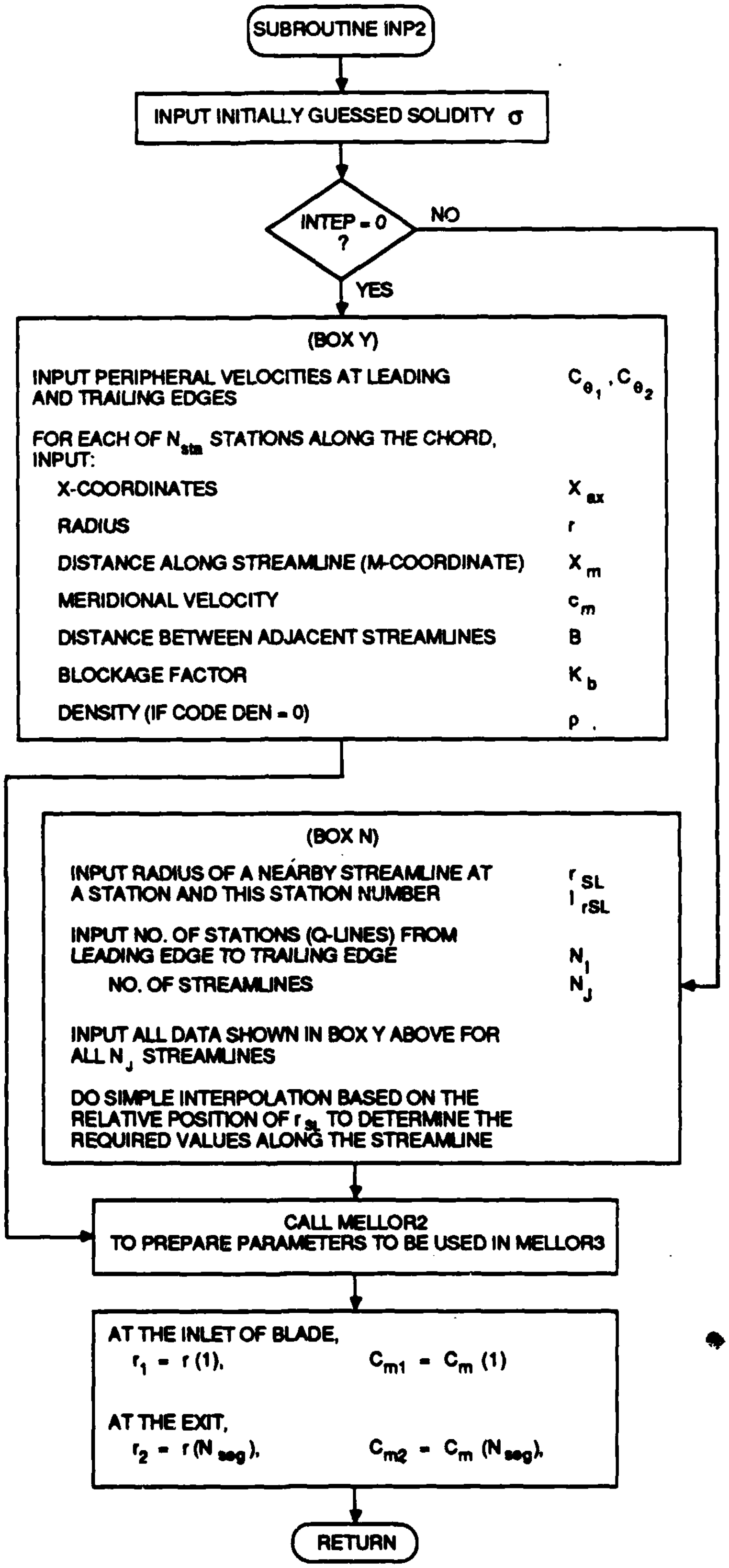


FIGURE 4-29. FLOW CHART FOR SUBROUTINE INP2

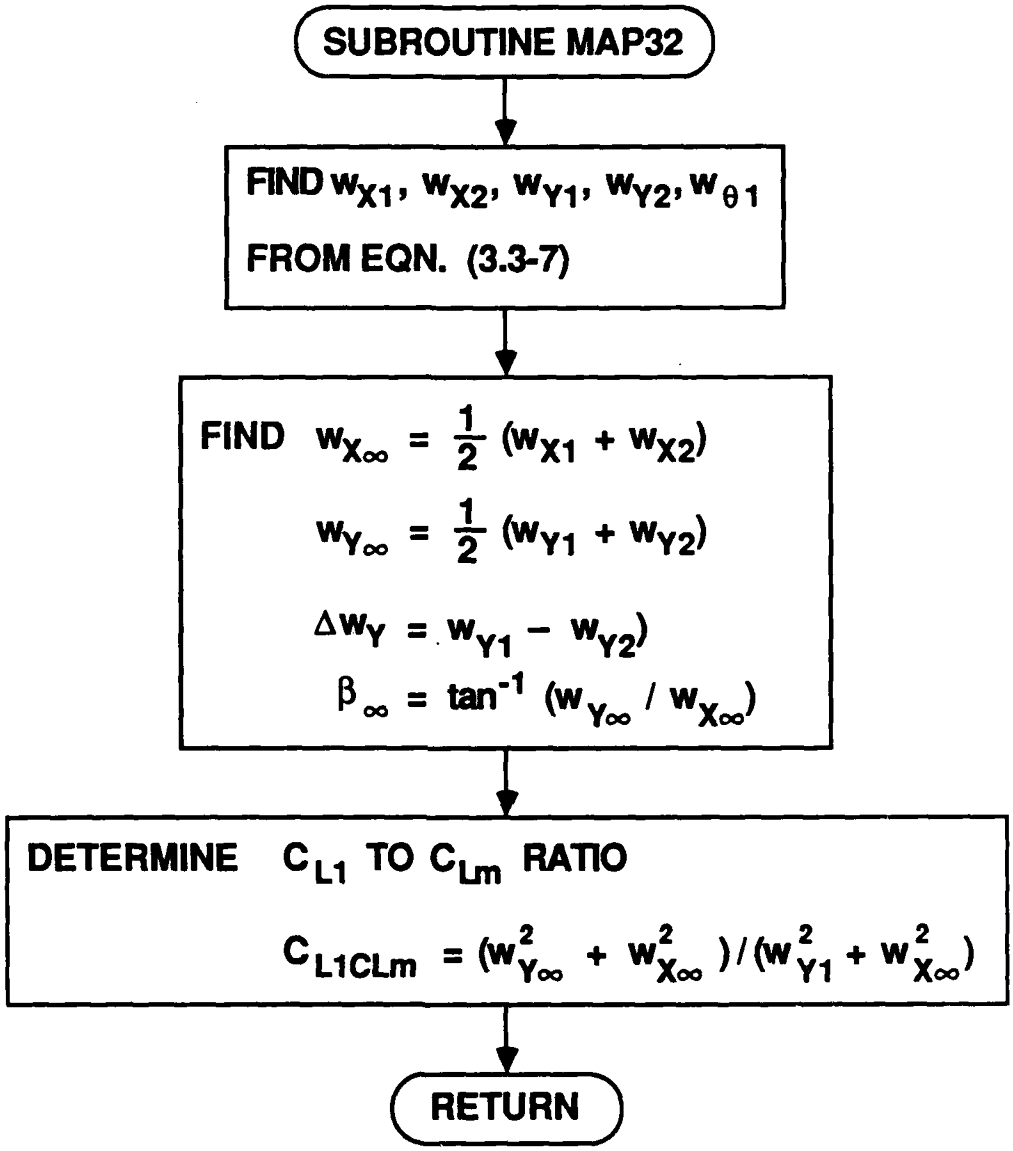


FIGURE 4-30. FLOW CHART FOR SUBROUTINE MAP32

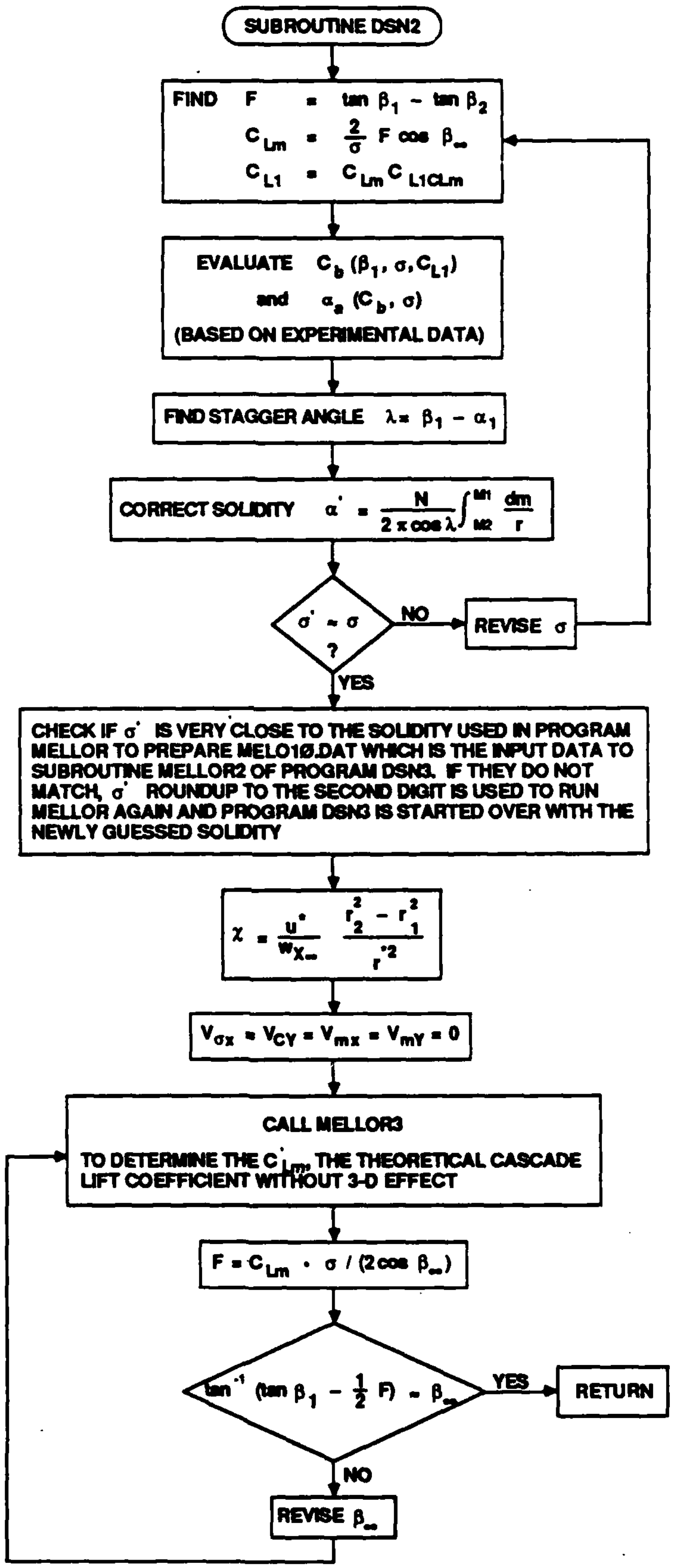


FIGURE 4-31. FLOW CHART FOR SUBROUTINE DSN2

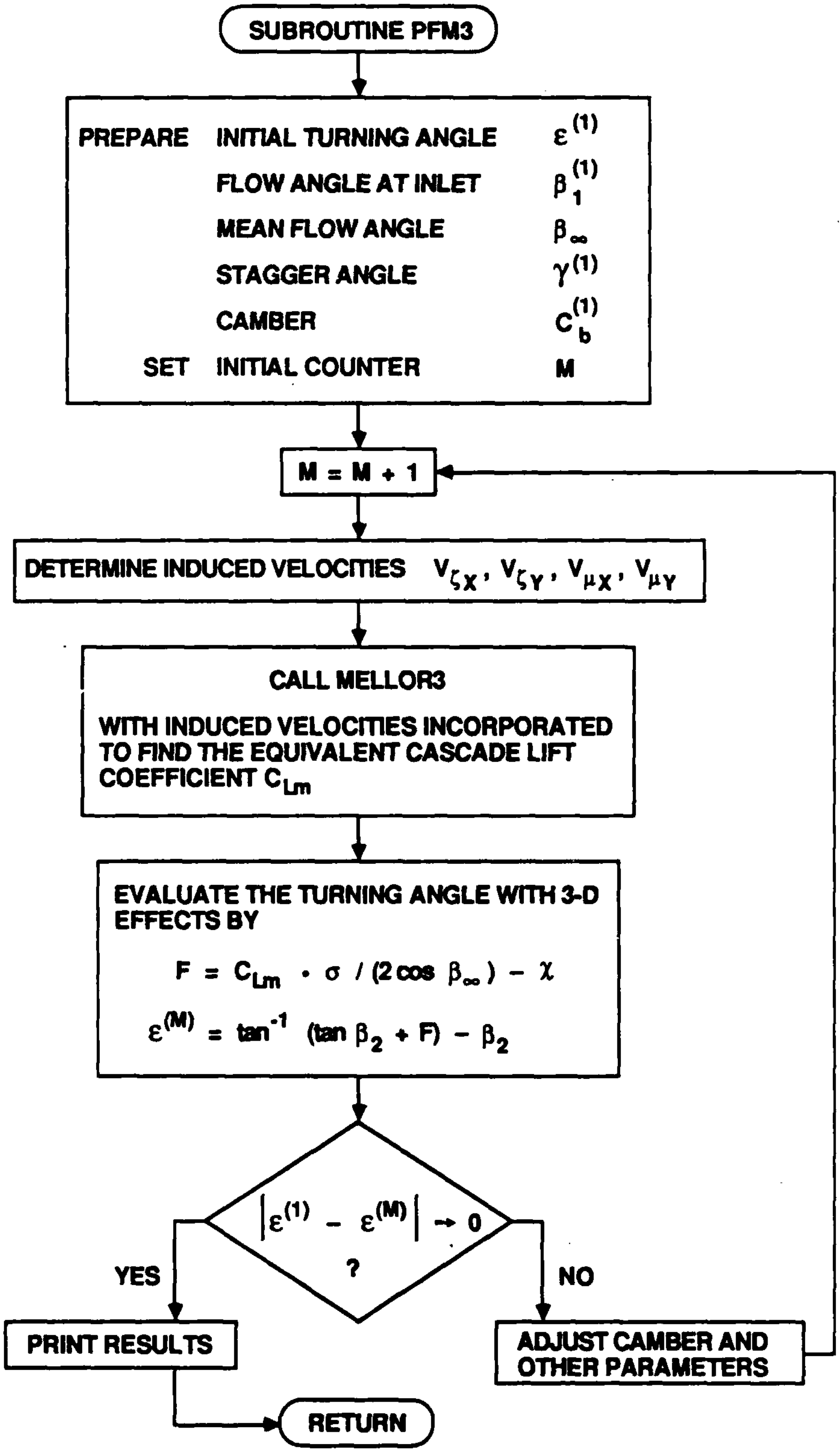


FIGURE 4-32. FLOW CHART FOR SUBROUTINE PFM3

964-8-38
100688

effects considered (Section 4.4), is used to calculate the cascade lift coefficient and the associated total circulation.

If the calculated turning angle for the three-dimensional flow case is different from the desired value, the camber and stagger angles are adjusted in an iteration process until the desired value is achieved to within certain tolerance criterion. When the result is converged, the final camber is the one which, under the influence of three-dimensional flow, will yield the desired lift coefficient.

5.0 PROGRAM

5.1 DESIGN SPECIFICATION

The propulsor configuration and its performance are highly dependent on the hydrodynamic characteristics of the vehicle. Specifically, the distortion of the velocity distribution of the incoming flow to the propulsor plays a key role in designing the rotor blade profiles. The following information are needed to design a pumpjet:

1) Geometry of an underwater vehicle and its drag,
2) Upstream flow velocity profile, and
3) Goal speed.

The geometric characteristics of an underwater vehicle are determined by its mission, hotel load requirement, and the launching equipment to be used. The hotel load will determine the required volume and weight of the vehicle, and the launching equipment limits the size of vehicle. Once the weight of the vehicle is properly determined, the stability-maneuverability requirement will determine the area and position required for the control surfaces. Then, the drag on the vehicle can be either measured from scaled model tests in a wind or water tunnel, or calculated from the empirical formula such as those given by J.D. Brook and T.G. Lang[1].

The pumpjet needs to accept the distorted flow of the boundary layer generated by the hull, and those requiring accurate measurement or prediction of the velocity profile at the inlet.

---

[1] Brooks, J.D., and T.G. Lang, "Simplified Methods for Estimating Torpedo Drag", **Underwater Missile Propulsion**, L. Greiner, ed., Compass Publications, Inc. (Arlington, Virginia), 1967, pp. 117-146.

In the normal procedure, the potential flow around the vehicle is calculated with an estimated value of the boundary layer thickn ss around the hull. An iterative procedure is employed until the estimated boundary layer thickness matches with that obtained from the boundary layer analysis. Once the correct boundary layer thickness is obtained, the velocity profile can be calculated as the result of a turbulent boundary layer theory analysis.

However due to the lack of time required to run such an iteration procedure, a fictitious velocity profile at a specified inlet position is used for exercising these sample design calculations.

5.2 MOMENTUM THEORY

A simple momentum theory should be applied to a control volume enclosing the target vehicle which a project is designed. The detailed theoretical basis is provided in Section 4.1.1 of the report of Furuya (and Chiang 1986)). By utilizing the velocity profile obtained through an experiment or use of a turbulent boundary layer computer code, the maximum propulsive efficiency can be determined with the optimum opening diameter also to be determined.

Due to the limited time available for completing the project, this routing was not exercised, but an arbitrarily chosen shroud diameter and profile shape were used for running the SCM in the following section.

5.3 INPUT TO AND OUTPUT FROM SCM (STREAMLINE CURVATURE METHOD)

It is difficult to obtain analytically the velocity profiles near the aft end of a body of revolution since the afterbody curvature often produces a strong adverse pressure gradient normal to the flow. One of the most reliable ways to estimate the velocity profile is to fabricate a scale model and measure the necessary velocity distribution either in a wind or water tunnel. Such model testings are usually very costly. An alternative way is to use the streamline curvature method (SCM) in which detailed flow profiles are numerically calculated at various stations. (See R.A. Novak[2] for detail explanation on SCM.)

To obtain the velocity profiles through the SCM program, coordinates of the vehicle hull are used as one of the boundaries. The outer contour is then defined by a streamline. Figure 5.2-1 shows the geometry of a vehicle afterbody and the locations of various stations. This geometric information, flow blockage effect produced by the rotor blade thickness, velocity profile at Station #1 and radial distribution of peripheral velocity generated by the rotor at Station #3, are the required input data for the program. Figures 5.2-2 and 5.2-3 show the velocity profile at Station #1 and the radial distribution of peripheral velocity at Station #3 respectively, used for the present sample calculations.

The output from the SCM program includes the locations of meridional streamlines and the velocity profiles at various downstream locations. Figure 5.2-1 shows the computed meridional streamlines in comparison to an initial guess. Figure 5.2-4 shows the velocity profiles at the rotor inlet and exit.

---

[2] Novak, R.A., "Streamline Cuıvature Computing Procedures for Fluid Flow Problems", ASME Paper 66-WA/GT-3, November 27, 1966.

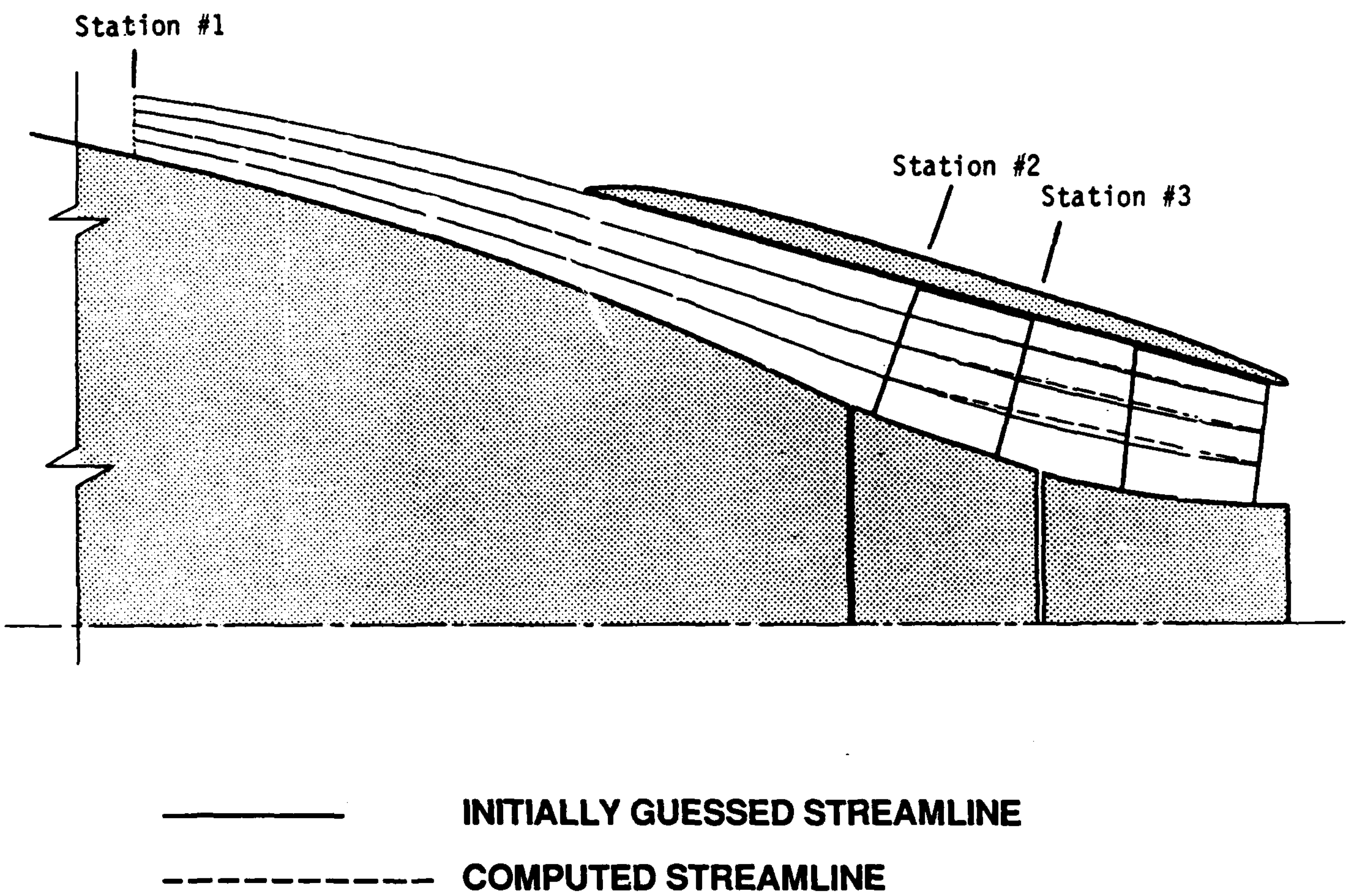


FIGURE 5.2-1. GEOMETRY OF AFTERBODY AND MERIDIONAL STREAMLINES

964-8-39
081088

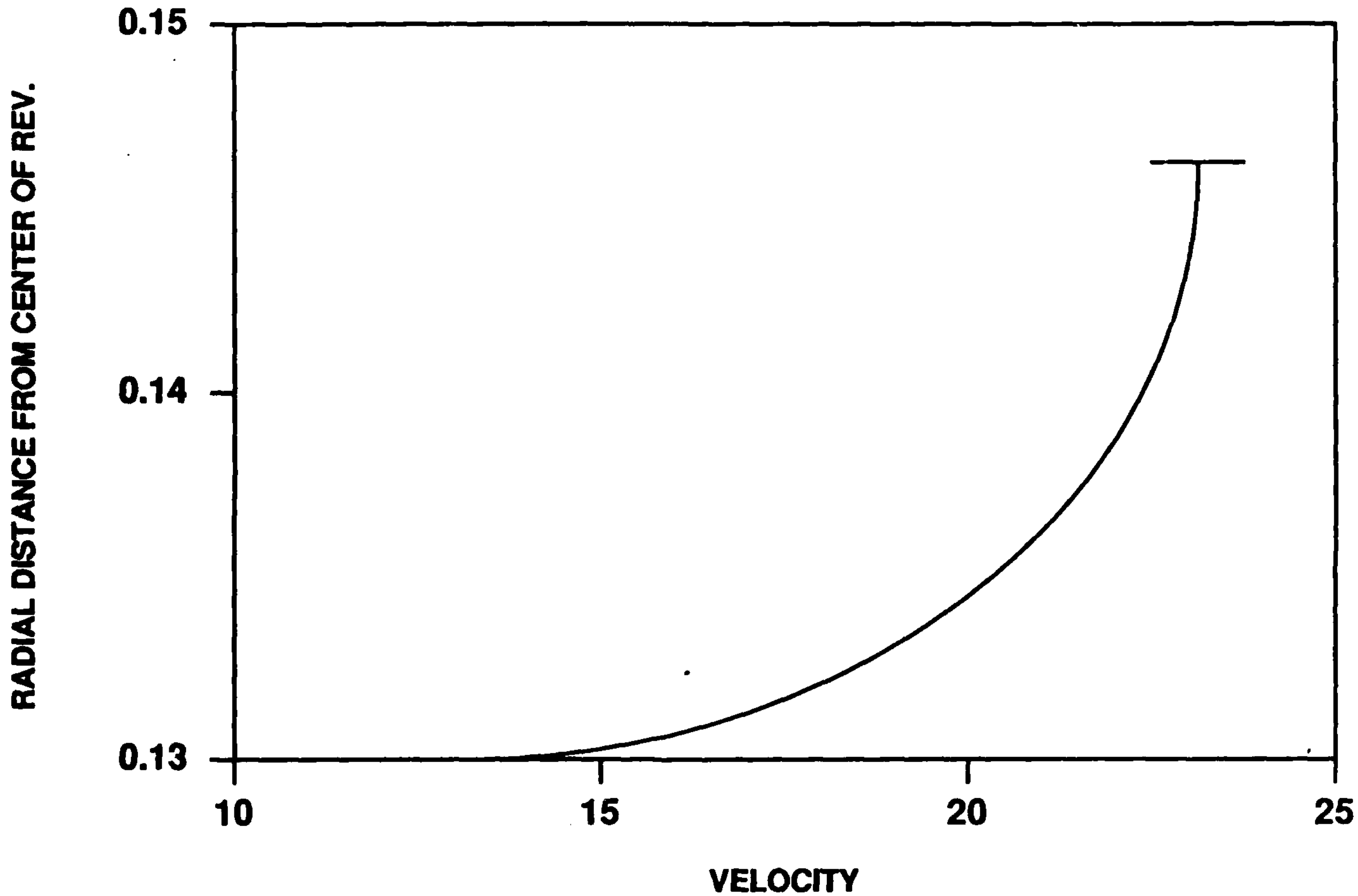


FIGURE 5.2-2. MERIDIONAL VELOCITY PROFILE AT STATION #1

964-8-40
100688

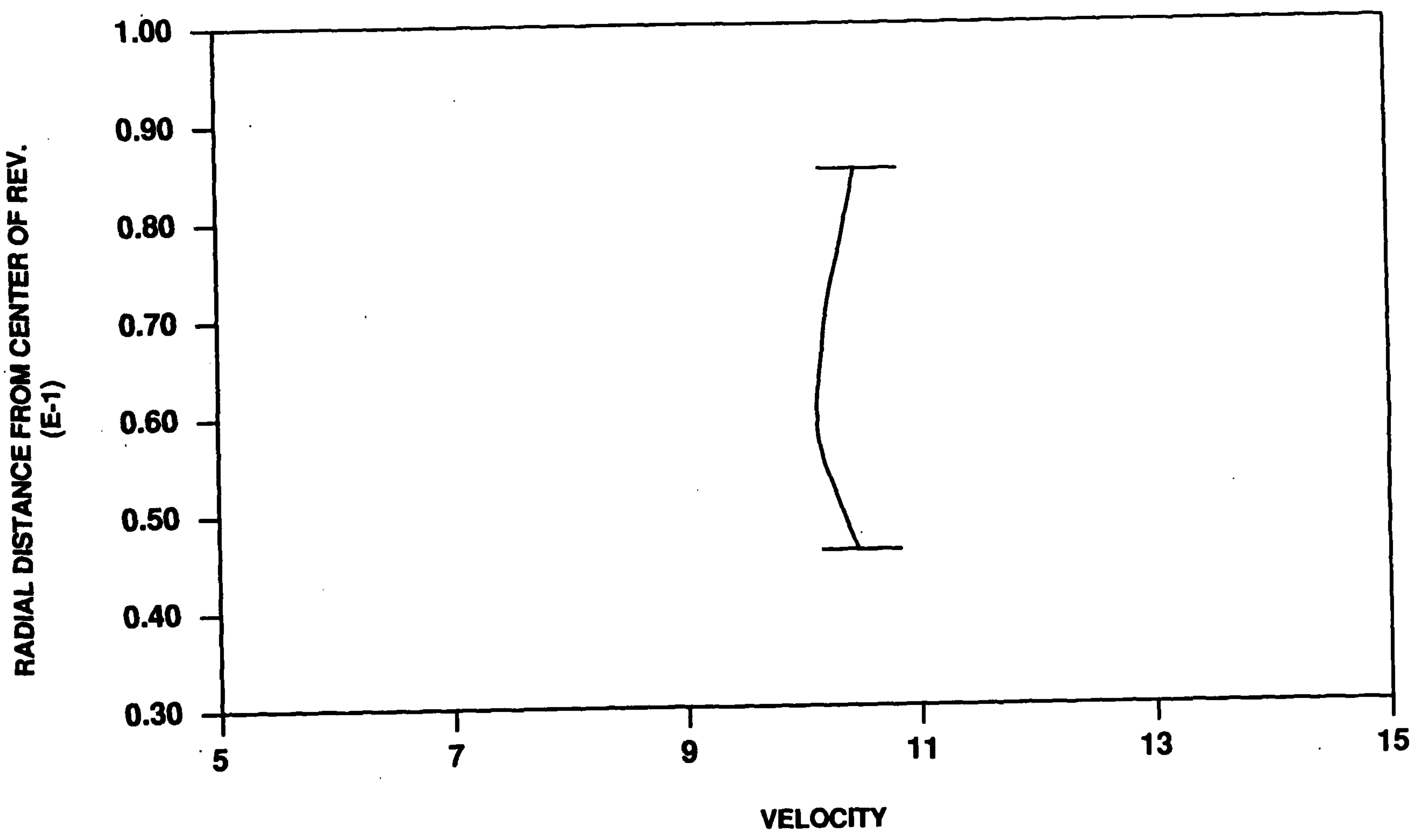


FIGURE 5.2-3. PERIPHERAL VELOCITY DISTRIBUTION AT STATION #3

964-8-41
100688

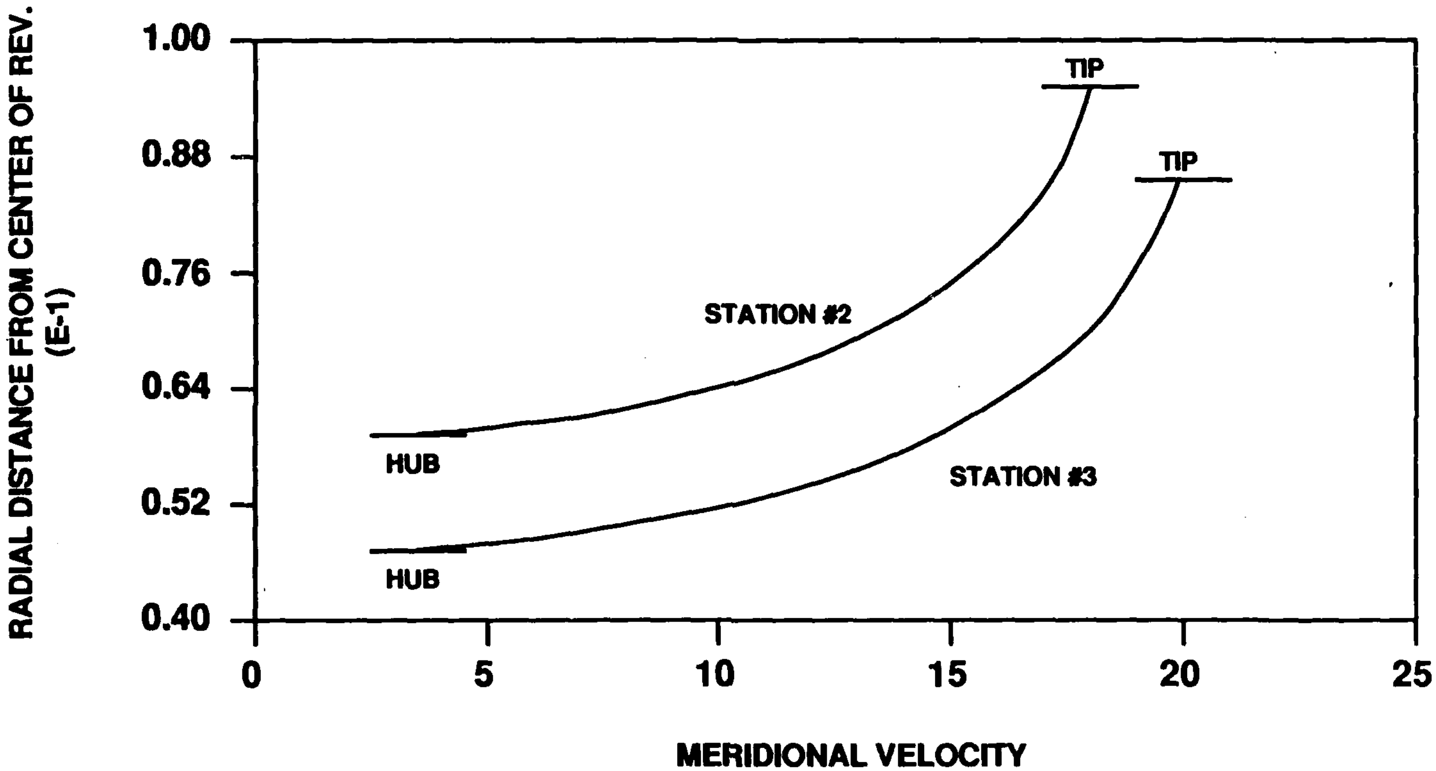


FIGURE 5.2-4. MERIDIONAL VELOCITY PROFILES AT STATIONS #2 AND #3

964-8-42
100688

TABLE 5.3-1. LISTING OF INPUT FILE TO PROGRAM SCM

```
      -2              1             1 (0) IBFLOW, IDSN3, LG
(3037-SCMI5NU_2.DAT) FOLLOWING ARE INPUT DATA FOR SUB NUFLOW...      86, REVISED 20OCT87        (1)
       5              5             2             3            2        (2) NM, NQ, NQI, NQB, ICM
 360.0000       0.1620000      1.000000      0.7000000     0.7000000
 1.0000000E-06  1.0000000E-03  9.9999997E-05  (3) OMG, RAS, DMPCM, DMPG, DMPL, EPSCM, EPSG, EPSL
 0.0000000E+00 -0.3002000     -0.3578000     -0.3386000     -0.2723000      (4) RUMRQ(I)
 2.493000       2.631000       2.715000       2.732000       2.750000       (5) Z(I,1)
 -7.000000     -12.20000      -17.50000      -20.00000      -15.00000
 -18.40000     -16.50000      -18.40000      -15.50000      -18.40000       (6)  PHID1(I), PHIDN(I)
 0.0000000E+00  0.0000000E+00  0.1380000      0.1499000      0.2220000
 0.2377000      0.2390000      0.2520000      0.2570000      0.2687000      (7)   SM(I,1), SM(I,NM)
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 5.150000       5.150000       5.150000       5.150000       5.150000
 10.30000       10.30000       10.30000       10.30000       10.30000       (8) CTH(I,J)
 1.000000       1.000000       1.000000       1.000000       1.000000
 1.000000       1.000000       1.000000       1.000000       1.000000
 1.000000       1.000000       1.000000       1.000000       1.000000
 1.000000       1.000000       1.000000       1.000000       1.000000
 1.000000       1.000000       1.000000       1.000000       1.000000       (9) BLO(I,J)
 68.58108       137.1622       205.7432       274.3243     (12) GS(I)
 12.86100       19.09565       20.87901       22.03418       22.94403
 10.08009       14.91218       16.24901       17.08810       17.72967
 9.555634       14.10877       15.35382       16.13017       16.72132
 9.975438       14.74704       16.06006       16.88045       17.50512
 10.43815       15.44135       16.82594       17.69516       18.35966       (13)   CM(I,J)
 0.1350450      0.1398175      0.1437123      0.1472649      0.1505900
 9.4233997E-02  0.1023083      0.1086411      0.1142517      0.1193800
 5.8674000E-02  7.1355730E-02  8.0497034E-02  8.8200070E-02  9.4995998E-02
 5.2577998E-02  6.5953061E-02  7.5394720E-02  8.3268791E-02  9.0170003E-02
 4.6735998E-02  6.1068624E-02  7.0922948E-02  7.9042293E-02  8.6106002E-02  (14)    R(I,J)
 150.9785       255.6788       292.8486       315.7634       334.5784
 150.9785       255.6788       292.8486       315.7634       334.5784
 150.9785       255.6788       292.8486       315.7634       334.5784
 248.4581       377.9558       432.6305       470.1437       501.7536
 324.2756       482.1213       555.8309       608.8522       653.8595       (15)    H(I,J)
END OF INPUT DAT SCMI5NU_2.DAT
```

5.4 INPUT DATA TO MELLOR'S PROGRAM

Program MELLOR, listed in Appendix B.3, requires three input data files in this study. These three files are named RI.DAT, BLADE.DAT, and MELI.DAT. The output data files include MELO.DAT and MELO10.DAT. The nomenclature of input data is presented in Appendix C.3.

The file RI.DAT is generated from Program RIS which is listed in Appendix B.2. Cascade influence functions, R and I, (Mellor, 1959) are calculated in this program to be used in Program MELLOR for a variety of applications. The input data to this program RIS is called RISI.DAT and listed in Table 5.4-1. The nomenclature of input data is listed in Appendix C.2. As this sample input data shows, the cascade influence functions are calculated for the stagger angle ranges from 0 to 60 degrees at an interval of 15 degrees. The (x0-x)/s ratio ranges from 0 to 2 at an interval of 0.05. The convergence criterion used in Program RIS is 0.000001.

The file RI.DAT as one of input files to Program MELLOR is listed in Table 5.4-2. In this sample input file, values of function R and function I are listed after the parameters which define the range of the functions. The function values inside this matrix range are interpolated in Program MELLOR.

The second input file, BLADE.DAT, is listed in Table 5.4-3. The thickness distribution of NACA 65-010 is taken from Herrig, et al. (1951). A flag in the input file MELI.DAT controls whether the distribution of camber slope is to be obtained from the input camber data given by BLADE.DAT or calculated from a formula. This ample study utilizes a formula (Abbott & von Doenhoff, 1959, eq. 4.25):

TABLE 5.4-1. LISTING OF INPUT DATA TO PROGRAM RIS

```
(3037-RISI.DAT) INPUT FILE FOR 3037-RIS
0., 60., 15., 0., 2., .05       AMDA, AMDZ, AMDI, XSA, XSZ, XSI
3000, 1E-6                      NMAX, EPS
END OF RISI.DAT
```

964-8-44
081088

TABLE 5.4-2. LISTING OF RI.DAT, AN INPUT DATA FILE TO PROGRAM MELLOR

```
(3037-RI.DAT) GENERATED FROM RIS.FOR FOR MELLOR.FOR
 1.0000000E-06  0.0000000E+00   60.00000        15.00000        0.0000000E+00
  2.000000      5.0000001E-02
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 5.2173398E-02  4.5208521E-02  2.6169103E-02  8.6473985E-05 -2.6082352E-02
 0.1038949      9.0213299E-02  5.2621908E-02  6.8930385E-04 -5.1932193E-02
 0.1546278      0.1347106      7.9560533E-02  2.3246906E-03 -7.7233396E-02
 0.2039254      0.1784404      0.1071651      5.5044037E-03 -0.1016441
 0.2514041      0.2211502      0.1355503      1.0727535E-02 -0.1247449
 0.2967545      0.2625987      0.1647505      1.8465405E-02 -0.1460087
 0.3397457      0.3025640      0.1947111      2.9137187E-02 -0.1647694
 0.3802256      0.3408486      0.2252868      4.3076873E-02 -0.1801937
 0.4181130      0.3772891      0.2562491      6.0493194E-02 -0.1912633
 0.4533932      0.4117613      0.2873063      8.1427529E-02 -0.1967846
 0.4861057      0.4441831      0.3181238      0.1057185     -0.1954414
 0.5163313      0.4745143      0.3483529      0.1329817     -0.1859200
 0.5441841      0.5027574      0.3776606      0.1626173     -0.1671294
 0.5697998      0.5289490      0.4057501      0.1938487     -0.1385061
 0.5933266      0.5531567      0.4323842      0.2257907     -0.1003568
 0.6149191      0.5754741      0.4573899      0.2575375     -5.4117262E-02
 0.6347337      0.5960090      0.4806646      0.2882531     -2.3720309E-03
 0.6529205      0.6148804      0.5021708      0.3172460      5.1462054E-02
 0.6696242      0.6322143      0.5219283      0.3440161      0.1037472
 0.6849818      0.6481345      0.5400006      0.3682702      0.1513210
 0.6991182      0.6627645      0.5564828      0.3899067      0.1920179
 0.7121500      0.6762204      0.5714924      0.4089816      0.2248574
 0.7241826      0.6886137      0.5851550      0.4256665      0.2499200
 0.7353121      0.7000443      0.5976023      0.4402044      0.2680355
 0.7456259      0.7106067      0.6089611      0.4528722      0.2804396
 0.7552003      0.7203860      0.6193488      0.4639508      0.2884880
 0.7641057      0.7294580      0.6288769      0.4737064      0.2934675
 0.7724064      0.7378905      0.6376457      0.4823773      0.2964911
 0.7801540      0.7457454      0.6457406      0.4901679      0.2984570
 0.7874034      0.7530785      0.6532381      0.4972489      0.3000457
 0.7941947      0.7599384      0.6602086      0.5037566      0.3017366
 0.8005722      0.7663677      0.6667088      0.5097985      0.3038370
 0.8065685      0.7724059      0.6727877      0.5154570      0.3065091
 0.8122170      0.7780849      0.6784909      0.5207939      0.3098033
 0.8175462      0.7834380      0.6838531      0.5258540      0.3136829
 0.8225805      0.7884922      0.6889084      0.5306704      0.3180547
 0.8273461      0.7932720      0.6936845      0.5352667      0.3227890
 0.8318605      0.7977986      0.6982051      0.5396599      0.3277435
 0.8361445      0.8020902      0.7024913      0.5438622      0.3327767
 0.8402145      0.8061675      0.7065633      0.5478841      0.3377616
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  2.6051806E-02  4.5177013E-02  5.2259099E-02  4.5325797E-02
 0.0000000E+00  5.1693503E-02  8.9961112E-02  0.1045710      9.1154113E-02
 0.0000000E+00  7.6480716E-02  0.1338598      0.1568561      0.1378854
 0.0000000E+00  0.1000385      0.1764258      0.2090313      0.1859587
 0.0000000E+00  0.1220554      0.2172237      0.2609444      0.2358032
 0.0000000E+00  0.1422951      0.2558396      0.3123513      0.2878196
```

964-8-45
081088

TABLE 5.4-2 (CONTINUE)

0.0000000E+00	0.1606027	0.2918963	0.3629004	0.3423521
0.0000000E+00	0.1769041	0.3250709	0.4121253	0.3996373
0.0000000E+00	0.1911983	0.3551127	0.4594491	0.4597335
0.0000000E+00	0.2035502	0.3818625	0.5042129	0.5224231
0.0000000E+00	0.2140741	0.4052587	0.5457183	0.5870879
0.0000000E+00	0.2229210	0.4253430	0.5832939	0.6525828
0.0000000E+00	0.2302646	0.4422548	0.6163695	0.7171478
0.0000000E+00	0.2362877	0.4562132	0.6445497	0.7784145
0.0000000E+00	0.2411720	0.4675004	0.6676732	0.8335955
0.0000000E+00	0.2450914	0.4764381	0.6858298	0.8798838
0.0000000E+00	0.2482054	0.4833625	0.6993546	0.9150272
0.0000000E+00	0.2506564	0.4886056	0.7087771	0.9378765
0.0000000E+00	0.2525685	0.4924799	0.7147540	0.9487000
0.0000000E+00	0.2540476	0.4952668	0.7179930	0.9491126
0.0000000E+00	0.2551827	0.4972095	0.7191823	0.9416457
0.0000000E+00	0.2560472	0.4985138	0.7189455	0.9291563
0.0000000E+00	0.2567002	0.4993477	0.7178043	0.9143054
0.0000000E+00	0.2571902	0.4998435	0.7161744	0.8992313
0.0000000E+00	0.2575548	0.5001053	0.7143623	0.8854305
0.0000000E+00	0.2578243	0.5002100	0.7125776	0.8737952
0.0000000E+00	0.2580214	0.5002157	0.7109549	0.8647228
0.0000000E+00	0.2581648	0.5001622	0.7095658	0.8582494
0.0000000E+00	0.2582674	0.5000779	0.7084345	0.8541788
0.0000000E+00	0.2583403	0.4999814	0.7075584	0.8521667
0.0000000E+00	0.2583915	0.4998844	0.7069145	0.8518068
0.0000000E+00	0.2584263	0.4997938	0.7064697	0.8526737
0.0000000E+00	0.2584499	0.4997140	0.7061867	0.8543637
0.0000000E+00	0.2584645	0.4996448	0.7060302	0.8565169
0.0000000E+00	0.2584735	0.4995878	0.7059656	0.8588344
0.0000000E+00	0.2584784	0.4995399	0.7059633	0.8610812
0.0000000E+00	0.2584804	0.4995025	0.7060025	0.8630911
0.0000000E+00	0.2584800	0.4994707	0.7060635	0.8647590
0.0000000E+00	0.2584781	0.4994464	0.7061325	0.8660343
0.0000000E+00	0.2584758	0.4994268	0.7062013	0.8669177

999

END OF RI.DAT

FIGURE 5.4-3. LISTING OF BLADE.DAT, AN INPUT DATA FILE TO PROGRAM MELLOR

```
(3037-BLADE.DAT) THICKNESS (REVISED THICKNESS OF NACA 65-010, HERRIG, ET AL, 1951, P.23)... 24OCT86 (1105) TITL
26      (1106) IN
  0.    .5    .75  1.25  2.5   5.    7.5  10.   15.   20.
 25.   30.   35.   40.   45.   50.   55.   60.   65.   70.
 75.   80.   85.   90.   95.  100.       (1107) ( XIN(I), I=1,IN)
  0.    .752  .890 1.124 1.571 2.222 2.709 3.111 3.746 4.218
  4.570 4.824 4.982 5.057 5.029 4.870 4.570 4.151 3.627 3.038
  2.451 1.847 1.251  .749  .354  .150   (1108) ( YIN(I), I=1,IN)
999
END OF (3037-MELLOR-BLADE.DAT)

(3037-BLADE.DAT) CAMBER (NACA MEAN LINE a = 1.0, ABBOTT & VON DOENHOFF, 1959) & THICKNESS... 24OCT86 (1101) TITL
31      (1102) IN
  0.    .5    .75  1.25  2.5   5.    7.5  10.   15.   20.
 25.   30.   35.   40.   45.   50.   55.   60.   65.   70.
 75.   80.   85.   90.   92.5  95.   97.5  98.75 99.25 99.5
100.    (1103) ( XIN(I), I=1,IN)
  0.    0.250 0.350 0.535 0.930 1.580 2.120 2.585 3.365 3.980
  4.475 4.860 5.150 5.355 5.475 5.515 5.475 5.355 5.150 4.860
  4.475 3.980 3.365 2.585 2.120 1.580 0.930 0.530 0.350 0.250
  0.    (1104) ( YIN(I), I=1,IN)
```

964-8-47
081088

$$dy / dx = \ln(c/x - 1) / (4\pi)$$

to calculate the slope. Instead, a set of camber data for NACA mean line a=0 (Abbott & von Doenhoff, 1959), shown as the lower half of Table 5.4-3, can be used in front of the thickness distribution data in the table.

Listed in Table 5.4-4 is the controlling input data MEL.DAT. In this sample study, 100 points along the chord line is used in the computation to establish the basis in determining the cascade lift coefficient. Fourier series after the seventh term is truncated. This input file is used to generate a data file MELO10.DAT for Program DSN3 (see Sec. 5.5). The solidity of 1.35 in Table 5.4-4 can be replaced by other values to generate the required data for the corresponding solidity.

5.5 INPUT DATA TO MAIN INTEGRATION DESIGN PROGRAM (MIDP)

The main integration design program (MIDP) is listed as Program DSN3 in Appendix B.4. It requires three input data files: DSN3ZI.DAT, MELO10.DAT, and DSN#I.DAT. The nomenclature of input data is presented in Appendix C.4. The output from the program is kept in the file DSN3O.DAT.

The file DSN3ZI.DAT (Table 5.5-1) is an output of Program SCM. The main information in this file is the meridional velocity and peripheral velocity distributions in the rotor area. Also included is the location of streamlines, based on various coordinates, and other parameters such as blockage coefficient and fluid density.

The file MELO10.DAT (Table 5.5-2) is generated by Program MELLOR. It contains tables of g_n, h_n, B, and T (Eq. 4.3) for a particular value of solidity and a range of stagger angles.

FIGURE 5.4-4. LISTING OF MELI.DAT, AN INPUT DATA FILE TO PROGRAM MELLOR

```
(3037-MELLOR-MELI.DAT) UNIT 5 INPUT FILE FOR (3037-MELLOR)          17JUN88
MAKE MEL010.DAT FOR PROGRAM DSN3... SOLIDITY=1.18
11, 3, 0, 100, 7, 0     (2) MH, INCAM, IFLAT, NSEC, NFR, LIST
1., 1.  (3) CBI, SEND1
1.18    (9) CSA
999     (10) IDUM
END OF (3037-MELLOR-MELI.DAT)
```

964-8-48
081088

FIGURE 5.5-1. DSN3ZI.DAT, AN INPUT DATA FILE TO PROGRAM DSN3 (MIDP)

OUTPUT FROM SCM, FOR INPUT TO DSN3 20-OCT-87

3	5			
0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
10.30000	9.976315	10.11178	10.21493	10.30000
2.715000	2.732000	2.750000	2.720332	2.737401
2.754555	2.723524	2.740501	2.757123	2.726212
2.743039	2.759205	2.728580	2.745239	2.760993
5.8674000E-02	5.2577998E-02	4.6735998E-02	7.2934225E-02	6.7915648E-02
6.3050024E-02	8.1472553E-02	7.6717153E-02	7.2243065E-02	8.8661440E-02
8.3924957E-02	7.9700582E-02	9.4995998E-02	9.0170003E-02	8.6106002E-02
0.2220000	0.2390000	0.2570000	0.2372768	0.2550692
0.2728999	0.2389733	0.2566037	0.2738170	0.2406130
0.2580943	0.2748030	0.2377000	0.2520000	0.2687000
3.855058	3.512583	4.083045	14.54134	15.05411
15.68967	16.28281	17.13391	17.87474	17.23342
18.49190	19.21535	17.96828	19.49733	20.26405
0.0000000E+00	0.0000000E+00	0.0000000E+00	1.5224391E-02	1.6260931E-02
1.6938113E-02	2.4340013E-02	2.5592268E-02	2.6482830E-02	3.2014962E-02
3.3233963E-02	3.4225628E-02	3.8777813E-02	3.9854944E-02	4.0876091E-02
1.000000	1.000000	1.000000	1.000000	1.000000
1.000000	1.000000	1.000000	1.000000	1.000000
1.000000	1.000000	1.000000	1.000000	1.000000
1000.000	1000.000	1000.000	1000.000	1000.000
1000.000	1000.000	1000.000	1000.000	1000.000
1000.000	1000.000	1000.000	1000.000	1000.000

FIGURE 5.5-2. MEL010.DAT, AN INPUT FILE TO PROGRAM DSN3 (MIDP)

```
(3037-MELLOR-MEL010) G & H FUNCTIONS (MELLOR, 1959, TABLE 3)
NFR
          7
CS, AMDA, AMDI, AMDZ
  1.180000       0.0000000E+00   15.00000        60.00000
NYS
        101
( XCC(I),      I=1,NYS) & ( FCP(I), I=1,NYS)
 0.0000000E+00  2.4673343E-04  9.8663568E-04  2.2190213E-03  3.9426386E-03
 6.1558187E-03  8.8563859E-03  1.2041628E-02  1.5708417E-02  1.9853175E-02
 2.4471730E-02  2.9559612E-02  3.5111755E-02  4.1122701E-02  4.7586467E-02
 5.4496735E-02  6.1846673E-02  6.9628984E-02  7.7836037E-02  8.6459719E-02
 9.5491491E-02  0.1049225      0.1147434      0.1249445      0.1355157
 0.1464466      0.1577265      0.1693441      0.1812880      0.1935465
 0.2061074      0.2189583      0.2320866      0.2454793      0.2591232
 0.2730048      0.2871104      0.3014261      0.3159377      0.3306310
 0.3454915      0.3605045      0.3756551      0.3909284      0.4063094
 0.4217828      0.4373334      0.4529459      0.4686048      0.4842946
 0.5000000      0.5157054      0.5313953      0.5470542      0.5626667
 0.5782173      0.5936907      0.6090716      0.6243450      0.6394956
 0.6545085      0.6693690      0.6840622      0.6985739      0.7128897
 0.7269953      0.7408769      0.7545208      0.7679134      0.7810418
 0.7938927      0.8064536      0.8187121      0.8306561      0.8422736
 0.8535534      0.8644843      0.8750556      0.8852566      0.8950775
 0.9045085      0.9135402      0.9221640      0.9303710      0.9381534
 0.9455033      0.9524136      0.9588774      0.9648883      0.9704404
 0.9755283      0.9801468      0.9842916      0.9879584      0.9911436
 0.9938442      0.9960573      0.9977810      0.9990134      0.9997532
  1.000000
 0.7713992      0.6610465      0.5506938      0.4860964      0.4402190
 0.4045864      0.3754243      0.3507195      0.3292697      0.3102995
 0.2932796      0.2778321      0.2636781      0.2506056      0.2384500
 0.2270804      0.2163914      0.2062968      0.1967249      0.1876157
 0.1789184      0.1705893      0.1625910      0.1548910      0.1474607
 0.1402750      0.1333116      0.1265507      0.1199746      0.1135673
 0.1073143      0.1012024      9.5219724E-02  8.9355148E-02  8.3598569E-02
 7.7940598E-02  7.2372496E-02  6.6886142E-02  6.1473899E-02  5.6128554E-02
 5.0843354E-02  4.5611825E-02  4.0427864E-02  3.5285547E-02  3.0179225E-02
 2.5103418E-02  2.0052830E-02  1.5022236E-02  1.0006566E-02  5.0008148E-03
 0.0000000E+00 -5.0008306E-03 -1.0006603E-02 -1.5022261E-02 -2.0052856E-02
-2.5103461E-02 -3.0179247E-02 -3.5285573E-02 -4.0427905E-02 -4.5611851E-02
-5.0843373E-02 -5.6128558E-02 -6.1473906E-02 -6.6886187E-02 -7.2372548E-02
-7.7940658E-02 -8.3598629E 02 -8.9355186E-02 -9.5219739E-02 -0.1012025
-0.1073143     -0.1135673     -0.1199746     -0.1265508     -0.1333116
-0.1402749     -0.1474607     -0.1548910     -0.1625911     -0.1705893
-0.1789184     -0.1876157     -0.1967250     -0.2062968     -0.2163915
-0.2270804     -0.2384500     -0.2506059     -0.2636783     -0.2778324
-0.2932798     -0.3102999     -0.3292701     -0.3507200     -0.3754249
-0.4045873     -0.4402193     -0.4860959     -0.5506907     -0.6610565
-0.7714224
TOC
```

FIGURE 5.5-2 (CONTINUE)

```
 0.1011400
FOLLOWING ARE ( ( G(N,K), K=1,NFR), N=1,NFR),
( ( H(N,K), K=1,NFR), N=1,NFR), ( B(N), N=1,NFR), &
( T(N), N=1,NFR), FOR AMDAD=AMDA,AMDZ,AMDI
  1.373091      -0.3730868      -4.7396030E-02  3.4194361E-02 -8.2726954E-03
 2.7298896E-02 -1.0916364E-02  0.0000000E+00  -1.420481       0.0000000E+00
 4.4038385E-02  0.0000000E+00  3.0418137E-02  0.0000000E+00  0.4072816
 0.0000000E+00  -1.057238      0.0000000E+00 -8.1798490E-03  0.0000000E+00
-1.2193669E-02  0.0000000E+00  3.9121512E-02  0.0000000E+00  -1.009937
 0.0000000E+00 -1.7021501E-03  0.0000000E+00 -6.8957144E-03  0.0000000E+00
 6.7244815E-03  0.0000000E+00  -1.001510      0.0000000E+00  1.6851907E-05
 0.0000000E+00 -2.6436450E-03  0.0000000E+00  1.3701854E-03  0.0000000E+00
 -1.000157      0.0000000E+00  9.5066230E-04  0.0000000E+00 -6.3849147E-04
 0.0000000E+00  2.7375855E-04  0.0000000E+00  -1.000018      0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  8.0536567E-02  6.2625073E-03
 2.7139602E-02 -6.4016335E-02 -3.5316154E-02 -4.0477101E-02 -4.4769414E-02
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  0.0000000E+00
  1.347499      -0.3474957      -3.9202519E-02  3.2114621E-02 -8.5944878E-03
 2.5209146E-02 -1.0509614E-02  0.0000000E+00  -1.386697       0.0000000E+00
 3.8775515E-02  0.0000000E+00  2.8209714E-02  0.0000000E+00  0.3796105
 0.0000000E+00  -1.045862      0.0000000E+00 -9.4877388E-03  0.0000000E+00
-1.1747241E-02  0.0000000E+00  3.0606378E-02  0.0000000E+00  -1.005767
 0.0000000E+00 -1.9178378E-03  0.0000000E+00 -6.9056829E-03  0.0000000E+00
 3.6600952E-03  0.0000000E+00  -1.000190      0.0000000E+00 -1.3431325E-05
 0.0000000E+00 -1.9150847E-03  0.0000000E+00  3.4441851E-04  0.0000000E+00
-0.9998317      0.0000000E+00  1.1059480E-03  0.0000000E+00 -2.5798939E-04
 0.0000000E+00 -4.0932264E-05  0.0000000E+00 -0.9999556     -4.2275999E-02
 1.6382568E-02 -2.4923453E-02  8.3927251E-04 -4.4739526E-03  4.5580021E-04
 7.3303141E-05 -5.0039165E-02  2.3007811E-08 -2.8945163E-02  1.8522019E-08
-4.6786284E-03  1.4217595E-08  1.8995389E-04 -7.5997146E-09  1.7127387E-02
-6.5222334E-09  3.0112659E-04 -4.4405821E-09  4.3533192E-04 -3.9832599E-09
 7.2376807E-03 -2.9231069E-09  3.5885754E-03 -2.2523543E-09  6.7295892E-05
-1.7320767E-09 -3.7230671E-05 -4.3239606E-10  5.4625032E-04 -3.9757711E-10
 6.1145436E-04 -3.5193545E-10 -3.7039332E-05 -2.1270107E-10 -5.0880027E-04
 1.6313151E-10 -1.6344152E-04  1.1067427E-10  7.2398361E-05  8.7663113E-11
-1.0959200E-05  3.0393448E-11 -6.8517540E-05  2.5702832E-11 -4.8604950E-05
 2.2095312E-11  1.9509838E-05  1.1200884E-11  7.9579927E-02  4.5247823E-03
 2.6716035E-02 -6.3679554E-02 -3.5268333E-02 -4.0534448E-02 -4.4753395E-02
 0.1274393      1.1902492E-02 -2.6182691E-02 -3.7317555E-03 -8.0651970E-04
-6.1859790E-04 -3.0790437E-03
  1.271879      -0.2718769      -1.5107879E-02  2.5142090E-02 -9.7642224E-03
```

FIGURE 5.5-2 (CONTINUE)

1.9316953E-02	-9.1699949E-03	0.0000000E+00	-1.286983	0.0000000E+00
2.3168059E-02	0.0000000E+00	2.1776550E-02	0.0000000E+00	0.2970188
0.0000000E+00	-1.013133	0.0000000E+00	-1.2784059E-02	0.0000000E+00
-1.0406170E-02	0.0000000E+00	5.3424207E-03	0.0000000E+00	-0.9950062
0.0000000E+00	-1.8803109E-03	0.0000000E+00	-5.8251210E-03	0.0000000E+00
-4.4336440E-03	0.0000000E+00	-0.9975595	0.0000000E+00	1.9819738E-04
0.0000000E+00	5.9427641E-04	0.0000000E+00	-1.7836202E-03	0.0000000E+00
-0.9995413	0.0000000E+00	1.3175826E-03	0.0000000E+00	7.2361820E-04
0.0000000E+00	-4.5466528E-04	0.0000000E+00	-1.000008	-8.0956489E-02
3.2189745E-02	-4.7548648E-02	2.2006668E-03	-8.4162131E-03	9.0854039E-04
1.1111570E-04	-9.4817720E-02	4.3713300E-08	-5.5010512E-02	3.5219902E-08
-9.0458356E-03	2.7035753E-08	3.4457454E-04	-1.5556630E-08	3.4429930E-02
-1.3391153E-08	1.5372403E-03	-9.2404253E-09	7.8572944E-04	-8.1296427E-09
1.2747332E-02	-5.4160925E-09	6.7664273E-03	-4.2513073E-09	5.2098563E-04
-3.2596383E-09	-1.0996135E-04	-3.1677194E-10	8.6005843E-05	-3.2284003E-10
7.9328520E-04	-3.2879249E-10	5.4763095E-05	-1.7059845E-10	-9.7209204E-04
4.2119003E-10	-5.1369460E-04	3.2783812E-10	-1.9341638E-05	2.5003330E-10
1.9595549E-05	-1.6118525E-11	3.7522997E-05	-8.9191718E-12	-6.6533481E-05
-4.6874061E-13	-6.1993569E-06	-8.2880326E-12	7.6744169E-02	-7.1780756E-04
2.5416022E-02	-6.2735349E-02	-3.5170496E-02	-4.0697958E-02	-4.4704270E-02
0.2450204	2.0531975E-02	-4.5227956E-02	-5.2862149E-03	-2.9505172E-03
-1.4394496E-03	-6.1726500E-03			
1.149131	-0.1491305	2.3739813E-02	1.0439388E-02	-1.2137624E-02
1.1075284E-02	-6.6884416E-03	0.0000000E+00	-1.125390	0.0000000E+00
-2.3351838E-03	0.0000000E+00	1.1558891E-02	0.0000000E+00	0.1595692
0.0000000E+00	-0.9634855	0.0000000E+00	-1.5604005E-02	0.0000000E+00
-8.3352849E-03	0.0000000E+00	-3.5877891E-02	0.0000000E+00	-0.9837594
0.0000000E+00	7.4595021E-04	0.0000000E+00	6.3624722E-04	0.0000000E+00
-1.3257822E-02	0.0000000E+00	-0.9977632	0.0000000E+00	1.2244676E-03
0.0000000E+00	5.4492075E-03	0.0000000E+00	-1.8195013E-03	0.0000000E+00
-1.000807	0.0000000E+00	3.6040516E-04	0.0000000E+00	1.2688811E-03
0.0000000E+00	3.3323502E-04	0.0000000E+00	-1.000337	-0.1129456
4.7200758E-02	-6.5922432E-02	4.6494445E-03	-1.1384689E-02	1.2915963E-03
9.5289499E-05	-0.1293276	5.9973004E-08	-7.5543880E-02	4.8408282E-08
-1.2873369E-02	3.7158557E-08	4.4841159E-04	-2.4476087E-08	5.2495953E-02
-2.1176463E-08	4.8442786E-03	-1.4937966E-08	9.8024239E-04	-1.2731615E-08
1.4163221E-02	-6.7496124E-09	8.6882701E-03	-5.5003064E-09	1.6934223E-03
-4.2003649E-09	-1.8684632E-04	1.0416350E-09	-2.4570615E-03	8.6356722E-10
-1.8132523E-04	5.9436805E-10	2.5100540E-04	4.8386645E-10	-8.4184058E-04
6.1454464E-10	-8.1782357E-04	5.4642868E-10	-3.8943885E-04	4.1352169E-10
2.7364616E-05	-3.0873998E-10	5.5368326E-04	-2.6878663E-10	1.5788809E-04
-2.0341992E-10	-3.8205435E-05	-1.5120923E-10	7.2070099E-02	-9.6396524E-03
2.3085548E-02	-6.1363548E-02	-3.5177931E-02	-4.0920448E-02	-4.4606462E-02
0.3441083	2.2130689E-02	-4.9091969E-02	-2.2818651E-03	-8.3887083E-03
-2.7330697E-03	-9.2090582E-03			
0.9826022	1.7396268E-02	7.7120878E-02	-2.0356141E-02	-1.4243137E-02
4.2365617E-03	-4.2757504E-03	0.0000000E+00	-0.9054829	0.0000000E+00
-3.7163317E-02	0.0000000E+00	-2.6489655E-03	0.0000000E+00	-3.7753463E-02
0.0000000E+00	-0.9060732	0.0000000E+00	-1.0528560E-02	0.0000000E+00
-6.5886504E-03	0.0000000E+00	-9.1363207E-02	0.0000000E+00	-0.9869078
0.0000000E+00	1.0417535E-02	0.0000000E+00	2.4593284E-02	0.0000000E+00

FIGURE 5.5-2 (CONTINUE)

-9.9210320E-03	0.0000000E+00	-1.007246	0.0000000E+00	1.4523084E-03
0.0000000E+00	9.9671828E-03	0.0000000E+00	6.0272766E-03	0.0000000E+00
-1.002671	0.0000000E+00	-5.2588726E-03	0.0000000E+00	-2.8012465E-03
0.0000000E+00	1.9369608E-03	0.0000000E+00	-0.9991357	-0.1368254
6.2569410E-02	-7.9283491E-02	9.0393247E-03	-1.3128875E-02	1.3043333E-03
1.5207623E-04	-0.1491351	7.0079793E-08	-8.8539913E-02	5.6797507E-08
-1.6235026E-02	4.3583558E-08	5.4878456E-04	-3.5917090E-08	7.3430695E-02
-3.1325666E-08	1.2264599E-02	-2.2781908E-08	1.0243268E-03	-1.8582393E-08
6.5866397E-03	-4.9736979E-09	6.7125019E-03	-4.5034425E-09	3.6693932E-03
-3.4428904E-09	4.9893584E-05	4.5531392E-09	-7.8372760E-03	4.1115014E-09
-3.7951947E-03	3.2604077E-09	-7.7260469E-05	2.3294098E-09	2.2036436E-03
-5.3185900E-10	8.2511769E-04	-3.1916997E-10	-5.3184887E-04	-2.1207246E-10
-2.8612340E-04	-8.7782304E-10	1.1438091E-03	-8.3455531E-10	9.0125075E-04
-6.9450229E-10	2.8996501E-04	-5.0035526E-10	6.5245569E-02	-2.2823405E-02
1.9044768E-02	-5.9699316E-02	-3.5709277E-02	-4.0989980E-02	-4.4369735E-02
0.4198163	1.0755750E-02	-2.5999840E-02	8.3293729E-03	-2.0982664E-02
-4.1257502E-03	-1.2179918E-02			
999				

964-8-51
081088

In this sample run, Fourier series is to be truncated after the seventh term, the solidity is 1.18, the stagger angle ranges from 0 to 60 degrees at a step of 15 degrees, and the number of data points along the chord is 101. The values of coordinate along the chord and slope of camber are listed in the table. The maximum thickness of the blade is 10.114% of the chord length. Matrices of g_n, h_n, B, and T are listed at the end of Table 5.5-2.

Listed in Table 5.5-3 is the input data file DSN3I.DAT which contains some controlling parameters. Input cards numbers 2 to 4 keeps some controlling data for the overall system. The fluid density, specified as 1000 kg/m^3 in the file, has no effects in the present model if the density is constant. The rotational speed is assumed to be 3475 rpm. The maximum body radius of 6.375 inches, which is converted into a value in meters by multiplying with a conversion factor 0.0254, is used as the reference radius. This reference radius may be set to other convenient values instead of the body radius. The multiplication factors and the convergence criteria used in this example have been determined by many test runs. The iterations should converge as long as the multiplication factors used in the prediction-correction process are reasonable. The number of blade is 11 based on experience. The number of stations in the rotor area, including leading edge and trailing edge, is 3 in correspondence to the value used in Program SCM. A flag is set such that the velocities and other data along a specified cross-section of the blade are obtained from an interpolation process based on the data stored in the file DSN3ZI.DAT.

According to Table 5.5-3, there is only one blade section to be studied. The solidity is assumed to be 1.18. The radius is 0.08582 meters at Station 2.

The solidity mentioned in this study is that defined in the mapped X-Y plane. The input solidity can be initially guessed to be any reasonable value. On this particular section of the blade, it has been tried with an initial guess of 1.50 and 0.50. Both guesses resulted in

FIGURE 5.5-3. DSN3I.DAT, AN INPUT DATA FILE TO PROGRAM DSN3 (MIDP)

```
(3037-DSN3-DSN3I.DAT) INPUT DATA FILE FOR DSN3, TEST SEC. 11          24JUN88
1000., 3475., 6.375, .0254              (2) DEN, RPM, RSTAR, CONVR
.5, 1., .0, 1., .001, .001, .005, 30, 1       (3) CONFB, CONFC, CONFSA, CONFSO, EPSA, EPSS, EPSSO, MMAX, IDBUG
11, 1, 3, 1     (4) NBLADE, NSECR, NSTA, INTERP
SEC. 11 AMONG 11 IN 1984 REPORT (AROUND SEC. J=4 AMONG 5 IN SCM TEST)             (201)
1.18    (202) SIG (INITIAL GUESS WAS 1.50) (same if initial guess is .50)
.085820, 2      (211) RSL, IRSL
END OF DSN3I.DAT
SEC. 10 AMONG 11 IN 1984 REPORT (AROUND SEC. J=4 AMONG 5 IN SCM TEST)             (201)
1.23    (202) SIG (INITIAL GUESS WAS 1.32)
.083868, 2      (211) RSL, IRSL
END OF DSN3I.DAT
SEC.  9 AMONG 11 IN 1984 REPORT (AROUND SEC. J=4 AMONG 5 IN SCM TEST)             (201)
1.28    (202) SIG (INITIAL GUESS WAS 1.37)
.079764, 2      (211) RSL, IRSL
END OF DSN3I.DAT
SEC.  8 AMONG 11 IN 1984 REPORT (AROUND SEC. J=3 AMONG 5 IN SCM TEST)             (201)
1.34    (202) SIG (INITIAL GUESS WAS 1.38)
.075825, 2      (211) RSL, IRSL
END OF DSN3I.DAT
SEC.  7 AMONG 11 IN 1984 REPORT (AROUND SEC. J=3 AMONG 5 IN SCM TEST)             (201)
1.40    (202) SIG (INITIAL GUESS WAS 1.40)
.071860, 2      (211) RSL, IRSL
END OF DSN3I.DAT
SEC.  6 AMONG 11 IN 1984 REPORT (AROUND SEC. J=2 AMONG 5 IN SCM TEST)             (201)
1.50    (202) SIG (INITIAL GUESS WAS 1.43)
.067694, 2      (211) RSL, IRSL
END OF DSN3I.DAT
SEC.  5 AMONG 11 IN 1984 REPORT (AROUND SEC. J=2 AMONG 5 IN SCM TEST)             (201)
1.69    (202) SIG (INITIAL GUESS WAS 1.50)  (N.G., 1.69>>1.50 limit)
.062930, 2      (211) RSL, IRSL
END OF DSN3I.DAT
```

a termination of the computer run with a message to use better solidity values which both rounded to 1.18. The computation would be carried out if the internally computed solidity, based on the chosen camber and incident angle, matches with the input value within the error criterion of 0.005. In this model, one run with trial solidity is usually sufficient to find a suitable value to be used. Note that the input file DSN3ZI.DAT should be replaced with the one with corresponding value of solidity.

Attached to the end of Table 5.5-3 are data used to study other sections in the present study. Results are depicted in the following section.

5.6 RESULTS OF SAMPLE DESIGN

The model is tested with some blade sections as shown in Table 5.5-3. Results for six sections from the middle area to the tip area of the blade are presented here.

Other sections at the hub area have solidity values much larger than 1.5 and therefore are not able to be analyzed in the current version of the model. The reason is that this model utilizes a multiple regression analysis (Section 4.3.3) to find expressions of C_b (β, σ, C_{L1}) and α_d (σ, C_b) based on a set of experimental data which has solidity values range from 0.5 to 1.5. Outside this range, the extrapolated result is unpredictable. In order to solve the problem of high solidity, experimental data for the high solidity case have to be included in the regression analysis. This is not implemented due to the time constraint.

Figures 5.6-1 and 5.6-2 depict the results of peripheral velocity and meridional velocity obtained from the stream-curvature method. There is no pre-rotation and so the peripheral velocity changes from 0 at the leading edge of the rotor to the value at the trailing edge as shown in Figure 5.6-1. The meridional velocity increases as the body is contracted.

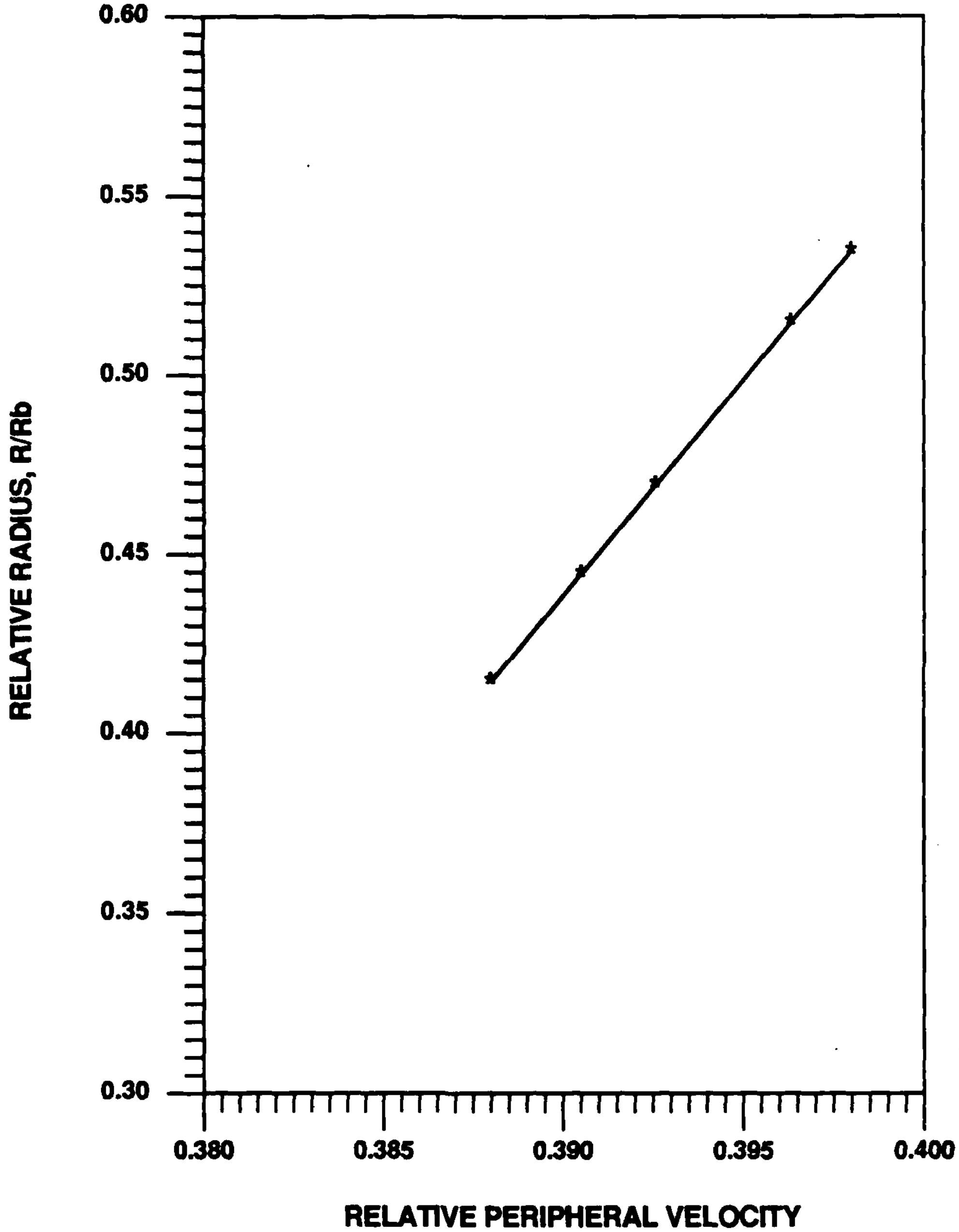


FIGURE 5.6-1. DISTRIBUTION OF PERIPHERAL VELOCITY, C_θ, AT THE TRAILING EDGE OF THE ROTOR

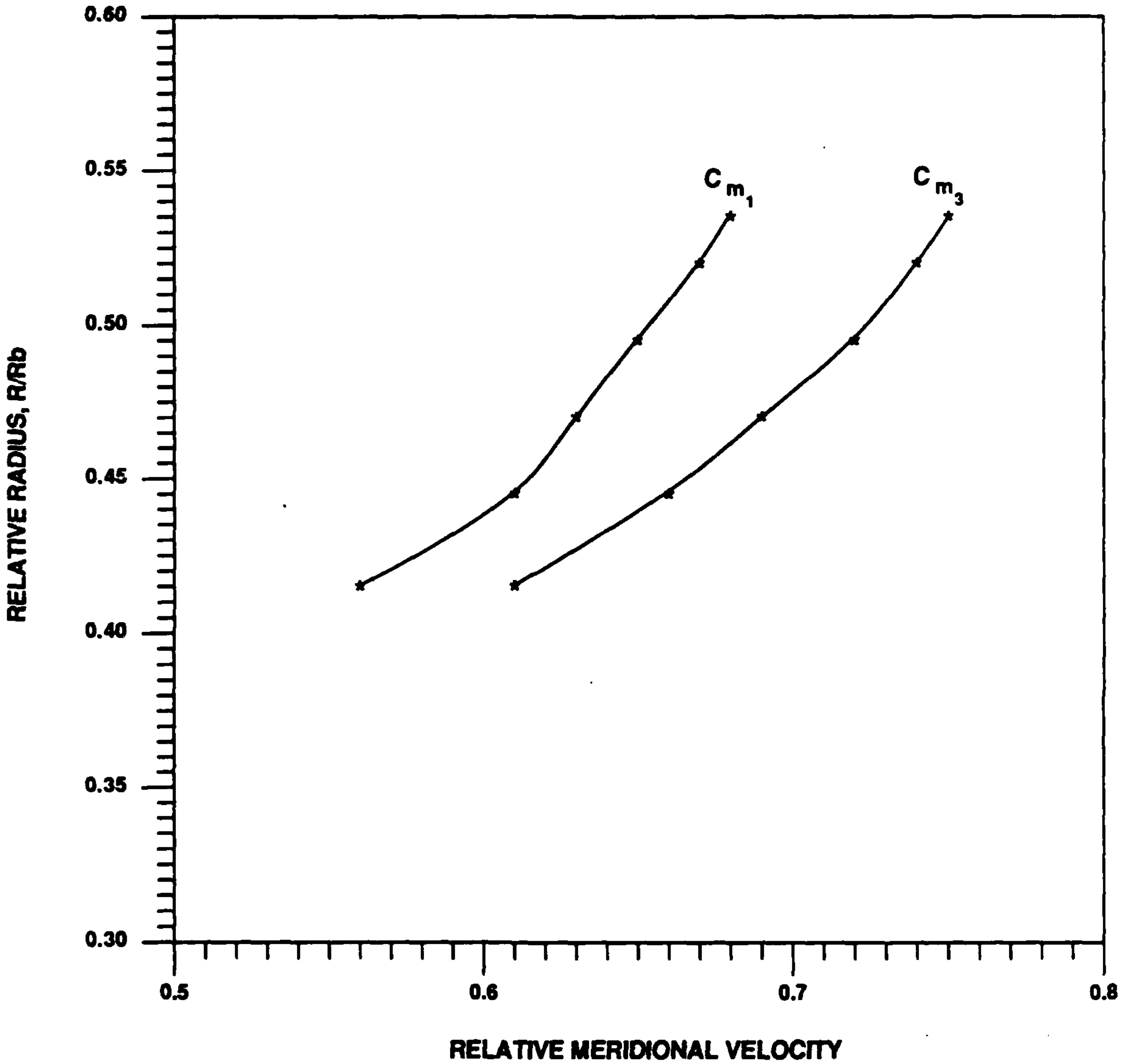


FIGURE 5.6-2. DISTRIBUTION OF MERIDIONAL VELOCITY AT LEADING EDGE, C_{m_1}, AND TRAILING EDGE, C_{m_2}, OF THE ROTOR

Notice that, in Figures 5.6-1 through 5.6-8, the relative radius as the y-axis represents the radius of the streamline at the middle station of the rotor while various data, even if they are those at leading edge or trailing edge, are plotted on the x-axis relative to this radius, curves in Figure 5.6-2 are the velocities at the leading edge and trailing edge which have different radii for the same streamline, but, in Figure 5.6-2, they are both represented by the single radius at the middle station for that streamline.

The flow angle β_1 (Figure 5.6-3) is determined as soon as the flow diagram in X-Y plan is established. The incident angle α_1 (Figure 5.6-4), which depends on flow angle, solidity, and lift coefficient, is obtained through regression analysis based on experimental data. The stagger angle λ (Figure 5.6-5) is simply the difference between β_1 and α_1. Figure 5.6-6 shows the resultant solidity obtained in the mapped X-Y plane. It ranges from 1.18 at the tip to 1.50 before the hub area is reached.

Without the three-dimensional effects, the cascade lift coefficient C_{LM} (Figure 5.6-7), are calculated based on the linear cascade theory discussed in Section 4.3.2. Also shown in the figure is the cascade lift coefficient C_{L1} which is based on the upstream velocity w_1 rather than the mean velocity w_m.

The camber which could produce a turning angle such that C_{LM} or C_{L1} is reached without the three-dimensional effects is shown in Figure 5.6-8 as the curve C_{b2}. With the three-dimensional effects introduced as induced vortices and sources/sinks, the required camber to produce the same turning angle is shown in Figure 5.6-8 as the curve C_{b3}. The required camber is smaller due to the contraction of the body. The differences reduce from 0.20 (12%) near the hub area to 0.11 (8%) near the tip and then suddenly increased to 0.21 (17%) at the tip. The reason for this sudden decrease of the required camber at the tip is not investigated due to the time constraint to finish this report.

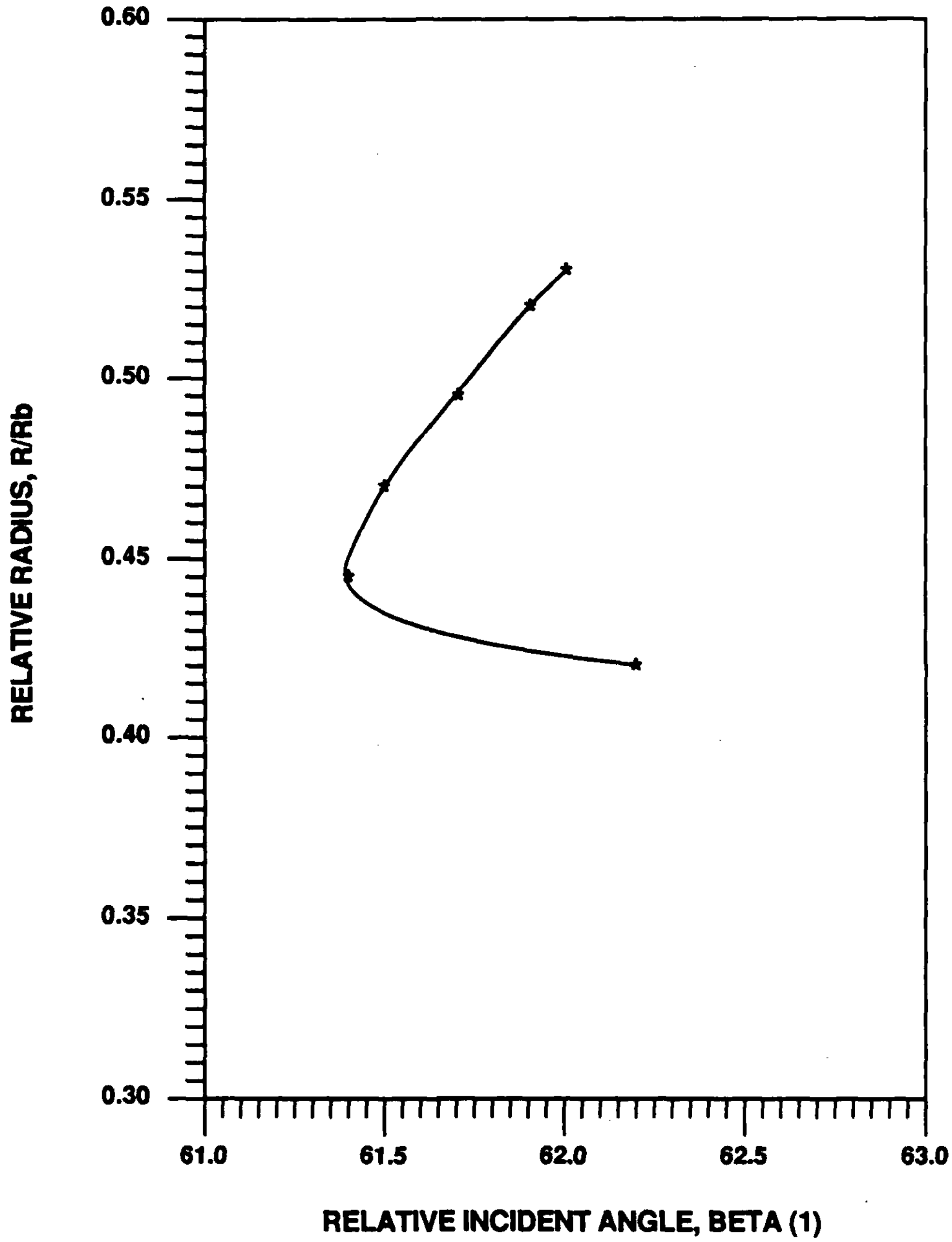


FIGURE 5.6-3. DISTRIBUTION OF RELATIVE FLOW ANGLE, β_1, IN DEGREES, RELATIVE TO X-AXIS, IN X-Y PLANE

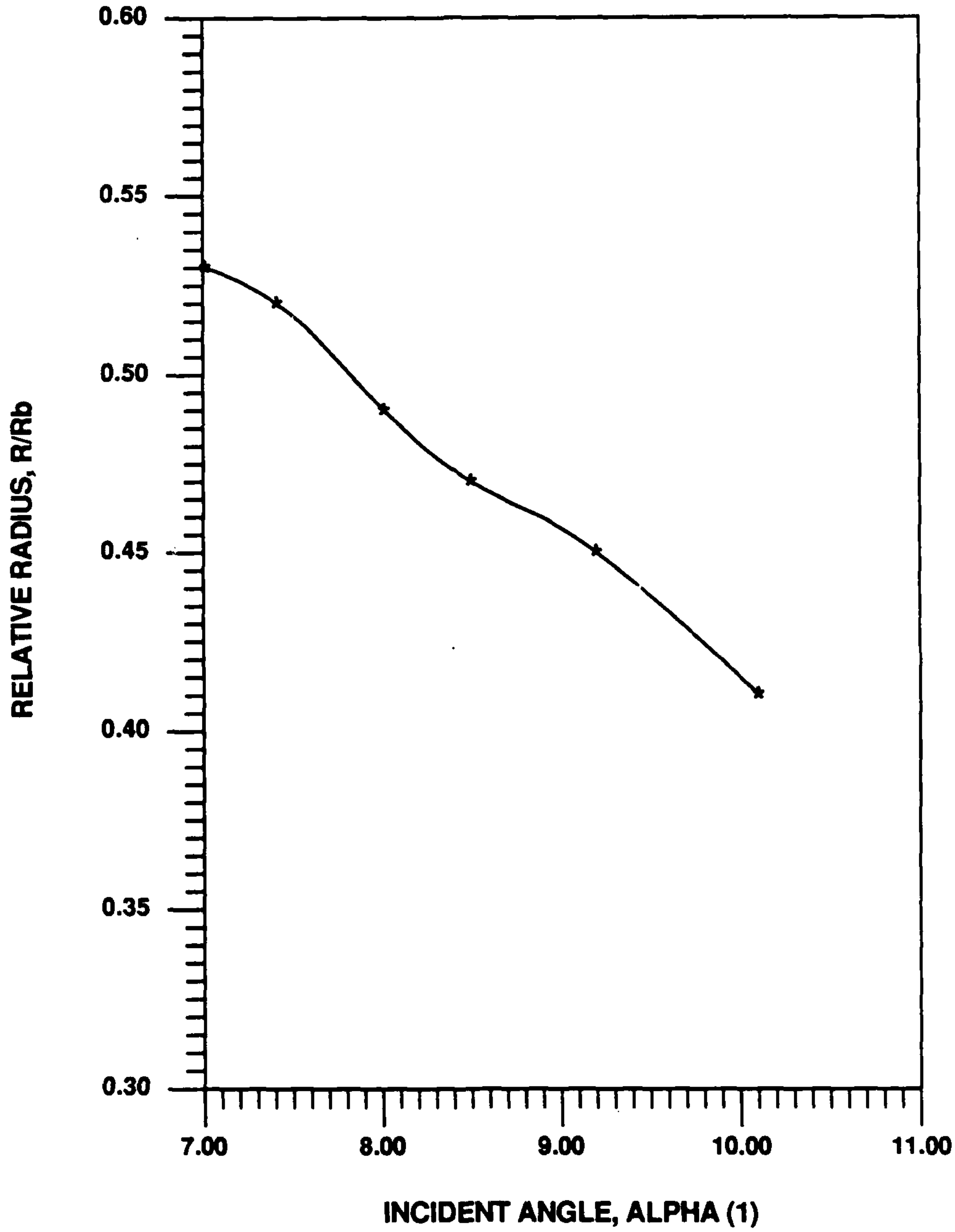


FIGURE 5.6-4. DISTRIBUTION OF INCIDENT ANGLE, χ_1, IN DEGREES, RELATIVE TO THE CHORD LINE, X-Y PLANE

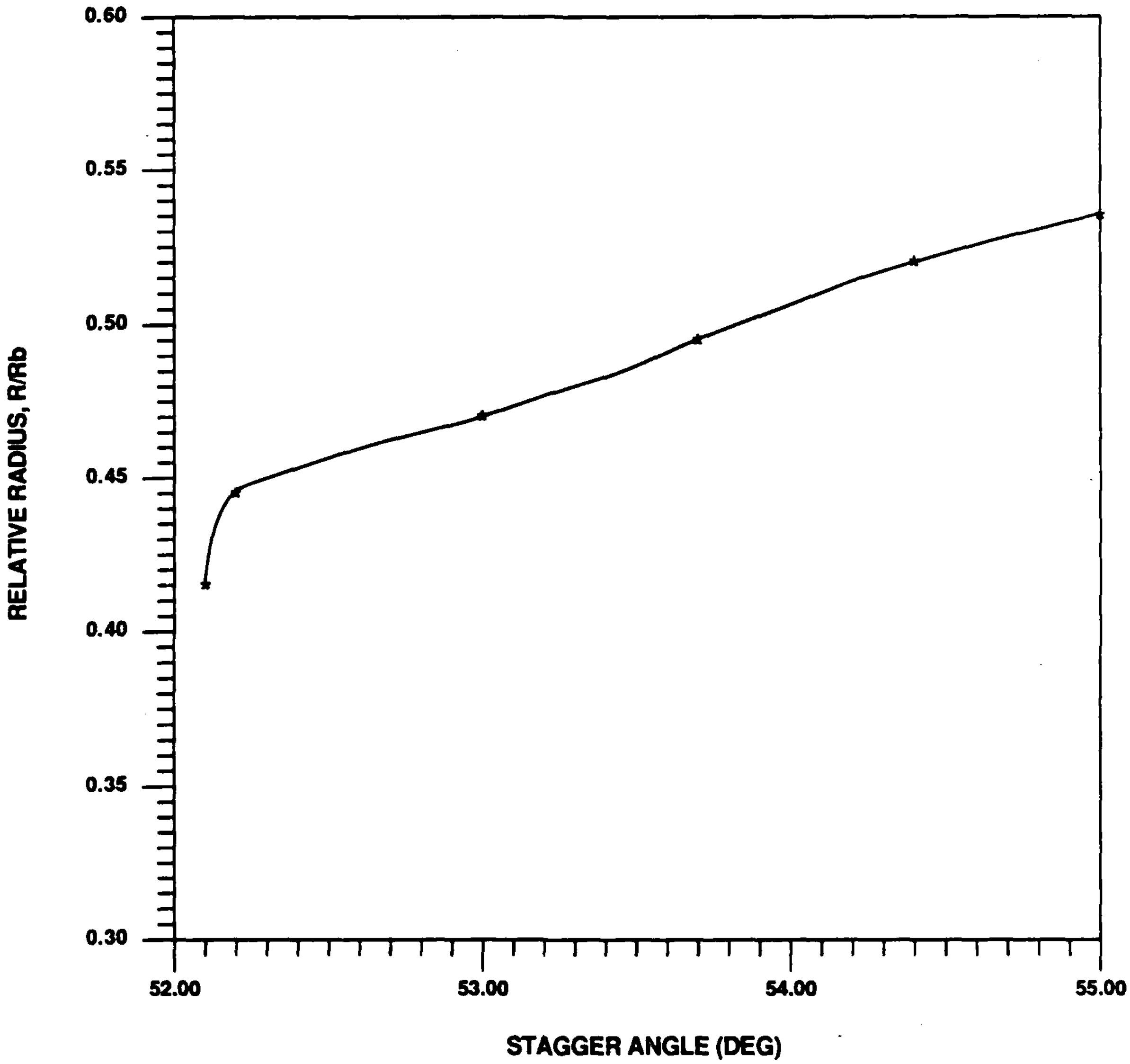


FIGURE 5.6-5. DISTRIBUTION OF STAGGER ANGLE IN X-Y PLANE

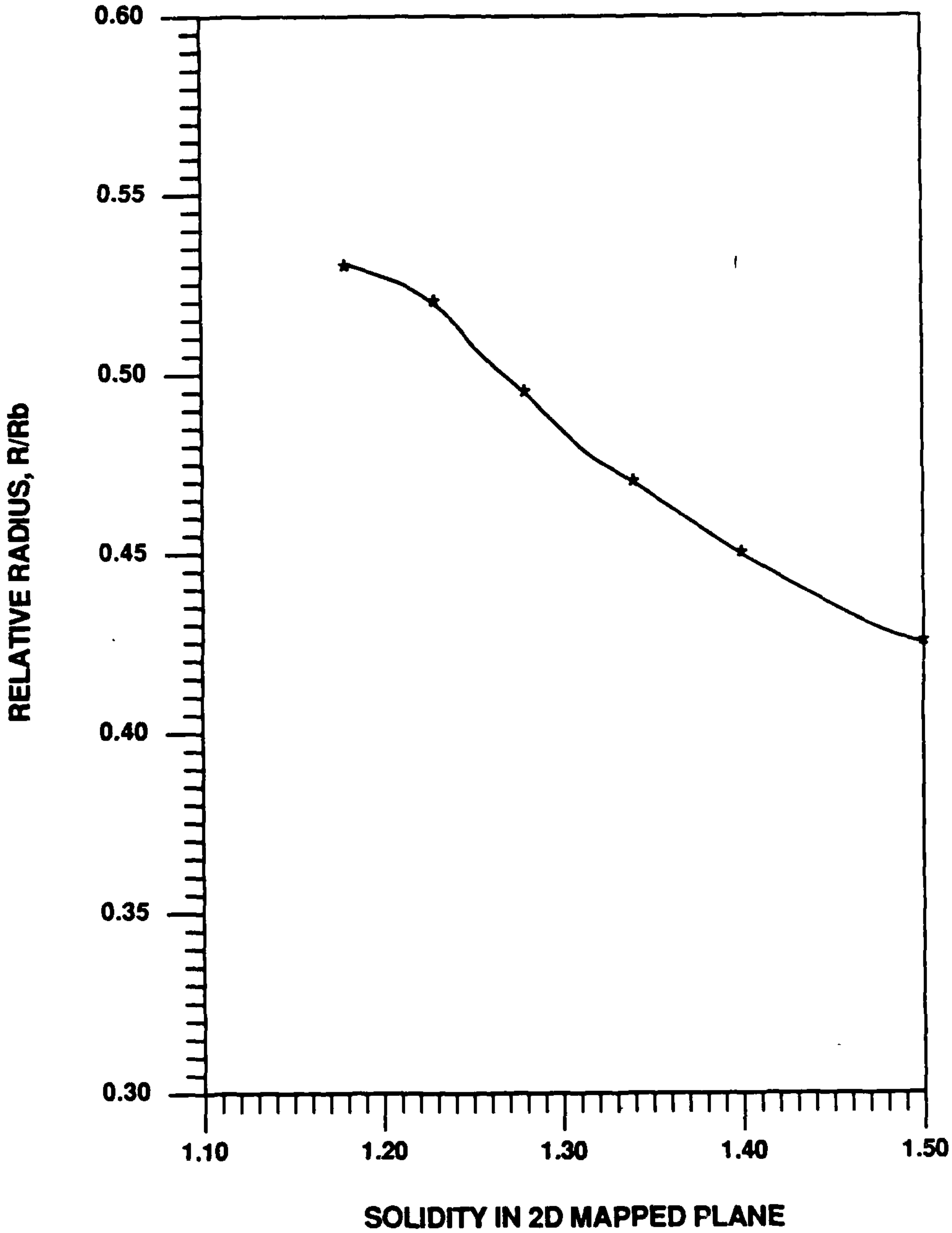


FIGURE 5.6-6. DISTRIBUTION OF SOLIDITY WHEN THE BLADE SECTIONS ARE MAPPED INTO THE X-Y PLANE

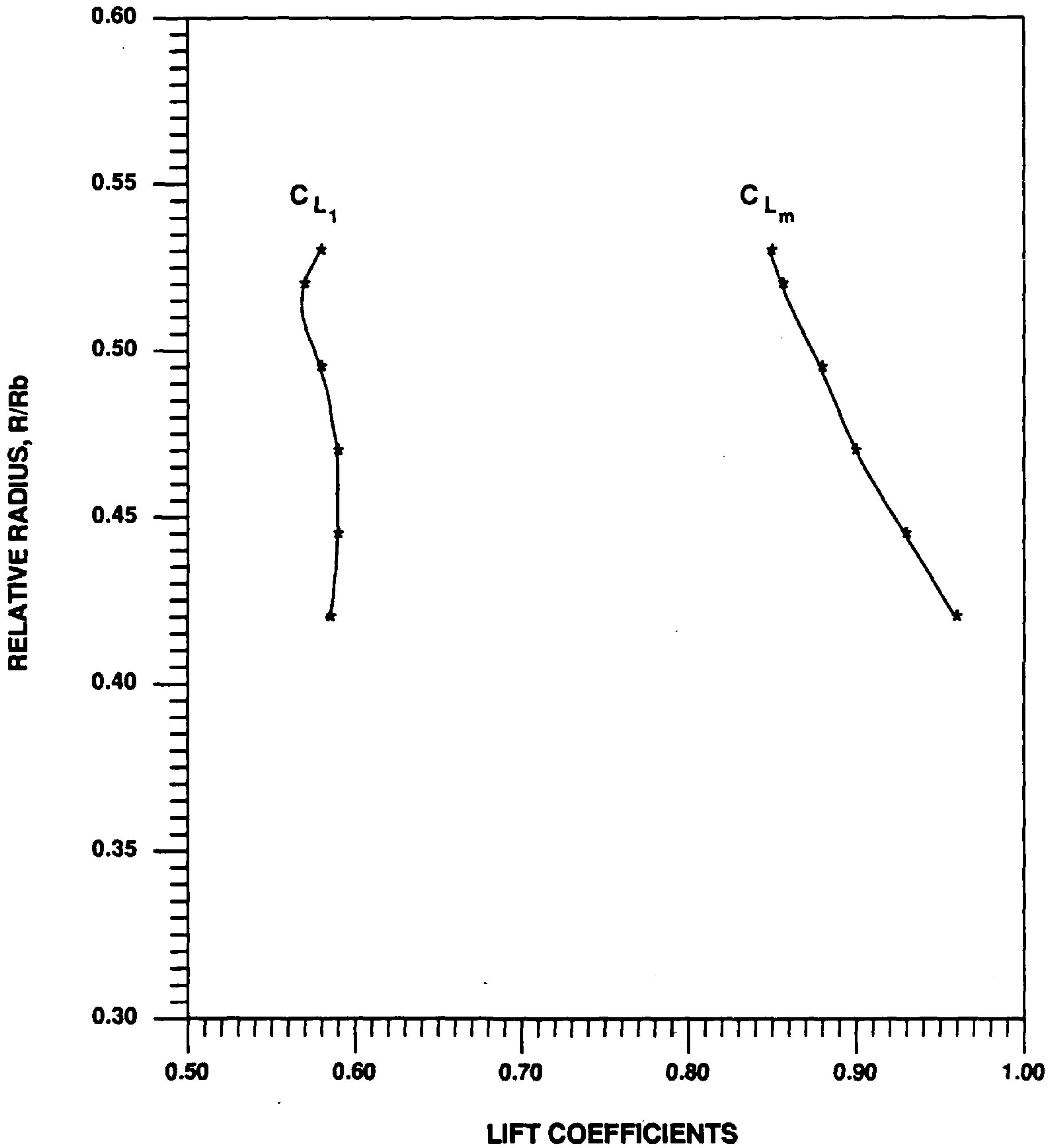


FIGURE 5.6-7. DISTRIBUTION OF REQUIRED LIFT COEFFICIENTS

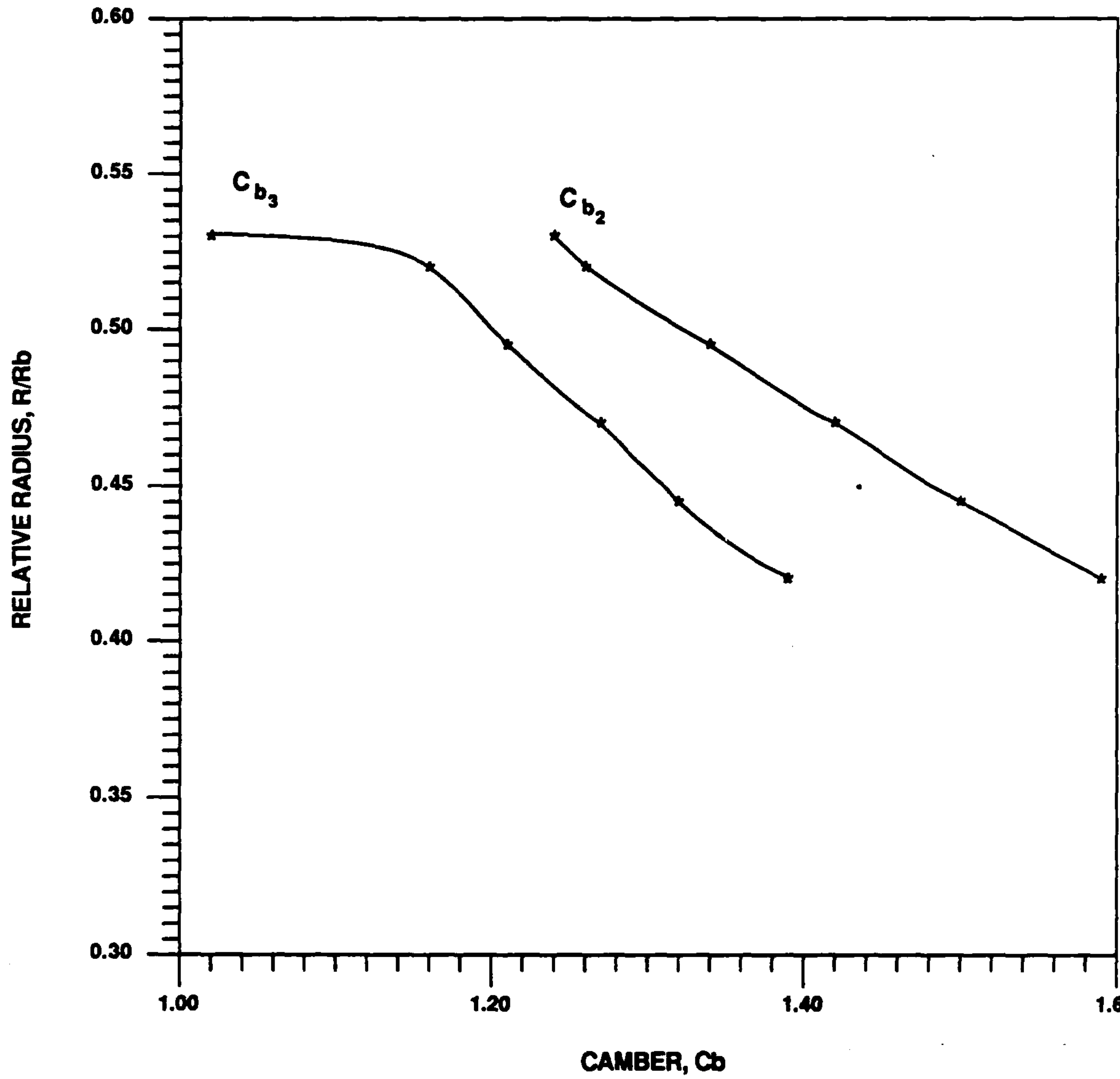


FIGURE 5.6-8. DISTRIBUTION OF REQUIRED CAMBER WITHOUT 3-D EFFECTS, C_{b_2}, AND THAT WHEN THE 3-D EFFECTS ARE PRESENTED, C_{b_3}

6.0 CONCLUSIONS AND RECOMMENDATIONS

The GHR project started in 1985 as a three-year contract with the objective of developing a new pumpjet design theory and then a computer code to satisfy the Navy's need to accurately design pumpjets for high speed underwater vehicles. The design method available then was based on a one-dimensional graphic approach with a simple correction made for the affects of proximity of adjacent blades. The recent need for design of pumpjets for high speed underwater vehicles make such a method totally inappropriate, having led to various hydrodynamic problems.

During the first two years of the contract work, various methodologies were assessed for selecting a candidate method. It has been decided then that a quasi-three-dimensional pumpjet design method is used to fulfill the objective of the contract, i.e., the combination of a blade-through flow analysis with a blade-to-blade flow analysis.

As has been demonstrated in Section 5, the pumpjet design procedure starts with the selection of the underwater vehicle geometry and its goal speed, the flow pattern and velocity profile should then be obtained within by using an axisymmetric turbulent boundary design layer theory or by an experiment. The propulsive efficiency is approximately calculated by using a straight-forward momentum theory with an optimum shroud opening diameter also determined. Once the shroud opening diameter as well as the overall profile shape are chosen, the blade-through flow without (i.e., in this use the streamline curvature method) is used to determine the streamline tube and velocity profile at the rotor section.

The streamline information obtained above will become the key data for the rest of design procedure. A contracting stream tube which represents a design section of the blade is mapped into a two-dimensional plane where an equivalent two-dimensional cascade flow diagram is established. It should be noted that the use of the equivalent flow diagram is made carefully without violating the circulation law; the mass conservation equivalent is violated (or not accurately maintained) but the circulation of the flow is maintained exactly. In order to compensate for such an approximate approach (i.e., use of two-dimensional flow approach for design of a pumpjet in the three-dimensional configuration, the source/sink and vortex are distributed along the row of cascade blades and thus the blade camber is corrected. The cascade theory of Mellor is used in the present analysis with the corrections made by utilizing experimental data. In order to facilitate the design method with the latter assurance, NACA 65 blade profile series are used in the present computer code.

An iterative method is required to determine the blade profile shape in this design method of highly nonlinear nature. Once the blade thickness is determined the streamline curvature method should be used again to correct the flow velocity and streamline tube configuration due to this blockage effect. After this blockage correction on the flow velocity is made, the camber correction is repeated. As noticed by now and explained in Sections 4 and 5, there exists two iterative loops, i.e., camber correction loop and flow velocity correction loop. These iterative loops are repeated until the convergent solution is obtained.

A sample design case has been run by using the computer code developed herein and the procedure has been described in Section 5.

Due to the limited time available, only the camber iteration loop has been used and seemed stable from the viewpoint of the camber determination. However, for exactly the same reason as above, many aspects have not been tested nor incorporated into the computer code. Table 6-1 provides such aspects not tested or unincorporated together with the impact of each aspect on the designed pumpjet blade.

As seen from Table 6-1, the computer code developed herein is still at the stage of development and testing of the detailed aspects. All the theoretical bases for those aspects have been developed and ready to be incorporated into the code. Unfortunately, due to the limited timeframe, the items and aspects stated in Table 6.1 are yet to be incorporated and tested in order to make the present computer code usable and reliable. Although it may not take an enormous amount of time to incorporate each item described above, careful handling of such item is necessary since a slight mistake or mishandling may lead to a fatal error in the outcome of computer code. With such cautions in mind, the present theory computer code should be used.

TABLE 6-1.

VARIOUS ITEMS *NOT* TESTED OR INCORPORATED INTO THE COMPUTER CODE AND THEIR IMPACTS ON THE DESIGNED PUMPJET BLADE

Items	Theory Basis	Impact	Actions to be Taken
1. Induced velocities due to source/sink (u) and vortex (S and X)	Section 4.4	As mentioned in the text, the effects of induced velocities on the camber were partially compared. The order of magnitude and sign seems to be correct, but more extensive testings and verification are necessary to provide confidence. This aspect is essential in the accuracy of the entire computer code and until these features are totally tested, the computer code may not be used with confidence.	More tests for the accuracies of the induced velocities should be made.
2. 2-D Cascade Data at High Solidity	Sections 4.3.2 and 4.3.3	The experimental data for cascade configuration at high solidity(s) are scarce and only those of less than and equal to 1.75 are incorporated in the current program. The result of regression analysis (Eqn. 4.3-20) is incorporated for $\sigma \leq 1.75$ so that the design of pumpjet having a higher solidity (in the two-dimensional plane) may not be possible with the current computer code.	More cascade data of high σ should be incorporated into Eqn. 4.3-20.
3. Smooth Entry Condition	Section 4.3.3	The camber correction made due to the induced velocities and χ may drastically change the smooth entry condition. In the present computer code, this aspect is not incorporated into the computer code so that the flow separation cantation may occur if the blade designed by this computer code is actually used.	A concept for smooth entry should be integrated into the camber iteration loop by changing the stagger angle (γ) of blade.
4. Velocity Calculation Around Blade	Section 4.3.3 of TC-3037 GHR Report (1986)	A singular perturbation method has particularly been developed for the purpose of calculating the velocity perturbations around the blade. This is necessary for determining the possibility of cavitation inception on the blade, but is not incorporated in the present computer code.	The singular perturbation can simply be used. Other blade-to-blade methods may also be used.

964-8-Table 6
092988

TABLE 6-1.

VARIOUS ITEMS NOT TESTED OR INCORPORATED INTO THE COMPUTER CODE AND THEIR IMPACTS ON THE DESIGNED PUMPJET BLADE

Items	Theory Basis	Impact	Actions to be Taken
5. Blade Thickness Correction	Section 4.2.1	The blade thickness cannot be determined until the cascade blade profile is determined. The meridional velocity calculation with SCM does not include such effect in the present computer code. Such blockage effect is substantial since the dynamic head is proportional to the square of velocity.	After the first camber iteration loop, the blade thickness should be incorporated for repeated use of SCM.
6. Flow Skewness	Section 4.4.6	The flow skewness is one of the main subjects in the present development of pumpjet since the effect is of the first order. Without this correction, the flow incidence angle is inaccurate, causing the lift or head generation error of the first order.	A simple exponential approximation to the contraction of stream tube may be used to correct the flow skewness.
7. Secondary Flow Effect	Section 4.4.7	The secondary flow effect is usually represented by the vortex at the exit of blade channel. This is not incorporated in the present computer code.	Eqn. 4.4-41 may be used for correcting the secondary flow effect.
8. Generation of 3-D Blade Profile	Section 4.4.2	The camber profile is determined in the X-Y plane, which should be transformed back to the actual 3-D plane.	The transportation formula in Section 4.4.2 shall be used to perform this work.
9. Performance Verification	Section 3.1	The final pumpjet blade configuration should be tested by an existing performance prediction method or tested by experiments. At least, the former can be done without much expense but has not been exercised in the present project.	The method of Katsanis, et al., may be used for verifying the total accuracy of the present design method.

7.0 REFERENCES

Abbott, I.H., and von Doenhoff, A.E., 1959, Theory of Wing Sections, Including a Summary of Airfoil Data, Dover Publications, Inc., New York, NY.

Bruce, E.P., Gearhart, W.S., Ross, J.R., and Treaster, A.L., 1974, "The design of pumpjets for hydrodynamic propulsion," Fluid Mechanics, Acoustics, and Design of Turbomachinery, Part I, NASA SP-304, 795-839.

Crocco, L., 1937, Z. Angew. Math. Mech., 17,1.

Furuya, O., and Acosta, A.J., 1973, "A note on the calculation of supercavitating hydrofoils with rounded noses," Journal of Fluids Eng., ASME, 95, 221-228.

Furuya, O., and Chiang, W.-L., 1986, "A new pumpject design theory," Report No. TC-3037, Tetra Tech, Inc., Pasadena, CA.

Furuya, O., Chiang, W.-L., and Maekawa, S., 1984, "A hydrodynamic study of the ALWT (MK-50) pumpjet," Tetra Tech Report TC-3275, Tetra Tech, Inc., Pasadena, CA.

Herrig, L.J., Emery, J.C., and Erwin, J.R., 1951, "Systematic two-dimensional cascade tests of NACA 65-series compressor blade at low speeds", NACA RM L51G31, National Advisory Committee for Aeronautics, Washington, D.C.

Inoue, M., Ikui, T., Kamada, Y., and Tashiro, M., 1979, "A design of axial-flow compressor blades with inclined stream surface and varying axial velocity," Bulletin of the JSME, 22 (171), Paper No. 171-4, 1190-1197.

Inoue, M., Ikui, T., Kamada, Y., and Tashiro, M., 1980, "A quasi three-dimensional design of diagonal flow impellers by use of cascade data," IAHR Symposium 1980, Tokyo, 403-414.

Katsanis, T., 1964, "Use of arbitrary quasi-orthogonals for calculating flow distribution in the meridional plane of a turbomachine," NASA TN D-2546, National Aeronautics and Space Administration, Washington, D.C.

Kerwin, J.E., and Leopold, R., 1964, "A design theory for subcavitating propellers", Trans. SNAME, 72.

Mani, R., 1966, "Quasi two-dimensional flows through cascades," Ph.D. Thesis, California Institute of Technology, Pasadena, CA.

Mani, R., and Acosta, A.J., 1968, "Quasi two-dimensional flows through a cascade," Journal of Engineering for Power, ASME, 90(2), 119-128.

Mellor, G.L., 1959, "An analysis of axial compressor cascade aerodynamics; Part I, Potential flow analysis with complete solutions for symmetrically cambered airfoil families; Part II, Comparison of potential flow results with experimental data", Journal of Basic Engineering, 81, 362-378 and 379-386.

von Karman, Th., and Burgers, J.M., 1963, "General aerodynamic theory-perfect fluids", In: Durand, W.F. (ed.), Aerodynamic Theory, Volume II, General Aerodynamic Theory, Perfect Fluids, Division E, 1-367, Dover Publications, Inc., New York, NY.

Weinig, F.S., 1964, "Theory of two-dimensional flow through cascades", In: Hawthorne, W.R. (ed.), Aerodynamics of Turbines and Compressors, Section B, 13-82, Princeton University Press, Princeton, NJ.

Wislicenus, G.F., 1965, Fluid Mechanics of Turbomachinery, Volumes I and II, Dover Publications, Inc., New York, NY.

Van Dyke, M., 1975, Perturbation Methods in Fluid Mechanics, Annotated edition, The Parobolic Press, Stanford, CA.

APPENDIX A

A BRIEF DESCRIPTION OF COMPUTER CODES

Four independent but related computer programs are used in this study. They are named SCM, RIS, MELLOR, and DSN3.

The principal program is the main integration design program (MIDP) which is named DSN3 in the computer code. Programs SCM and MELLOR are used to prepare data for DSN3. Program RIS is used to prepare data for MELLOR.

The stream curvature method (Sec. 4.2) is used in Program SCM to determine streamline pattern and then calculate velocities and other parameters along any streamline. Non-uniform inflow condition is allowed at the upstream. The input data is provided by SCMI.DAT. The output file DSN3IZ.DAT serves to input data to Program DSN3. The computer code SCM is listed in Appendix B.1. Sample input and output data are discussed in Section 5.3.

A small program RIS is designed to prepare the cascade influence functions R and I (Mellor, 1959) which are required in Program Mellor. With a suitable range of solidity and stagger angle, Program RIS is needed only once for many computer runs of Program MELLOR. Program RIS is listed in Appendix B.2 with input data listed in Table 5.4.1.

Program Mellor, which is listed in Appendix B.3, prepared several tables of functions required in the linear cascade theory which is discussed in Section 4.3.2. The input data include RI.DAT, BLADE.DAT, and MELI.DAT, as shown in Tables 5.4.2, 5.4.3, and 5.4.4. The output MELO10.DAT is to be used by Program DSN3.

Program DSN3 (MIDP) is the essential part of this study. The code is listed in Appendix B.4. The main flow chart is depicted in Figure 4.27, while some detailed flow charts are shown in Figures 4.28 to 4.32. Based on the velocity distributions obtained from Program SCM, the velocity diagram on the mapped X-Y plane is constructed. From the turning angle shown in the velocity diagram, the cascade lift coefficient is determined and then the camber which could produce this lift coefficient is found from a relationship based on experimental data. Finally, three-dimensional effects are simulated by induced vortices and sources/sinks, and then camber is adjusted to produce the original turning angle which has a one-to-one relationship with the required lift coefficient.

APPENDIX B

LISTING OF COMPUTER CODES

B.1 PROGRAM SCM

Provided in this section is a listing of the computer program SCM and the following subroutines:

AKIMAI
AKIMAO
BCMLIN
ENTHAL
INTSPL
MLINE
NUF
NUFLOW
QHIRSH
QRZ
SOLOUT
SPLIN3
and SPLINE.

```
C (3037-SCM) (V3.1) STREAM CURVATURE METHOD
C            CALCULATING VEL. DISTRIBUTIONS AFTER ADJUSTING STREAMLINES
C V2.1 - A RUNNING VERSION BASED ON INOUE'S LISTING, 06DEC85
C V2.2 - VAX VERSION OF VERSION 2.1, 08DEC85
C V2.3 - PREPARING FOR NON-UNIFORM INFLOW COMPUTATIONS, 20DEC85
C V2.4 - KICK A SMALL BUG OUT FROM VS. 2.3, 08JAN86
C V2.5 - IN QHIRTH, DEFINE AREAH(I), THEN USE CM(I,1)=G/(AREAH(I)*2*DEFREF)
C        INSTEAD OF CM(I,1)=G/(2*PI*RAV*Q(I,NM),DENREF), 21JAN86
C V2.6 - USE CM(I,J)=DG(J)/(AREA(I,J)*DENREF) IN QHIRTH, 22JAN86;
C        INCLUDE SUB. "NUFLOW," 07MAY86
C V2.7 - Q CALCULATED FROM R & RUMRQ, ASSUMING STRAIGHT Q-LINES, INSTEAD
C        OF INPUT, 19MAY86
C V3.0 - ALLOW NON-UNIFORM INFLOW, 10JUN86; TOUCHED 21JUL86
C V3.1 - ADDED OPTION TO OUTPUT RESULTS FOR (3037-DSN3), 19OCT87;
C        TOUCHED 16JUN88

C REQUIRES SUBS. AKIMAI, AKIMAO, BCMLIN, ENTHAL, INTSPL, MLINE, NUF,
C       NUFLOW, QHIRSH, QRZ, SOLOUT, SPLIN3, & SPLINE
C REFERENCES BBFLOW, DATOUT, FOR$DATE_T_DS, FOR$OPEN, NUFLOW, QHIRSH, &
C       SOLOUT.
C TO BE COMPILED BY VAX FORTRAN V4.0-2.

C IBFLOW = -2 IF G, CM(*), & R(*) ARE INPUT DATA;
C        = -1 FOR NONUNIFORM (OR UNIFORM) INFLOW... CALC. G, CM(*), &
C         R(*) INTERNALLY
C        =  0 FOR UNIFORM INFLOW;
C        =  1 IF DATA AVAILABLE FROM BLADE TO BLADE CALCULATIONS.
C IDSN3  = 0 FOR NO SPECIAL ACTION;
C        = 1 FOR WRITING RESULTS INTO FILE DSN3ZI.DAT TO BE USED BY
C            (3037-DSN3)
C LG     = 0 FOR SEA WATER (DEN=1025 KG/CU. M.);
C        = 1 FOR PURE WATER (DEN=1000);
C        = 2 FOR GAS.

      CHARACTER VS*5, TODAY*9

      DATA      VS/'3.1'/

C       AMONG B(BBFLOW), DAT(DATOUT), DEN(DENSI), E(ENTHAL), NUF, NUFLOW,
C               QH(QHIRSH), QRZ, & S(SOLOUT):
C       COMMON/DESIGN/( B, DAT,                                   S)
C       COMMON/GASCON/( B, DAT, DEN,                              S)
C       COMMON/LOAD/  ( B, DAT,                                   S)
C       COMMON/MFLOW/ ( B, DAT,      E, NUF, NUFLOW, QH,          S)
C       COMMON/MLIN/  ( B, DAT,      E, NUF, NUFLOW, QH,          S)
C       COMMON/NONAXI/( B, DAT,              NUFLOW, QH,          S)
C       COMMON/QLIN/  ( B, DAT,      E, NUF, NUFLOW, QH,          S)
C       COMMON/ROTO/  ( B, DAT,                                   S)
C       COMMON/SCROLL/(    DAT,                                   S)
C       COMMON/STAN/  (    DAT,                                   S)
C       COMMON/XY/    ( B, DAT,         NUF, NUFLOW, QH,          S)

C       COMMON/IQ/    ( B,                   NUFLOW, QH            )
C       COMMON/IQS/   ( B,                   NUFLOW, QH,          S)
C       COMMON/NU/    ( B,              NUF, NUFLOW,     QRZ       )
C       COMMON/QS/    (                              QH,          S)

      IN       =5

C     OPEN (  2, FILE='DSN3ZI', STATUS='NEW')... in SOLOUT
      OPEN ( IN, FILE='SCMI',   STATUS='OLD')
      OPEN (  6, FILE='SCMO',   STATUS='NEW')
C     OPEN (  8, FILE='SCMJ',   STATUS='NEW')... in NUFLOW

      CALL    DATE (TODAY)

      WRITE(6,110) VS, TODAY
  110 FORMAT(' OUTPUT FROM SCM.FOR, VERSION ',A5,17XA9//)

      READ (IN,*)  IBFLOW, IDSN3, LG                                READO
      WRITE(6,120) IBFLOW, IDSN3, LG
  120 FORMAT(' IBFLOW ='I3/' IDSN3 ='I3/'      LG ='I3)

      IF (IBFLOW.EQ.10) THEN

        CALL    BBFLOW(IN)
```

```
0078           CALL    DATOUT(IBFLOW)
0079
0080          ELSE
0081
0082           CALL    NUFLOW( IBFLOW, IDSN3, IN, LG)
0083
0084          END IF
0085
0086         CALL    QHIRSH( IBFLOW, LG)
0087
0088         CALL    SOLOUT( IDSN3, IBFLOW, LG, TODAY)
0089
0090         WRITE(6,900)
0091     900 FORMAT(/////' ***END OF OUTPUT***')
0092         STOP 'Done...'
0093         END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	259	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	128	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	200	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	587	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SCMM$MAIN	

VARIABLES

Address	Type	Name	Attributes	References						
2-00000014	I*4	IBFLOW		70=	71	74	78A	82A	86A	88A
2-00000018	I*4	IDSN3		70=	71	82A	88A			
2-00000010	I*4	IN		58=	61A	70	76A	82A		
2-0000001C	I*4	LG		70=	71	82A	86A	88A		
2-00000005	CHAR	TODAY		35	65A	67	88A			
2-00000000	CHAR	VS		35	37D	67				

LABELS

Address	Label	References	
1-00000011	110'	67	68#
1-0000003A	120'	71	72#
1-00000064	900'	90	91#

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References	
	BBFLOW	76	
	DATOUT	78	
	FOR$DATE_T_DS	65	
	FOR$OPEN	61	62
	NUFLOW	82	
	QHIRSH	86	
	SOLOUT	88	

```
C
C+++++++++    SUBROUTINE AKIMAI    +++++++++
C
        SUBROUTINE AKIMAI(X,Y,N,XI,NI,YI,Y1I,W,NW,ICON)

C INPUT N, NI, NW, X(*), XI(*), & Y(*)
C OUTPUT ICON, W(*), YI(*), & Y1I(*)
C TO BE COMPILED BY VAX FORTRAN V4.0-2
C TOUCHED 30MAY86

        DIMENSION X(N),Y(N),XI(NI),YI(NI),Y1I(NI),W(NW)

C      IF ( NW.LT.(N+N+3) ) THEN
C       PRINT *, 'NW.LT.N+3...', NW, N
C       STOP 'AKIMAI.5'
C       ICON    =-1
C       GOTO 999
C       END IF

      DO      I=2,N
        IF ( X(I-1).GE.X(I) ) THEN
          PRINT *, 'X(I-1).GE.X(I)', I, X(I-1), X(I)
          IF (N.LE.50) PRINT *, ( X(K), K=1,N)
          STOP 'AKIMAI.15'
C         ICON  =1
          END IF
        END DO

        NWM=NW/2+2
        DO 20 I=1,N-1
        K=I+2
20      W(K)=(Y(I+1)-Y(I))/(X(I+1)-X(I))
        W(2)=2.*W(3)-W(4)
        W(1)=2.*W(2)-W(3)
        W(N+2)=2.*W(N+1)-W(N)
        W(N+3)=2.*W(N+2)-W(N+1)
        DO 30 I=1,N
        K=I+2
        AM21=ABS(W(K-1)-W(K-2))
        AM43=ABS(W(K+1)-W(K))
        AM=AM21+AM43
        IF(AM.LT.1.0E-5) GO TO 31
        W(NWM+I)=(AM43*W(K-1)+AM21*W(K))/AM
        GO TO 30
31      W(NWM+I)=(W(K-1)+W(K))/2.
30      CONTINUE
        DO 40 J=1,NI
        IF(XI(J).GT.X(N)) GO TO 100
        IF(X(1)-XI(J)) 42,41,100
41      YI(J)=Y(1)
        Y1I(J)=W(NWM+1)
        GO TO 40
42      DO 45 I=2,N
        IF(X(I)-XI(J)) 45,46,47
46      YI(J)=Y(I)
        Y1I(J)=W(NWM+I)
        GO TO 40
47      K=I+1
        P0=Y(I-1)
        P1=W(NWM+I-1)
        P2=(3.*W(K)-2.*W(NWM+I-1)-W(NWM+I))/(X(I)-X(I-1))
        P3=(W(NWM+I-1)+W(NWM+I)-2.*W(K))/(X(I)-X(I-1))**2
        DX=XI(J)-X(I-1)
        YI(J)=P0+DX*(P1+DX*(P2+DX*P3))
        Y1I(J)=P1+DX*(2.*P2+DX*3.*P3)
        GO TO 40
45      CONTINUE
100     PRINT *, 'XI(1).GT.XI(J) .OR. XI(J).GT.X(N)',
     +         XI(1), XI(J), J, X(N), N
        STOP 'AKIMAI.1100'
C       ICON=ICON+10
40      CONTINUE
        ICON=0

C 999   RETURN
        RETURN
        END
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 $CODE	1544	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 $PDATA	67	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 $LOCAL	296	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
Total Space Allocated	1907	

ENTRY POINTS

Address	Type	Name	References
0-00000000		AKIMAI	4#

VARIABLES

Address	Type	Name	Attributes	References							
**	R*4	AM		41=	42	43					
**	R*4	AM21		39=	41	43					
**	R*4	AM43		40=	41	43					
**	R*4	DX		63=	64(3)	65(2)					
**	I*4	I		20=	21(2)	22(3)	30=	31	32(4)	37=	38
				43	45	53=	54	55	56	58	59
				60	61(4)	62(4)	63				
AP-00000028@	I*4	ICON		4	73=						
2-0000000C	I*4	J		47=	48	49	50	51	54	55	56
				63	64	65	68(2)				
**	I*4	K		23(2)=	31=	32	38=	39(2)	40(2)	43(2)	45(2)
				58=	61	62					
AP-0000000C@	I*4	N		4	11(2)	20	23(2)	30	35(3)	36(3)	37
				48	53	68(2)					
AP-00000014@	I*4	NI		4	11(3)	47					
AP-00000024@	I*4	NW		4	11	29					
**	I*4	NWM		29=	43	45	51	56	60	61(2)	62(2)
**	R*4	P0		59=	64						
**	R*4	P1		60=	64	65					
**	R*4	P2		61=	64	65					
**	R*4	P3		62=	64	65					

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
AP-00000020@	R*4	W		**	(*)	4	11	32=	33(3)=	34(3)=
						35(3)=	36(3)=	39(2)	40(2)	43(3)=
						45(3)=	51	56	60	61(3)
						62(3)				
AP-00000004@	R*4	X		**	(*)	4	11	21(2)	22(2)	23
						32(2)	48	49	54	61(2)
						62(2)	63	68		
AP-00000010@	R*4	XI		**	(*)	4	11	48	49	54
						63	68(2)			
AP-00000008@	R*4	Y		**	(*)	4	11	32(2)	50	55
						59				
AP-0000001C@	R*4	Y1I		**	(*)	4	11	51=	56=	65=
AP-00000018@	R*4	YI		**	(*)	4	11	50=	55=	64=

LABELS

Address	Label	References				
**	20	30	32#			
0-00000340	30	37	44	46#		
0-0000030C	31	42	45#			
0-0000057D	40	47	52	57	66	72#
**	41	49	50#			

0-000003E0	42	49	53#	
0-00000588	45	53	54	67#
**	46	54	55#	
0-00000458	47	54	58#	
0-0000058F	100	48	49	68#

```
C
C+++++++++    SUBROUTINE AKIMAO    +++++++++
C
          SUBROUTINE AKIMAO(X,Y,N,Y1,Y2,W,NW,ICON)

C INPUT N, NW, X(*), & Y(*); OUTPUT ICON, W(*), Y1(*), & Y2(*)
C TO BE COMPILED BY VAX FORTRAN V4.0-2
C TOUCHED 10JUN86

          DIMENSION X(N),Y(N),Y1(N),Y2(N),W(NW)

C        IF ( NW.LT.(N+3) ) THEN
C         PRINT *, 'NW.LT.N+3...', NW, N
C         STOP 'AKIMAO.5'
C         ICON    =-1
C         GOTO 999
C         END IF

        DO       I=2,N
          IF ( X(I-1).GE.X(I) ) THEN
            PRINT *, 'X(I-1).GE.X(I)', I, X(I-1), X(I)
            IF (N.LE.50) PRINT *, ( X(K), K=1,N)
            ICON  =1
            GOTO 999
            END IF
          END DO

          DO 20 I=1,N-1
          K=I+2
20        W(K)=(Y(I+1)-Y(I))/(X(I+1)-X(I))
          W(2)=2.*W(3)-W(4)
          W(1)=2.*W(2)-W(3)
          W(N+2)=2.*W(N+1)-W(N)
          W(N+3)=2.*W(N+2)-W(N+1)
          DO 30 I=1,N
          K=I+2
          AM21=ABS(W(K-1)-W(K-2))
          AM43=ABS(W(K+1)-W(K))
          AM=AM21+AM43
          IF(AM.LT.1.0E-5) GO TO 31
          Y1(I)=(AM43*W(K-1)+AM21*W(K))/AM
          GO TO 30
31        Y1(I)=(W(K-1)+W(K))/2.
30        CONTINUE
          DO 40 I=1,N
          Y2(I)=0.0
          K=I+2
          IF(I.NE.1) Y2(I)=2.*(Y1(I-1)+2.*Y1(I)-3.*W(K-1))/(X(I)-X(I-1))
          IF(I.NE.N) Y2(I)=Y2(I)
     +                   +2.*(3.*W(K)-2.*Y1(I)-Y1(I+1))/(X(I+1)-X(I))
40        CONTINUE
          DO 45 I=2,N-1
45        Y2(I)=Y2(I)/2.
          ICON=0
  999     RETURN
          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1079	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	14	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	228	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	1321	

ENTRY POINTS

Address	Type	Name	References
0-00000000		AKIMAO	4#

VARIABLES

Address	Type	Name	Attributes	References							
**	R*4	AM		39=	40	41					
**	R*4	AM21		37=	39	41					
**	R*4	AM43		38=	39	41					
**	I*4	I		19=	20(2)	21(3)	28=	29	30(4)	35=	36
				41	43	45=	46	47	48(6)	49(7)	52=
				53(2)							
AP-00000020@	I*4	ICON		4	23=	54=					
**	I*4	K		22(2)=	29=	30	36=	37(2)	38(2)	41(2)	43(2)
				47=	48	49					
AP-0000000C@	I*4	N		4	10(4)	19	22(2)	28	33(3)	34(3)	35
				45	49	52					
AP-0000001C@	I*4	NW		4	10						

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
AP-00000018@	R*4	W		**	(*)	4	10	30=	31(3)=	32(3)=
						33(3)=	34(3)=	37(2)	38(2)	41(2)
						43(2)	48	49		
AP-00000004@	R*4	X		**	(*)	4	10	20(2)	21(2)	22
						30(2)	48(2)	49(2)		
AP-00000008@	R*4	Y		**	(*)	4	10	30(2)		
AP-00000010@	R*4	Y1		**	(*)	4	10	41=	43=	48(2)
						49(2)				
AP-00000014@	R*4	Y2		**	(*)	4	10	46=	48=	49(2)=
						53(2)=				

LABELS

Address	Label	References		
**	20	28	30#	
0-0000030A	30	35	42	44#
0-000002DC	31	40	43#	
**	40	45	51#	
**	45	52	53#	
0-00000436	999	24	55#	

```
0001    C
0002    C+++++++++    SUBROUTINE BCMLIN    +++++++++
0003    C
0004              SUBROUTINE BCMLIN(NQ,NM,Q,PHIR,CUR)
0005
0006    C INPUT CUR(*), NM, NQ, & PHIR(*), Q(*); OUTPUT CUR(*), & PHIR(*)
0007    C TO BE COMPILED BY VAX FORTRAN V4.0-2
0008    C TOUCHED 10JUN86
0009
0010            PARAMETER ( NQMX=14)
0011
0012            DIMENSION CUR(NQMX,NM), PHIR(NQMX,NM), Q(NQMX,NM)
0013
0014              DO 10 I=1,NQ,NQ-1
0015              DO 10 J=2,NM-1
0016              PHIR(I,J)=PHIR(I,1)+(PHIR(I,NM)-PHIR(I,1))/Q(I,NM)*Q(I,J)
0017              CUR(I,J)=CUR(I,1)+(CUR(I,NM)-CUR(I,1))/Q(I,NM)*Q(I,J)
0018    10        CONTINUE
0019              RETURN
0020              END
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 $CODE	380	PIC CON REL LCL SHR EXE RD NOWRT LONG
2 $LOCAL	168	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
Total Space Allocated	548	

ENTRY POINTS

Address	Type	Name	References
0-00000000		BCMLIN	4#

VARIABLES

Address	Type	Name	Attributes	References				
**	I*4	I		14=	16(6)	17(6)		
**	I*4	J		15=	16(2)	17(2)		
AP-00000008@	I*4	NM		4	12(3)	15	16(2)	17(2)
AP-00000004@	I*4	NQ		4	14(2)			

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References			
AP-00000014@	R*4	CUR		**	(14, *)	4	12	17(4)=	
AP-00000010@	R*4	PHIR		**	(14, *)	4	12	16(4)=	
AP-0000000C@	R*4	Q		**	(14, *)	4	12	16(2)	17(2)

PARAMETER CONSTANTS

Type	Name	References	
I*4	NQMX	10#	12(3)

LABELS

Address	Label	References		
**	10	14	15	18#

```
0001    C
0002    C+++++++++         SUBROUTINE ENTHAL          +++++++++
0003    C
0004          SUBROUTINE ENTHAL( CPT, NQI1, NQIB, PHO, VO, VOC)                  IN
0005
0006    C (3037-ENTHAL) DETERMINE ENTHALPY H
0007    C REFERENCED BY NUFLOW; REFERENCES none
0008    C TO BE COMPILED BY VAX FORTRAN V4.0-2
0009    C CODED BY W.-L. CHIANG, 06JUN86; REVISED 09JUN86
0010
0011          PARAMETER ( NMMX=11, NQMX=14)
0012
0013          DIMENSION CPT(NMMX)
0014
0015          COMMON/MFLOW/     GS(NMMX), G, OMG,
0016         +  CM(NQMX,NMMX), CTH(NQMX,NMMX), RCTH(NQMX,NMMX), H(NQMX,NMMX)
0017    CNOT     +       , A(NQMX,NMMX), DEN(NQMX,NMMX)
0018    CNOT     +       , ENT(NQMX,NMMX), DMF(NMMX), BLO(NQMX,NMMX)
0019    CNOT     +       , DMBLO(NQMX,NMMX)
0020    CNOT     +       , DMRCTH(NQMX,NMMX), DMCM(NQMX,NMMX), WTH(NQMX,NMMX)
0021          COMMON/MLIN/      NM, R(NQMX,NMMX)
0022    CNOT     +       , PHIR(NQMX,NMMX), CUR(NQMX,NMMX), SM(NQMX,NMMX), Z(NQMX,NMMX)
0023          COMMON/QLIN/      RUMRQ(NQMX), NQ
0024    CNOT     +       , Q(NQMX,NMMX), RUMR(NQMX,NMMX)
0025    CNPT     +       , ZQH(NQMX), ZQS(NQMX), NQI, NQB
0026
0027          WRITE(6,*) 'VO, VOC, PHO = ', VO, VOC, PHO
0028          VOSQ5      =VO*VO*VOC*VOC*.5
0029          PHO        =PHO*9.8062
0030          DO        J=1,NM
0031            TEM      =CPT(J)*VOSQ5+PHO
0032            DO      I=1,NQ
0033              H(I,J)=TEM
0034              END DO
0035            DO      I=NQI1,NQIB
0036              H(I,J)=CTH(I,J)*R(I,J)*OMG+TEM
0037              END DO
0038            END DO
0039          RETURN
0040          END
```

PROGRAM SECTIONS

Name	Bytes	Attributes								
0 $CODE	290	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
1 $PDATA	15	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT	LONG
2 $LOCAL	28	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	LONG
3 MFLOW	2516	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
4 MLIN	620	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
5 QLIN	60	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
Total Space Allocated	3529									

ENTRY POINTS

Address	Type	Name	References
0-00000000		ENTHAL	4#

VARIABLES

Address	Type	Name	Attributes	References			
3-0000002C	R*4	G	COMM	15			
**	I*4	I		32=	33	35=	36(3)
**	I*4	J		30=	31	33	36(3)
4-00000000	I*4	NM	COMM	21	30		
5-00000038	I*4	NQ	COMM	23	32		
AP-00000008@	I*4	NQI1		4	35		
AP-0000000C@	I*4	NQIB		4	35		
3-00000030	R*4	OMG	COMM	15	36		
AP-00000010@	R*4	PHO		4	27	29(2)=	31
**	R*4	TEM		31=	33	36	

AP-00000014@	R*4	VO	4	27	28(2)
AP-00000018@	R*4	VOC	4	27	28(2)
**	R*4	VOSQ5	28=	31	

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References		
3-00000034	R*4	CM	COMM	616	(14, 11)	15		
AP-00000004@	R*4	CPT		44	(11)	4	13	31
3-0000029C	R*4	CTH	COMM	616	(14, 11)	15	36	
3-00000000	R*4	GS	COMM	44	(11)	15		
3-0000076C	R*4	H	COMM	616	(14, 11)	15	33=	36=
4-00000004	R*4	R	COMM	616	(14, 11)	21	36	
3-00000504	R*4	RCTH	COMM	616	(14, 11)	15		
5-00000000	R*4	RUMRQ	COMM	56	(14)	23		

PARAMETER CONSTANTS

Type	Name	References			
I*4	NMMX	11#	13	15(5)	21
I*4	NQMX	11#	15(4)	21	23

```
+----------------------------------------------+
|          KEY TO REFERENCE FLAGS               |
|   =   - Value Modified                        |
|   #   - Defining Reference                    |
|   A   - Actual Argument, possibly modified    |
|   D   - Data Initialization                   |
|  (n)  - Number of occurrences on line         |
+----------------------------------------------+
```

COMMAND QUALIFIERS

F/LIS/CHE/CRO ENTHAL

```
/CHECK=(BOUNDS,OVERFLOW,UNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE
```

COMPILATION STATISTICS

Run Time:	2.20 seconds
Elapsed Time:	3.93 seconds
Page Faults:	201
Dynamic Memory:	339 pages

```
0001      C
0002      C+++++++++   SUBROUTINE INTSPL    +++++++++
0003      C
0004                SUBROUTINE INTSPL(X,Y,N,DI,L,Y1,Y2,W,NW,ICON)
0005                REAL X(N),Y(N),DI(N),L(N),Y1(N),Y2(N),W(NW)
0006                CALL SPLINE(X,Y,N,L,Y1,Y2,W,NW,ICON)
0007                DI(1)=0.0
0008                DO 10 I=1,N-1
0009      10        DI(I+1)=(Y(I+1)+Y(I))*L(I)/2.-(Y2(I+1)+Y2(I))*L(I)**3/24.
0010                RETURN
0011                END
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 $CODE	396	PIC CON REL LCL SHR EXE RD NOWRT LONG
2 $LOCAL	324	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
Total Space Allocated	720	

ENTRY POINTS

Address	Type	Name	References
0-00000000		INTSPL	4#

VARIABLES

Address	Type	Name	Attributes	References			
**	I*4	I		8=	9(7)		
AP-00000028@	I*4	ICON		4	6A		
AP-0000000C@	I*4	N		4	5(6)	6A	8
AP-00000024@	I*4	NW		4	5	6A	

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References			
AP-00000010@	R*4	DI		**	(*)	4	5	7=	9=
AP-00000014@	R*4	L		**	(*)	4	5	6A	9(2)
AP-00000020@	R*4	W		**	(*)	4	5	6A	
AP-00000004@	R*4	X		**	(*)	4	5	6A	
AP-00000008@	R*4	Y		**	(*)	4	5	6A	9(2)
AP-00000018@	R*4	Y1		**	(*)	4	5	6A	
AP-0000001C@	R*4	Y2		**	(*)	4	5	6A	9(2)

LABELS

Address	Label	References	
**	10	8	9#

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	SPLINE	6

```
C
C+++++++++     SUBROUTINE MLINE +++++++++
C
      SUBROUTINE MLINE ( ITRX, NQ, NM, RAS, Q, R, Z, SM, PHIR, CUR, XM,
     *                          ICON, X1, X2, Y1, Y2)

C INPUT ITRX, NM, NQ, R(*), RAS, Q(*), & Z(*)
C OUTPUT CUR(*), ICON, PHIR(*), SM(*), X1(*), X2(*), XM(*), Y1(*), &
C        Y2(*)
C TO BE COMPILED BY VAX FORTRAN V4.0-2
C TOUCHED 10JUN86

      PARAMETER ( ND=100, NQMX=14)
      PARAMETER ( ND4M=ND*4-10)

      REAL      X1(NQ), X2(NQ), Y1(NQ), Y2(NQ),
     *  DI(ND), F(ND), L(ND), XI(ND), Y1I(ND), Y2I(ND), YI(ND),
     *  DX1MIN(3), WORK(ND4M),
     *  CUR(NQMX,NM), PHIR(NQMX,NM), Q(NQMX,NM), R(NQMX,NM),
     *  SM(NQMX,NM), XM(NQMX,NM), Z(NQMX,NM)

      DATA      PI/3.14159265359/

      RAD45     =PI*45./180.
        DO 2 K=1,3
        THER    =(K-2)*RAD45
        CT=COS(THER)
        ST=SIN(THER)
        DX1MIN(K)=ABS((Z(2,2)-Z(1,2))*CT+(R(2,2)-R(1,2))*ST)
        DO 2 I=3,NQ
        DX1=ABS((Z(I,2)-Z(I-1,2))*CT+(R(I,2)-R(I-1,2))*ST)
        IF(DX1.LT.DX1MIN(K)) DX1MIN(K)=DX1
2       CONTINUE
        KMAX=1
        DXMAX=DX1MIN(1)
        DO 4 K=2,3
        IF(DX1MIN(K).LT.DXMAX) GO TO 4
        KMAX=K
        DXMAX=DX1MIN(K)
4       CONTINUE
      THER      =(KMAX-2)*RAD45
        CT=COS(THER)
        ST=SIN(THER)
        DO 10 J=2,NM-1
        DO 20 IQ=1,NQ
        X1(IQ)=Z(IQ,J)*CT+R(IQ,J)*ST
        X2(IQ)=-Z(IQ,J)*ST+R(IQ,J)*CT
20      CONTINUE
601     CALL AKIMA0(X1,X2,NQ,Y1,Y2,WORK,NQ+3,ICON)
        LINE=601
        IF(ICON.NE.0) GO TO 9000
        DO 30 I=1,NQ
        PHIR(I,J)=ATAN(Y1(I))+THER
30      CUR(I,J)=-Y2(I)/(1.+Y1(I)**2)**(3./2.)
        SM(1,J)=0.
        XM(1,J)=0.
        DO 40 I=2,NQ
        DX=(X1(I)-X1(I-1))/(ND-1)
        DO 41 II=1,ND
41      XI(II)=X1(I-1)+DX*(II-1)
602     CALL AKIMAI(X1,X2,NQ,XI,ND,YI,Y1I,WORK,2*NQ+3,ICON)
        LINE=602
        IF(ICON.NE.0) GO TO 9000
        DO 43 II=1,ND
43      F(II)=SQRT(1.+Y1I(II)**2)
603     CALL INTSPL(XI,F,ND,DI,L,Y1I,Y2I,WORK,4*ND-10,ICON)
        LINE=603
        IF(ICON.NE.0) GO TO 9000
        SM(I,J)=SM(I-1,J)
        DO 45 II=2,ND
45      SM(I,J)=SM(I,J)+DI(II)
        IF(ITRX.EQ.0) GO TO 40
        DO 46 II=1,ND
        RI=XI(II)*ST+YI(II)*CT
46      F(II)=F(II)*RAS/RI
604     CALL INTSPL(XI,F,ND,DI,L,Y1I,Y2I,WORK,4*ND-10,ICON)
        LINE=604
```

```
0078            IF(ICON.NE.0) GO TO 9000
0079            XM(I,J)=XM(I-1,J)
0080            DO 48 II=2,ND
0081      48    XM(I,J)=XM(I,J)+DI(II)
0082      40    CONTINUE
0083      10    CONTINUE
0084            CALL BCMLIN(NQ,NM,Q,PHIR,CUR)
0085            ICON=0
0086       999  RETURN
0087
0088      9000 PRINT 9010, ICON, LINE
0089      9010 FORMAT('??? ICON ='I7' AT LINE 'I3' IN SUB MLINE ???')
0090           STOP 'MLINE.9000'
0091
0092      611   FORMAT(//5X,I2)
0093      618   FORMAT(8F10.5)
0094            END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1775	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	65	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	5092	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	6932	

ENTRY POINTS

Address	Type	Name	References
0-00000000		MLINE	4#

VARIABLES

Address	Type	Name	Attributes	References							
2-0000111C	R*4	CT		27=	29	31	42=	46	47	74	
**	R*4	DX		58=	60						
**	R*4	DX1		31=	32(2)						
**	R*4	DXMAX		35=	37	39=					
**	I*4	I		30=	31(4)	52=	53(2)	54(3)	57=	58(2)	60
				69(2)	71(2)	79(2)	81(2)				
AP-00000030@	I*4	ICON		4	49A	51	61A	63	66A	68	76A
				78	85=	88					
**	I*4	II		59=	60(2)	64=	65(2)	70=	71	73=	74(2)
				75(2)	80=	81					
**	I*4	IQ		45=	46(3)	47(3)					
AP-00000004@	I*4	ITRX		4	72						
2-00001124	I*4	J		44=	46(2)	47(2)	53	54	55	56	69(2)
				71(2)	79(2)	81(2)					
**	I*4	K		25=	26	29	32(2)	36=	37	38	39
**	I*4	KMAX		34=	38=	41					
2-00001130	I*4	LINE		50=	62=	67=	77=	88			
AP-0000000C@	I*4	NM		4	16(7)	44	84A				
AP-00000008@	I*4	NQ		4	16(4)	30	45	49(2)A	52	57	61(2)A
				84A							
2-00001114	R*4	PI		22D	24						
**	R*4	RAD45		24=	26	41					
AP-00000010@	R*4	RAS		4	75						
**	R*4	RI		74=	75						
2-00001120	R*4	ST		28=	29	31	43=	46	47	74	
2-00001118	R*4	THER		26=	27	28	41=	42	43	53	

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
AP-00000028@	R*4	CUR		**	(14, *)	4	16	54=	84A	
2-00000000	R*4	DI		400	(100)	16	66A	71	76A	81
2-00000AF0	R*4	DX1MIN		12	(3)	16	29=	32(2)=	35	37
						39				
2-00000190	R*4	F		400	(100)	16	65=	66A	75(2)=	76A

```
 2-00000320  R*4  L                 400  (100)     16      66A     76A

AP-00000024@ R*4  PHIR               **  (14, *)    4      16      53=      84A
AP-00000014@ R*4  Q                  **  (14, *)    4      16      84A
AP-00000018@ R*4  R                  **  (14, *)    4      16      29(2)    31(2)    46
                                                   47
AP-00000020@ R*4  SM                 **  (14, *)    4      16      55=      69(2)=   71(2)=
 2-00000AFC  R*4  WORK             1560  (390)     16      49A     61A      66A      76A

AP-00000034@ R*4  X1                 **  (*)        4      16      46=      49A      58(2)
                                                   60      61A
AP-00000038@ R*4  X2                 **  (*)        4      16      47=      49A      61A
 2-000004B0  R*4  XI                400  (100)     16      60=     61A      66A      74
                                                   76A
AP-0000002C@ R*4  XM                 **  (14, *)    4      16      56=      79(2)=   81(2)=
AP-0000003C@ R*4  Y1                 **  (*)        4      16      49A      53       54

 2-00000640  R*4  Y1I               400  (100)     16      61A     65       66A      76A
AP-00000040@ R*4  Y2                 **  (*)        4      16      49A      54
 2-000007D0  R*4  Y2I               400  (100)     16      66A     76A
 2-00000960  R*4  YI                400  (100)     16      61A     74
AP-0000001C@ R*4  Z                  **  (14, *)    4      16      29(2)    31(2)    46
                                                   47
```

PARAMETER CONSTANTS

```
  Type  Name                 References

  I*4   ND                   13#     14      16(7)   58      59      61      64      66(2)
                             70      73      76(2)   80
  I*4   ND4M                 14#     16
  I*4   NQMX                 13#     16(7)
```

LABELS

```
    Address     Label        References

       **       2            25      30      33#
  0-.0000253    4            36      37      40#
       **       10           44      83#
       **       20           45      48#
       **       30           52      54#

  0-00000679    40           57      72      82#
       **       41           59      60#
       **       43           64      65#
       **       45           70      71#
       **       46           73      75#

       **       48           80      81#
       **       601          49#
       **       602          61#
       **       603          66#
       **       604          76#

       **       611'         92#
       **       618'         93#
       **       999          86#
  0-0000068C    9000         51      63      68      78      88#
  1-00000012    9010'        88      89#
```

FUNCTIONS AND SUBROUTINES REFERENCED

```
  Type  Name                 References

        AKIMAI               61
        AKIMAO               49
        BCMLIN               84
        INTSPL               66      76
  R*4   MTH$ATAN             53

  R*4   MTH$COS              27      42
  R*4   MTH$SIN              28      43
  R*4   MTH$SQRT             65
```

```
C
C+++++++++          SUBROUTINE NUF          +++++++++
C
      SUBROUTINE NUF( NCMNU, CMNU, CPTNU, RNU,                        IN
     +                           CPT)                                 OUT

C (3037-NUF) DETERMINE G, GS, R, & CM, FOR THE CASE OF NONUNIFORM
C             INFLOW
C REFERENCED BY NUFLOW
C REFERENCES MTH$COS (THRICE), MTH$SQRT, & SPLIN3 (4 TIMES)
C TO BE COMPILED BY VAX FORTRAN V.4.0-2.
C CODED BY W. CHIANG, 21MAY86, FOR SCMV3.0; REVISED 18JUN86

      PARAMETER ( NCMMX=20, NMMX=11, NQMX=14)

      COMMON/MFLOW/     GS(NMMX), G, OMG,
     +  CM(NQMX,NMMX), CTH(NQMX,NMMX),
     +  RCTH(NQMX,NMMX), H(NQMX,NMMX), A(NQMX,NMMX), DEN(NQMX,NMMX)
CNOT     +       , ENT(NQMX,NMMX), DMF(NMMX), BLO(NQMX,NMMX)
CNOT     +       , DMBLO(NQMX,NMMX)
CNOT     +       , DMRCTH(NQMX,NMMX), DMCM(NQMX,NMMX), WTH(NQMX,NMMX)
      COMMON/MLIN/      NM, R(NQMX,NMMX), PHIR(NQMX,NMMX)
CNOT     +       , CUR(NQMX,NMMX), SM(NQMX,NMMX), Z(NQMX,NMMX)
      COMMON/NU/        NMM1, PI, COSRUM(NQMX)
CNOT     +       , SINRUM(NQMX)
      COMMON/QLIN/      RUMRQ(NQMX), NQ, Q(NQMX,NMMX)
CNOT     +       , RUMR(NQMX,NMMX)
CNOT     +       , ZQH(NQMX), ZQS(NQMX), NQI, NQB

      DIMENSION AR1(NMMX), AREAR(NQMX),
     +  CM1(NMMX), CMNU(NCMNU), COE1(NMMX), CPT(NMMX), CPTNU(NCMNU),
     +  DUM1(1), ENU(NCMMX),
     +  R1(NMMX), RCMNU(NCMMX), RNU(NCMNU), SUMA(NCMMX)

C1900 FIND TOTAL FLOW RATE, ASSUMING STRAIGHT Q-LINE & CONSTANT DENSITY
      RUMRQ1    =RUMRQ(1)
      ENU1      =PHIR(1,1)-RUMRQ1
      RNU1      =RNU(1)
      TEM       =( PHIR(1,1)-PHIR(1,NM) )/( RNU1-RNU(NCMNU) )
      DO        K=1,NCMNU
        RCMNU(K)=CMNU(K) * RNU(K) * COS( ( RNU(K)-RNU1 )*TEM + ENU1 )
        END DO

      CALL    SPLIN3( NCMNU, RNU, RCMNU, 0, DUM1, DUM1, DUM1, SUMA, .5,
     +                EPS)

C1920 FLOW RATE IN EACH STREAM TUBE, ASSUMMING UNIFORM DENSITY
      CORUM1    =COSRUM(1)
      PI2DEN    =DEN(1,1)*(PI+PI)
      G         =SUMA(NCMNU)*PI2DEN/CORUM1
      NMM1      =NM-1
      DELQ      =SUMA(NCMNU)/NMM1
      GS(1)     =0.
      DO        J=2,NMM1
        GS(J)   =GS(J-1)+DELQ
        END DO
      GS(NM)    =SUMA(NCMNU)

C1930 DETERMINE VEL. PROFILE & LOCATION OF STREAMLINES AT UPSTREAM

      CALL    SPLIN3( NCMNU, SUMA,   RNU,   -NM, GS,  R1, AR1, DUM1, .5,
     +                EPS)

      CALL    SPLIN3( NCMNU,  RNU,  CMNU, -NMM1, R1, CM1, AR1, DUM1, .5,
     +                EPS)

      CALL    SPLIN3( NCMNU,  RNU, CPTNU,   -NM, R1, CPT, AR1, DUM1, .5,
     +                EPS)

      CM(1,1)   =CMNU(1)
      CM1(NM)   =CMNU(NCMNU)
      CM(1,NM)  =CMNU(NCMNU)
      TEM       =R(1,1)*R(1,1)
      DO        J=2,NMM1
        GS(J-1) =GS(J)*PI2DEN
        R(1,J)  =R1(J)
        CM(1,J) =CM1(J)
```

```
0078            AR1(J)  =R1(J)*R1(J)-TEM
0079            END DO
0080
0081      C1960 INITIAL STREAMLINE LOCATONS (R & Q) AT 1<I<=NQ, 1<J<NM
0082          AR1NM     =R(1,NM)*R(1,NM)-TEM
0083          DO        I=2,NQ
0084            TEM     =R(I,1)*R(I,1)
0085            TEMP    =( R(I,NM)*R(I,NM)-TEM )/AR1NM
0086            AREAR(I)=TEMP
0087            DO      J=2,NMM1
0088              R(I,J)=SQRT( AR1(J)*TEMP + TEM )
0089              END DO
0090            END DO
0091          DO        I=1,NQ
0092            RI1     =R(I,1)
0093            CORUMI  =COSRUM(I)
0094            Q(I,1)  =0.
0095            DO      J=2,NM
0096              Q(I,J)=( R(I,J)-RI1 )/CORUMI
0097              END DO
0098            PHIRI1  =PHIR(I,1)
0099            TEM     =( PHIR(I,NM)-PHIRI1 )/Q(I,NM)
0100            DO      J=2,NMM1
0101              PHIR(I,J)      =Q(I,J)*TEM+PHIRI1
0102              END DO
0103            END DO
0104
0105      C1970 INITIAL CM AT I>1
0106          DO        J=1,NM
0107            COE1(J) =COS( PHIR(1,J)-RUMRQ1 )
0108            END DO
0109          DO        I=2,NQ
0110            TEM     =COSRUM(I)/( CORUM1*AREAR(I) )
0111            RUMRQI  =RUMRQ(I)
0112            DO      J=1,NM
0113              CM(I,J)        =CM1(J)*TEM*COE1(J)/COS( PHIR(I,J)-RUMRQI )
0114              END DO
0115            END DO
0116
0117          RETURN
0118          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes								
0	$CODE	1552	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
1	$PDATA	8	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT	LONG
2	$LOCAL	808	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	LONG
3	MFLOW	3748	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
4	MLIN	1236	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
5	NU	64	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
6	QLIN	676	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
	Total Space Allocated	8092									

ENTRY POINTS

Address	Type	Name	References
0-00000000		NUF	4#

VARIABLES

Address	Type	Name	Attributes	References							
**	R*4	AR1NM		82=	85						
2-000001E4	R*4	CORUM1		48=	50	110					
**	R*4	CORUMI		93=	96						
**	R*4	DELQ		52=	55						
**	R*4	ENU1		37=	41						
2-000001E0	R*4	EPS		44A	61A	64A	67A				
3-0000002C	R*4	G	COMM	16	50=						
**	I*4	I		83=	84(2)	85(2)	86	88	91=	92	93

```
                                   94      96(2)   98      99(2)   101(2)  109=    110(2)  111
                                   113(2)
       **          I*4  J          54=     55(2)   74=     75(2)   76(2)   77(2)   78(3)   87=
                                   88(2)   95=     96(2)   100=    101(2)  106=    107(2)  112=
                                   113(4)
       **          I*4  K          40=     41(4)

AP-00000004@       I*4  NCMNU            4       30(3)   39      40      44A     50      52      57
                                   61A     64A     67A     71      72
 4-00000000        I*4  NM         COMM    22      39      51      57      61      67      71      72
                                   82(2)   85(2)   95      99(2)   106     112
 5-00000000        I*4  NMM1       COMM    24      51=     52      54      64      74      87      100
 6-00000038        I*4  NQ         COMM    26      83      91      109
 3-00000030        R*4  OMG        COMM    16

       **          R*4  PHIRI1             98=     99      101
 5-00000004        R*4  PI         COMM    24      49(2)
       **          R*4  PI2DEN             49=     50      75
       **          R*4  RI1                92=     96
       **          R*4  RNU1               38=     39      41

 2-000001DC        R*4  RUMRQ1             36=     37      107
       **          R*4  RUMRQI             111=    113
       **          R*4  TEM                39=     41      73=     78      82      84=     85      88
                                           99=     101     110=    113
       **          R*4  TEMP               85=     86      88
```

ARRAYS

```
  Address     Type  Name      Attributes    Bytes  Dimensions    References

 3-000009D4   R*4   A         COMM          616  (14, 11)       16
 2-00000000   R*4   AR1                      44  (11)           30      61A      64A       67A     78=
                                                                88
 2-0000002C   R*4   AREAR                    56  (14)           30      86=      110
 3-00000034   R*4   CM        COMM          616  (14, 11)       16      70=      72=       77=     113=
 2-00000064   R*4   CM1                      44  (11)           30      64A      71=       77      113

AP-00000008@  R*4   CMNU                     **  (*)             4      30       41        64A     70
                                                                71      72
 2-00000090   R*4   COE1                     44  (11)           30      107=     113
 5-00000008   R*4   COSRUM    COMM           56  (14)           24      48       93        110
AP-00000014@  R*4   CPT                      44  (11)            4      30       67A
AP-0000000C@  R*4   CPTNU                    **  (*)             4      30       67A

 3-0000029C   R*4   CTH       COMM          616  (14, 11)       16
 3-00000C3C   R*4   DEN       COMM          616  (14, 11)       16      49
 2-000000BC   R*4   DUM1                      4  (1)            30      44(3)A   61A       64A     67A
 2-000000C0   R*4   ENU                      80  (20)           30
 3-00000000   R*4   GS        COMM           44  (11)           16      53=      55(2)=    57=     61A
                                                                75(2)=

 3-0000076C   R*4   H         COMM          616  (14, 11)       16
 4-0000026C   R*4   PHIR      COMM          616  (14, 11)       22      37       39(2)     98      99
                                                                101=    107      113
 6-0000003C   R*4   Q         COMM          616  (14, 11)       26      94=      96=       99      101
 4-00000004   R*4   R         COMM          616  (14, 11)       22      73(2)    76=       82(2)   84(2)
                                                                85(2)   88=      92        96
 2-00000110   R*4   R1                       44  (11)           30      61A      64A       67A     76
                                                                78(2)

 2-0000013C   R*4   RCMNU                    80  (20)           30      41=      44A
 3-00000504   R*4   RCTH      COMM          616  (14, 11)       16
AP-00000010@  R*4   RNU                      **  (*)             4      30       38        39      41(2)
                                                                44A     61A      64A       67A
 6-00000000   R*4   RUMRQ     COMM           56  (14)           26      36       111
 2-0000018C   R*4   SUMA                     80  (20)           30      44A      50        52      57
                                                                61A
```

PARAMETER CONSTANTS

```
  Type  Name                     References

   I*4  NCMMX                      14#     30(3)
   I*4  NMMX                       14#     16(7)    22(2)    26      30(5)
   I*4  NQMX                       14#     16(6)    22(2)    24      26(2)    30
```


FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References			
R*4	MTH$COS	41	107	113	
R*4	MTH$SQRT	88			
	SPLIN3	44	61	64	67

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|   =    - Value Modified                       |
|   #    - Defining Reference                   |
|   A    - Actual Argument, possibly modified   |
|   D    - Data Initialization                  |
|  (n)   - Number of occurrences on line        |
+-----------------------------------------------+
```

COMMAND QUALIFIERS

```
F/LIS/CHE/CRO NUF

/CHECK=(BOUNDS,OVERFLOW,UNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE
```

COMPILATION STATISTICS

```
Run Time:          6.58 seconds
Elapsed Time:      8.19 seconds
Page Faults:       298
Dynamic Memory:    425 pages
```

```
C
C+++++++++          SUBROUTINE NUFLOW          +++++++++
C
      SUBROUTINE NUFLOW( IBFLOW, IDSN3, IN, LG)

C (3037-NUFLOW) I.O. SUBROUTINE FOR NONUNIFORM (OR UNIFORM) WATER FLOW
C "CNOTUSED" INDICATES STATEMENTS OF NO USE IN THE SCM PROGRAM;
C "CGAS" INDICATES STATEMENTS TO BE USED ONLY IF THE FLUID IS A GAS;
C "CNOT" INDICATES STATEMENTS NOT NECESSARY IN THE PRESENT APPLICATION,
C        PART OF WHICH COULD BE REQUIRED FOR GAS FLOW.
C INPUT IBFLOW, IN, LG, & OTHERS; OUTPUT OTHERS.
C REFERENCED BY MAIN (SCM)
C REFERENCES ENTHAL (TWICE), FOR$CLOSE, FOR$OPEN, MTH$COS (TWICE),
C TO BE COMPILED BY VAX FORTRAN V.4.0-2.
C ADOPTED FROM BBFLOW & DATOUT, 07MAY86, FOR V2.7;
C        REVISED 01AUG86, FOR V3.0; CHIANG; REVISED 20OCT87

C IBFLOW=-2 IF G, CM(*), R(*) & Q(*) ARE INPUT DATA;
C        =-1 FOR NONUNIFORM INFLOWS;
C        = 0 FOR UNIFORM INFLOW;
C        = 1 IF DATA ARE AVAILABLE FROM BLADE TO BLADE CALCULATIONS;
C        =10 TO USE SUB BBFLOW TO PREPARE INPUT DATA.
C IDSN3 = 0 FOR NO ACTION;
C        = 1 TO STORE A PART OF RESULTS TO DSN3ZI.DAT TO BE USED BY
C          (3037-DSN3)
C IN     LOGICAL UNIT # FOR INPUT FILE
C LG     = 0 FOR SEA WATER (DEN=1025 KG/CU.M.);
C        = 1 FOR PURE WATER (DEN=1000);
C        = 2 FOR GAS.

C READ1 (A DUMMY LINE TO STORE TITLE, NOTE...)
C READ2 NM       # OF MERIDIONAL STREAM LINES, >=5
C       NQ       # OF Q-LINES, >=5
C       NQI      LINE # OF Q-LINE AT THE LINE PRIOR TO THE LEADING EDGE
C                OF BLADE
C       NQB      # OF Q-LINES ON THE BLADE
C       ICM      = 1 TO SOLVE D(CM)/DQ (INOUE); =2 TO SOLVE D(CM*CM)/DQ
C READ3 OMG      ANGULAR VEL. in rad/sec
C       RAS      REFERENCE RADIUS in m
C       DMPCM    DAMPING FACTOR USED IN CM ITERATION, INOUE 1.
C       DMPG     DAMPING FACTOR USED IN G ITERATION, INOUE .5
C       DMPL     DAMPING FACTOR USED IN Q ITERATION, INOUE .1
C       EPSCM    CONVERGING CRITERION FOR CM ITERATION, INOUE 1.E-6
C       EPSG     CONVERGING CRITERION FOR G ITERATION, INOUE 1.E-4
C       EPSL     CONVERGING CRITERION FOR Q ITERATION, INOUE 1.E-4

C READ4 RUMRQ    ANGLE BETWEEN Q-LINE(STRAIGHT) & RADIUS DIR., POSITIVE
C                C.W., in rad.
C READ5 Z        Z-COORINATE in m
C READ6 PHID1    ANGLE BETWEEN HUB & Z-ZXIS in deg
C       PHIDN    ANGLE BETWEEN CASING & Z-ZXIS in deg
C READ7 SM       M-COORDINATE in m
C READ8 CTH      PERIPHERAL VEL. in m/sec
C READ9 BLO      BLOCKAGE FACTOR, Kb=(theta2-theta1)/(2*pi/N), 0 TO 1.,
C                1 IF NO BLOCKAGE

C ***** READ10 & READ11 ARE REQUIRED ONLY IF LG=2 *****
C READ10DEN      DENSITY in kg/cu.m.
C READ11ENT      CHANGE OF ENTROPY in J/kg

C ***** READ12 TO READ15 ARE REQUIRED ONLY IF IBFLOW=-2 *****
C READ12GS       CUMULATIVE MASS FLOW RATE IN FLOW-TUBES, GS(1) TO
C                DESIGNATES THAT BN. J=1 & J=2, GS(NM-1)=G, in kg/sec
C       G        MASS FLOW RATE in kg/sec
C READ13CM       MERIDIONAL VEL. in m/sec
C READ14R        RADIAL DISTANCE FROM AXIS OF RATATION in m
C READ15H        ENTHALPY in J/kg or (m/sec)**2

C ***** READ16 TO READ21 ARE REQUIRED ONLY IF IBFLOW=-1 *****
C READ16R        RADIAL DISTANCE FROM AXIS OF RATATION in m
C READ17NCMNU    # OF INPUT DATA FOR INFLOW VEL. PROFILE
C READ18RNUC1    1ST CONVERSION FACTOR TO BE MULTIPLIED TO THE INPUT RNU
C       RNUC2    2ND CONVERSION FACTOR TO BE MULTIPLIED TO THE INPUT RNU
C       CMNUC1   1ST CONVERSION FACTOR TO BE MULTIPLIED TO THE INPUT CMNU
C       CMNUC2   2ND CONVERSION FACTOR TO BE MULTIPLIED TO THE INPUT CMNU
C       RERR     ALLOWABLE ERROR IN RNU DATA IN m
C       EPS      CONVERGENCE CRITERION TO BE USED IN SPLINE ROUTINE
```

```
C       VO      FLOW VEL. (OR VEHICLE VEL.) AT FREE STREAM, in m/s AFTER
C               BEEN MULTIPLIED BY VOC
C       VOC     CONVERSION FAC. TO BE MULTI. TO VO
C       PHO     STATIC PRESSURE HEAD AT FREE STREAM in m
C READ19RNU     RADII OF NCMNU PTS., in m AFTER MULTIPLYED W/ RNUC1*RNUC2
C READ20CMNU    MERIDIONAL VEL. AS FUNCTION OF RNU, in m/sec AFTER BEEN
C               MULTIPLIED BY CMNUC1*CMNUC2
C READ21CPTNU   PRESSURE COEFFICIENT AS FUNCTION OF RNU

C ***** READ22 & READ23 ARE REQUIRED ONLY IF IBFLOW=0 *****
C READ22G       MASS FLOW RATE in kg/sec
C       CPTC    CONSTAT PRESSURE COEFFICIENT
C       VO    . FLOW VEL. (OR VEHICLE VEL.) AT FREE STREAM, in m/s AFTER
C               BEEN MULTIPLIED BY VOC
C       VOC     CONVERSION FAC. TO BE MULTI. TO VO
C       PHO     STATIC PRESSURE HEAD AT FREE STREAM in m
C READ23R       RADIAL DISTANCE FROM AXIS OF RATATION in m

C ***** READ24 TO READ29 ARE REQUIRED ONLY IF IBFLOW=1 *****
C READ24G       MASS FLOW RATE in kg/sec
C READ25CM      MERIDIONAL VEL. in m/sec
C READ26R       RADIAL DISTANCE FROM AXIS OF RATATION in m
C READ27H       ENTHALPY in J/kg or (m/sec)**2
C READ28TNA     TN OF EQ. 10
C READ29CTHD    A KIND OF [DELTA(CM*CTH)]/CM, FROM BLADE TO BLADE PROGRAM

      PARAMETER ( NCMMX=20, NMMX=11, NQMX=14)

CNOT      COMMON/DESIGN/       BSBE, BE, RT, RPM, UT, EFFI, AVIL, VELR, HEADAD,
CNOT     +  FLOWC, FLOWR, CONST, POW
CGAS      COMMON/GASCON/       SHR, GASC, CP, POSABS, TOSABS
CNOTUSED      COMMON/LOAD/     PCON(NMMX), IPCON(NMMX)
      COMMON/MFLOW/     GS(NMMX), G, OMG,
     +  CM(NQMX,NMMX), CTH(NQMX,NMMX),
     +  RCTH(NQMX,NMMX), H(NQMX,NMMX), A(NQMX,NMMX), DEN(NQMX,NMMX),
     +  ENT(NQMX,NMMX), DMF(NMMX), BLO(NQMX,NMMX)
CNOT     +       , DMBLO(NQMX,NMMX),
CNOT     +      DMRCTH(NQMX,NMMX), DMCM(NQMX,NMMX), WTH(NQMX,NMMX)
C       A(*), DMF(*), & DMRCTH(*) ARE NOT USED IN "SCM"
      COMMON/MLIN/      NM, R(NQMX,NMMX), PHIR(NQMX,NMMX),
     +  CUR(NQMX,NMMX), SM(NQMX,NMMX), Z(NQMX,NMMX)
      COMMON/NONAXI/    TNA(NQMX,NMMX), WCTHD(NQMX,NMMX)
      COMMON/QLIN/      RUMRQ(NQMX), NQ, Q(NQMX,NMMX),-RUMR(NQMX,NMMX),
     +  ZQH(NQMX), ZQS(NQMX), NQI, NQB
C       RUMR(*,*) IS REPLACED BY RUMRQ(*) SINCE Q-LINE IS TAKEN TO BE STRAIGHT
C       ZQS(*) IS NOT USED IN "SCM"; ZQH(*) IS USED IN BBFLOW & DATOUT ONLY
CNOTUSED      COMMON/ROTO/     ZLE(2), ZTE(2), NB, IPRF(NMMX), TMPC(NMMX),
CNOTUSED     +  AXIM(NMMX), ZHAXI, RUMAXI, IAXI, TTE
CNOTUSED      COMMON/SCROLL/   ALPSC, DHU, BS
CNOT      COMMON/STAN/  STANP, STANT, STANDE
      COMMON/XY/        RAS
CNOT     +          , XM(NQMX,NMMX)
      COMMON/NU/        NMM1, PI, COSRUM(NQMX), SINRUM(NQMX)
      COMMON/IQ/        ICM
      COMMON/IQS/       DMPCM, DMPG, DMPL, EPSCM, EPSG, EPSL

      DIMENSION CMNU(NCMMX), CPT(NMMX), CPTNU(NCMMX),
     +  PHID1(NQMX), PHIDN(NQMX), RNU(NCMMX)

      DATA      PI/3.14159265359/

      DEGRAD    =180./PI
      RADDEG    =PI/180.

      IF (IBFLOW.LT.-2 .OR. IBFLOW.GT.1 .OR. LG.LT.0 .OR. LG.GT.1) THEN
        PRINT *, 'CHECK IBFLOW & LG... ', IBFLOW, LG
        STOP 'NUFLOW.50'
        END IF

C 100 *****
      READ (IN,*)                                                  READ1
      READ (IN,*) NM, NQ, NQI, NQB, ICM                            READ2

      WRITE(6,*) 'ICM =', ICM
      WRITE(6,110)
  110 FORMAT(///'0'4X81('+')/5X'+'79X'+'/5X'+'12X
     + 'THE QUASI THREE-DIMENSIONAL DESIGN OF THE DIAGONAL FLOW'12X'+'/
     + 5X,'+'79X'+'/5X81('+'))

CNOTUSED        READ (IN,*) NB
```

```
CGAS            READ (IN,*) SHR, GASC, CP, POSABS, TOSABS
CNOTUSED        READ (IN,*) POW, RPM, CONST, AVIL
      READ (IN,*) OMG, RAS, DMPCM, DMPG, DMPL, EPSCM, EPSG, EPSL          READ3
CNOTUSED        READ (IN,*) STANP
CGAS            READ (IN,*) STANT
CNOTUSED        READ (IN,*) STANDE, ALPSC, DHU, BS
CNOTUSED        READ (IN,*) ZHAXI, RUMAXI, TTE, BSBE, BE, RT
CNOT            READ (IN,*) UT, EFFI
CNOTUSED        READ (IN,*) VELR
CGAS            READ (IN,*) HEADAD
CNOTUSED        READ (IN,*) FLOWC, FLOWR

CNOTUSED        READ (IN,*) ( DMF(J), TMPC(J), AXIM(J), PCON(J), J=1,NM)

      READ (IN,*) ( RUMRQ(I), I=1,NQ)                                    READ4
      READ (IN,*) (   Z(I,1), I=1,NQ)                                    READ5
CNOTUSED        READ (IN,*) ( ZQS(I), I=1,NQ)

      READ (IN,*) ( PHID1(I), PHIDN(I), I=1,NQ)                          READ6
      READ (IN,*) (  SM(I,1), SM(I,NM), I=1,NQ)                          READ7

      READ (IN,*) ( (   CTH(I,J), J=1,NM), I=1,NQ)                       READ8
      READ (IN,*) ( (   BLO(I,J), J=1,NM), I=1,NQ)                       READ9

C 120 INPUT FOR GAS FLOWS *****
      IF (LG.EQ.2) THEN
        READ (IN,*) ( ( DEN(I,J), J=1,NM), I=1,NQ)                       READ10
        READ (IN,*) ( ( ENT(I,J), J=1,NM), I=1,NQ)                       READ11
CNOT            DENREF  =PREF/(GASC*TREF)
        DENREF  =DEN(1,1)
       ELSE
        IF (LG.EQ.0) THEN
          DENREF=1025.
         ELSE
          DENREF=1000.
         END IF
        DO     J=1,NM
          DO   I=1,NQ
            DEN(I,J)    =DENREF
            ENT(I,J)    =0.
            END DO
          END DO
       END IF

C 130 *****
      NMM1      =NM-1
      NQI1      =NQI+1
      NQIB      =NQI+NQB
      DO        I=1,NQ
        COSRUM(I)       =COS( RUMRQ(I) )
        PHIR(I,1)       =PHID1(I)*RADDEG
        PHIR(I,NM)      =PHIDN(I)*RADDEG
        END DO

C 170 *****
      GOTO ( 180, 190, 200, 210), IBFLOW+3

C 180 IBFLOW=-2, INPUT KNOWN G, GS, CM, & R *****
  180 READ (IN,*) ( GS(J), J=1,NM-2), G                                  READ12
      READ (IN,*) ( ( CM(I,J), J=1,NM), I=1,NQ)                          READ13
      READ (IN,*) ( (  R(I,J), J=1,NM), I=1,NQ)                          READ14
      READ (IN,*) ( (  H(I,J), J=1,NM), I=1,NQ)                          READ15
      GOTO 250

C 190 IBFLOW=-1, NONUNIFORM FLOW *****
  190 READ (IN,*) ( R(I,1), R(I,NM), I=1,NQ)                             READ16
      READ (IN,*) NCMNU                                                  READ17
      IF (NCMNU.GT.NCMMX) THEN
        PRINT *, 'NCMNU.GT.NCMMX...', NCMNU, NCMMX
        STOP 'NUFLOW.191'
        END IF
      WRITE(6,192) NCMNU
  192 FORMAT('0INPUT NONUNIFORM VEL. DISTRIBUTION',
     + '& DETERMINE STREAMLINE LOCATIONS AT UPSTREAM'/'0NCMNU ='I3/)
      READ (IN,*)  RNUC1, RNUC2, CMNUC1, CMNUC2, RERR, EPS,
     +  V0, VOC, PHO                                                     READ18
      WRITE(6,*)  'RNUC1, RNUC2, CMNUC1, CMNUC2, RERR, EPS = ',
     +               RNUC1, RNUC2, CMNUC1, CMNUC2, RERR, EPS
      IF (EPS.GE.1.) THEN
        PRINT *, 'EPS.GE.1...', EPS
```

```
          STOP 'NUFLOW.194'
          END IF
        READ (IN,*) (   RNU(K), K=1,NCMNU)                                  READ19
        WRITE(6,*)   '  RNU(m) =', (   RNU(K), K=1,NCMNU)
        READ (IN,*) (  CMNU(K), K=1,NCMNU)                                  READ20
        WRITE(6,*)   ' CMNU(m) =', (  CMNU(K), K=1,NCMNU)
        READ (IN,*) ( CPTNU(K), K=1,NCMNU)                                  READ21
        WRITE(6,*)   'CPTNU(m) =', ( CPTNU(K), K=1,NCMNU)

        RNUC1    = RNUC1* RNUC2
        CMNUC1   =CMNUC1*CMNUC2
        DO      K=1,NCMNU
          CMNU(K) =CMNU(K)*CMNUC1
          RNU(K)  = RNU(K)* RNUC1
          END DO
        IF ( ABS( R(1,1)-RNU(1) ) .GT. RERR  .OR.
     +     ABS( R(1,NM)-RNU(NCMNU) ) .GT. RERR ) THEN
          PRINT *, 'CHECK R(1,1), R(1,NM), RNU(1), RNU(NCMNU)...',
     +                    R(1,1), R(1,NM), RNU(1), RNU(NCMNU), NM, NCMNU,
     +                    RERR
          STOP 'NUFLOW.198'
          END IF
        RNU(1)    =R(1,1)
        RNU(NCMNU)=R(1,NM)

        CALL    NUF( NCMNU, CMNU, CPTNU, RNU, CPT)

        CALL    ENTHAL( CPT, NQI1, NQIB, PHO, VO, VOC)

C 199 CREATE A REVISED INPUT FILE
        OPEN( 8, FILE='SCMJ', STATUS='NEW')
        WRITE(8,*) -2, IDSN3, 1, ' (0) IBFLOW, IDSN3, LG'
        WRITE(8,*) '(3037-SCMI.DAT) FOLLOWING ARE INPUT DATA FOR SUB ',
     +  'NUFLOW... (1)'
        WRITE(8,*) NM, NQ, NQI, NQB, ICM, ' (2) NM, NQ, NQI, NQB, ICM'
        WRITE(8,*) OMG, RAS, DMPCM, DMPG, DMPL, EPSCM, EPSG, EPSL,
     +  ' (3) OMG, RAS, DMPCM, DMPG, DMPL, EPSCM, EPSG, EPSL'
        WRITE(8,*) (  RUMRQ(I), I=1,NQ), ' (4) RUMRQ(I)'
        WRITE(8,*) (     Z(I,1), I=1,NQ), ' (5) Z(I,1)'
        WRITE(8,*) ( PHID1(I), PHIDN(I), I=1,NQ),
     +       ' (6)  PHID1(I), PHIDN(I)'
        WRITE(8,*) (  SM(I,1), SM(I,NM), I=1,NQ),
     +       ' (7)   SM(I,1), SM(I,NM)'
        WRITE(8,*) ( (   CTH(I,J), J=1,NM), I=1,NQ),  ' (8) CTH(I,J)'
        WRITE(8,*) ( (   BLO(I,J), J=1,NM), I=1,NQ),  ' (9) BLO(I,J)'
        WRITE(8,*) ( GS(I), I=1,NM-2), G, ' (12) GS(I)'
        WRITE(8,*) ( (     CM(I,J), J=1,NM), I=1,NQ), ' (13)  CM(I,J)'
        WRITE(8,*) ( (      R(I,J), J=1,NM), I=1,NQ), ' (14)   R(I,J)'
        WRITE(8,*) ( (      H(I,J), J=1,NM), I=1,NQ), ' (15)   H(I,J)'
        WRITE(8,*) 'END OF INPUT DAT SCMI.DAT'
C       ASSUMING LG=1 IF IBFLOW=1
        CLOSE(8)
        PRINT *, 'A NEW FILE SCMJ.DAT CREATED...'
        GOTO 250

C 200 IBFLOW=0, UNIFORM INFLOW *****
  200 READ (IN,*) G, CPTC, VO, VOC, PHO                                     READ22
        WRITE(6,*) 'CONSTANT PRESSURE COEFFICIENT =', CPTC
        READ (IN,*) ( R(I,1), R(I,NM), I=1,NQ)                              READ23

C 205 INITIAL CM, R, & Q, FOR UNIFORM INFLOW & EQUAL CROSS-SEC. FLOW-TUBES
        TEM       =PI*DENREF*.5
        DO        I=1,NQ
          RR1     =R(I,1)*R(I,1)
          AR      =R(I,NM)*R(I,NM)-RR1
          DAR     =AR/NMM1
CNONO     IF (LG.EQ.2)  DENREF=DENSI( H(I,1), ENT(I,1), CM(I,1), CTH(I,1))
CNONO     CMI     =G/( AREAN(I) * (DENREF+DENREF) )
          CMI     =G*COSRUM(I)
     +            /( COS( ( PHIR(I,1)+PHIR(I,NM) )*.5-RUMRQ(I) )
     +              *( BLO(I,1)+BLO(I,NM) )*TEM*AR )
          CM(I,1) =CMI
          CM(I,NM)=CMI
          DO      J=2,NMM1
            CM(I,J)       =CMI
            R(I,J)=SQRT( DAR*(J-1)+RR1 )
            END DO
          END DO
        DO        J=1,NM
          CPT(J)  =CPTC
```

```
          END DO

        CALL    ENTHAL( CPT, NQI1, NQIB, PHO, V0, VOC)

        GOTO 250

C 210 IBFLOW=1, INPUT FROM BLADE-TO-BLADE COMPUTATIONS *****
  210 READ (IN,*) G                                                  READ24
      READ (IN,*) ( (    CM(I,J), J=1,NM), I=1,NQ)                   READ25
      READ (IN,*) ( (     R(I,J), J=1,NM), I=1,NQ)                   READ26
      READ (IN,*) ( (     H(I,J), J=1,NM), I=1,NQ)                   READ27
      READ (IN,*) ( (   TNA(I,J), J=1,NM), I=1,NQ)                   READ28
      READ (IN,*) ( ( WCTHD(I,J), J=1,NM), I=1,NQ)                   READ29
      GOTO 300

C 250 *****
  250 DO       J=1,NM
        DO     I=1,NQ
          TNA(I,J)      =0.
          WCTHD(I,J)    =0.
          END DO
        END DO

C 300 GS *****
  300 IF (IBFLOW.GE.0) THEN
        TEM     =G/NMM1
        DO     J=1,NMM1-1
          GS(J) =TEM*J
          END DO
        END IF

C 700 **********
      WRITE(6,710) MAX(IBFLOW,0)+1, G*60., OMG, RAS
  710 FORMAT('0    '81('+')/'0'10XI2
     +  '-TH THROUGH FLOW CALCULATION BY USE OF THE EXTENDED',
     +  ' HIRSCH''S METHOD'//10X'* TOTAL DESIGN SPECIFICATION :'/
     +  26X'MASS FLOW RATE                   ='E12.5' (KG/MIN)'/
     +  26X'ANGULAR VELOCITY                 ='E12.5' (RAD/SEC)'/
     +  26X'REFERENCE RADIUS                 ='E12.5' (M)')
CNOTUSED        POS     =(POSABS-STANP)/9.8
CNOTUSED        ETS     =( EFFI * HEADAD * UT**2 )/( CP * (TOSABS-STANT) )
CNOT    WRITE(6,720) RT, RPM, VELR, EFFI, HEADAD, FLOWC, ETS, SHR, GASC,
CNOT      + CP, POS, TOSABS, STANP, STANT
CNOT   720 FORMAT(
CNOT      +     26X'BLADE TIP RADIUS                 ='E12.5,' (M)'/
CNOT      +     26X'ROTATION FREQUENCY               ='E12.5,' (R.P.M.)'/
CNOT      +     26X'VELOCITY RATIO                   ='E12.5,/
CNOT      +     26X'TOTAL TO TOTAL EFFICIENCY        ='E12.5,/
CNOT      +     26X'ISENTROPIC HEAD COEFFICIENT      ='E12.5,/
CNOT      +     26X'FLOW COEFFICIENT                 ='E12.5,/
CNOT      +     26X'TOTAL TO STATIC EFFICIENCY       ='E12.5,/
CNOT      +     '0'10X'* WORKING GAS: '/
CNOT      +     26X'SPECIFIC HEAT RATIO              ='E12.5,/
CNOT      +     26X'GAS CONSTANT                     ='E12.5,' (J/KG.K)'/
CNOT      +     26X'SPEC. HEAT AT CONST. PRESSURE    ='E12.5,' (J/KG.K)'/
CNOT      +     '0'10X'* PLENUM STATE :'/
CNOT      +     26X'GAUGE PRESSURE                   ='E12.5,' (MM.AQ)'/
CNOT      +     26X'ABSOLUTE TEMPERATURE             ='E12.5,' (K)'/
CNOT      +     '0'10X'* ATMOSPHERIC STATE :'/
CNOT      +     26X'ABSOLUTE PRESSURE                ='E12.5,' (N/M**2)'/
CNOT      +     26X'ABSOLUTE TEMPERATURE             ='E12.5,' (K)')
CNOTUSED        WRITE(6,730) ALPSC, BSBE, BE, DHU, CONST, AVIL
CNOTUSED  730 FORMAT('0'
CNOTUSED      + 10X'* SCROLL NOZZLE GEOMETRIES AND PERFORMANCE :'/
CNOTUSED      + 26X'SCROLL ANGLE                     ='E12.5,' (DEG)'/
CNOTUSED      + 26X'PASSAGE HEIGHT RATIO             ='E12.5,/
CNOTUSED      + 26X'NOZZLE PASSAGE HEIGHT            ='E12.5,' (M)'/
CNOTUSED      + 26X'HUB DIAMETER                     ='E12.5,' (M)'/
CNOTUSED      + 26X'FREE VORTEX CONSTANT             ='E12.5,' (M**2/SEC)'/
CNOTUSED      + 26X'AVAILABILITY OF FREE VORTEX      ='E12.5)
      WRITE(6,740) NQ, NM
  740 FORMAT(/5X81('+')/
     +  '1'10X'+++++ THE WORKING STATIONS (Q-LINES) +++++'///
     +  15X'THE NUMBER OF Q-LINES'16X'='I2/
     +  15X'THE NUMBER OF MERIDIONAL STREAMLINES ='I2///15X'I'5X
     +  'ZQH(M)'7X'RUM(DEG)'7X'QMAX'/)
CNOT      +     'ZQH(M)'7X'ZQS(M)'6X'RUM(DEG)'7X'QMAX'/)

C 820 INITIAL Q-COORDINATES
      IF (IBFLOW.NE.-1) THEN
```

```
         DO     I=1,NQ
           RI1   =R(I,1)
           CORUMI=COSRUM(I)
           Q(I,1)=0.
           DO   J=2,NM
             Q(I,J)=( R(I,J)-RI1 )/CORUMI
             END DO
           END DO
         END IF

C 900 *****
       DO  950  I=1,NQ
CNOTUSED         RUM       =RUMRQ(I)*DEGRAD
         WRITE(6,5905) I, Z(I,1), RUMRQ(I)*DEGRAD, Q(I,NM)
         IF (I.EQ.NQI1) THEN
           WRITE(6,910)
  910      FORMAT('+'55X'(LEADING EDGE)')
           GOTO 950
           END IF
         IF (I.EQ.NQIB) WRITE(6,920)
  920    FORMAT('+'55X'(TRAILING EDGE)')
CNOTUSED         IF (I.NE.NQI1 .AND. I.NE.NQIB) WRITE(6,5905) I, Z(I,1)
CNOTUSED     +   ,ZQS(I),RUM,Q(I,NM)
CNOTUSED         IF (I.EQ.NQI1) WRITE(6,930) I, Z(I,1), ZQS(I), RUM, Q(I,NM)
CNOTUSED  930    FORMAT(15XI2,4(1XE12.5)' (LEADING EDGE)')
CNOTUSED         IF (I.EQ.NQIB) WRITE(6,940) I, Z(I,1), ZQS(I),RUM,Q(I,NM)
CNOTUSED  940    FORMAT(15XI2,4(1XE12.5)' (TRAILING EDGE)')
  950    CONTINUE

       DO        J=1,NM
         DO      I=1,NQ
           RCTH(I,J)      =R(I,J)*CTH(I,J)
           END DO
         END DO

       WRITE(6,960)
  960 FORMAT(/////11X
     +  '+++++ DESIGN CONDITIONS AT THE WORKING STATIONS +++++'///
     +  15X'J     RCTH'12X'H'10X'ENT'10X'BLO'10X'TNA'8X'WCTHD'8X
     +  'D(BLO)/DM    PHI')
       DO        I=1,NQ
         WRITE(6,5965) I
         DO      J=1,NM
           WRITE(6,5905) J, RCTH(I,J), H(I,J), ENT(I,J), BLO(I,J),
     +                   TNA(I,J), WCTHD(I,J)
CNOTUSED     +             ,DMBLO(I,J)
           END DO
         END DO
       WRITE(6,970)
  970 FORMAT(/////11X
     +  '+++++ INITIAL ASSUMPTION OF MERIDIONAL STREAMLINES +++++'/17X
     +  '(THIS ASSUMPTION IS A PREVIOUS THROUGH FLOW SOLUTION IF THIS ',
     +  'THROUGH FLOW CALCULATION IS NOT THE FIRST ONE.)')
CNOTUSED      DO       J=1,NM
CNOTUSED         WRITE(6,980) J, DMF(J)
CNOTUSED  980    FORMAT(11X'I ='I3'    DMF ='E12.5)
CNOTUSED         END DO

       DO        I=1,NQ
         TEMP    =Z(I,1)
         TEM     =SIN( RUMRQ(I) )
         SINRUM(I)       =TEM
         DO      J=2,NM
CNOT IF Q-LINE IS STRAIGHT        RUMR(I,J)      =RUMRQ(I)
           Z(I,J)=TEMP - Q(I,J)*TEM
           END DO
         END DO
       NQM1      =NQ-1
       DO        I=2,NQM1
         CUR(I,1)          =( ( R(I-1,1)-R(I,1) )/( Z(I-1,1)-Z(I,1) )
     +                       - ( R(I,1)-R(I+1,1) )/( Z(I,1)-Z(I+1,1) ) )*2.
     +                      /( Z(I-1,1)-Z(I+1,1) )
         CUR(I,NM)         =( ( R(I-1,NM)-R(I,NM) )/( Z(I-1,NM)-Z(I,NM) )
     +                       - ( R(I,NM)-R(I+1,NM) )/( Z(I,NM)-Z(I+1,NM) ) )
     +                      * 2. / ( Z(I-1,NM)-Z(I+1,NM) )
         END DO
         CUR(1,1)          =CUR(2,1)    *.5
         CUR(NQ,1)         =CUR(NQM1,1) *.5
         CUR(1,NM)         =CUR(2,NM)   *.5
         CUR(NQ,NM)        =CUR(NQM1,NM)*.5
```

```
      WRITE(6,990)
  990 FORMAT(///15X'J'6X'PHI'10X'CUR'12X'M')
      DO      I=1,NQ
        WRITE(6,5965) I
        WRITE(6,5905)  1, PHID1(I),  CUR(I,1),  SM(I,1)
        WRITE(6,5905) NM, PHIDN(I), CUR(I,NM), SM(I,NM)
        END DO
      WRITE(6,1000)
CNOT 1000 FORMAT(///15X'J'8X'Q'12X'R'10X'PHI'10X'CUR'11X'CM'10X'DEN'12X'M'/)
 1000 FORMAT(///15X'J'8X'Q'12X'R'11X'CM'10X'DEN'10X'CTH')
      DO      I=1,NQ
        WRITE(6,5965) I
        DO     J=1,NM
          WRITE(6,5905) J, Q(I,J), R(I,J), CM(I,J), DEN(I,J), CTH(I,J)
          END DO
        END DO
CNOTUSED      WRITE(6,1010) NB
CNOTUSED 1010 FORMAT(/////11X'+++++ THE CASCADE GEOMETRIES ( NB='I2' ) +++++'///
CNOTUSED     +  15X'J   IPRF     TMPC'/)
CNOTUSED      WRITE(6,1020) ( J, IPRF(J), TMPC(J), J=1,NM)
CNOTUSED 1020 FORMAT(14XI2,4XI2,2XE12.5)

      RETURN

 5905 FORMAT(I16,8E13.5)
 5965 FORMAT('0'10X'I='I2)
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes								
0	$CODE	7119	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
1	$PDATA	1754	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT	LONG
2	$LOCAL	832	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	LONG
3	MFLOW	5024	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
4	MLIN	3084	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
5	NONAXI	1232	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
6	QLIN	1412	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
7	XY	4	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
8	NU	120	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
9	IQ	4	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
10	IQS	24	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
	Total Space Allocated	20609									

ENTRY POINTS

Address	Type	Name	References
0-00000000		NUFLOW	4#

VARIABLES

Address	Type	Name	Attributes	References							
2-000001DC	R*4	AR		302=	303	306					
**	R*4	CMI		306=	309	310	312				
2-000001B4	R*4	CMNUC1		232=	234	248(2)=	250				
2-000001B8	R*4	CMNUC2		232=	234	248					
**	R*4	CORUMI		400=	403						
2-000001D0	R*4	CPTC		294=	295	317					
2-000001E0	R*4	DAR		303=	313						
2-0000018C	R*4	DEGRAD		139=	411						
2-0000019C	R*4	DENREF		187=	190=	192=	196	299			
10-00000000	R*4	DMPCM	COMM	132	160=	273					
10-00000004	R*4	DMPG	COMM	132	160=	273					
10-00000008	R*4	DMPL	COMM	132	160=	273					
2-000001C0	R*4	EPS		232=	234	236	237				
10-0000000C	R*4	EPSCM	COMM	132	160=	273					
10-00000010	R*4	EPSG	COMM	132	160=	273					
10-00000014	R*4	EPSL	COMM	132	160=	273					
3-0000002C	R*4	G	COMM	110	216=	283	294=	306	325=	343	350
2-00000194	I*4	I		172(2)=	173(2)=	176(3)=	177(3)=	179(2)=	180(2)=	184(2)=	185(2)=

				195=	196	197	206=	207(2)	208(2)	209(2)	217(2)=
				218(2)=	219(2)=	223(3)=	275(2)=	276(2)=	277(3)=	279(3)=	281(2)=
				282(2)=	283(2)=	284(2)=	285(2)=	286(2)=	296(3)=	300=	301(2)
				302(2)	306(6)	309	310	312	313	326(2)=	327(2)=
				328(2)=	329(2)=	330(2)=	335=	336	337	398=	399
				400	401	403(2)	409=	411(4)	412	417	428=
				429(3)	438=	439	441(6)	456=	457	458	459
				462(2)	466=	467(11)	470(11)	481=	482	483(3)	484(3)
				489=	490	492(5)					
AP-00000004@	I*4	IBFLOW		4	142(2)	143	213	342	350	397	
9-00000000	I*4	ICM	COMM	131	149=	151	272				
AP-00000008@	I*4	IDSN3		4	269						
AP-0000000C@	I*4	IN		4	148	149	160	172	173	176	177
				179	180	184	185	216	217	218	219
				223	224	232	240	242	244	294	296
				325	326	327	328	329	330		
2-00000198	I*4	J		179(2)=	180(2)=	184(2)=	185(2)=	194=	196	197	216(2)=
				217(2)=	218(2)=	219(2)=	281(2)=	282(2)=	284(2)=	285(2)=	286(2)=
				311=	312	313(2)	316=	317	326(2)=	327(2)=	328(2)=
				329(2)=	330(2)=	334=	336	337	344=	345(2)	402=
				403(2)	427=	429(3)	440=	441(7)	460=	462(2)	491=
				492(6)							
**	I*4	K		240(2)=	241(2)=	242(2)=	243(2)=	244(2)=	245(2)=	249=	250(2)
				251(2)							
AP-00000010@	I*4	LG		4	142(2)	143	183	189			
2-000001A8	I*4	NCMNU		224=	225	226	229	240	241	242	243
				244	245	249	253	255(2)	261	263A	
4-00000000	I*4	NM	COMM	117	149=	177	179	180	184	185	194
				203	209	216	217	218	219	223	253
				255(2)	261	272	279	281	282	283	284
				285	286	296	302(2)	306(2)	310	316	326
				327	328	329	330	334	388	402	411
				427	440	460	470(11)	476(2)	477(2)	484(3)	491
8-00000000	I*4	NMM1	COMM	130	203=	303	311	343	344		
6-00000038	I*4	NQ	COMM	120	149=	172	173	176	177	179	180
				184	185	195	206	217	218	219	223
				272	275	276	277	279	281	282	284
				285	286	296	300	326	327	328	329
				330	335	388	398	409	428	438	456
				465	475	477	481	489			
6-00000580	I*4	NQB	COMM	120	149=	205	272				
6-0000057C	I*4	NQI	COMM	120	149=	204	205	272			
2-000001A0	I*4	NQI1		204=	265A	320A	412				
2-000001A4	I*4	NQIB		205=	265A	320A	417				
**	I*4	NQM1		465=	466	475	477				
3-00000030	R*4	OMG	COMM	110	160=	273	350				
2-000001CC	R*4	PHO		232=	265A	294=	320A				
8-00000004	R*4	PI	COMM	130	137D	139	140	299			
2-00000190	R*4	RADDEG		140=	208	209					
7-00000000	R*4	RAS	COMM	128	160=	273	350				
2-000001BC	R*4	RERR		232=	234	253(2)	255				
**	R*4	RI1		399=	403						
2-000001AC	R*4	RNUC1		232=	234	247(2)=	251				
2-000001B0	R*4	RNUC2		232=	234	247					
2-000001D8	R*4	RR1		301=	302	313					
2-000001D4	R*4	TEM		299=	306	343=	345	458=	459	462	
**	R*4	TEMP		457=	462						
2-000001C4	R*4	VO		232=	265A	294=	320A				
2-000001C8	R*4	VOC		232=	265A	294=	320A				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
3-00000904	R*4	A	COMM	616	(14, 11)	110				
3-00001138	R*4	BLO	COMM	616	(14, 11)	110	180=	282	306(2)	441
3-00000034	R*4	CM	COMM	616	(14, 11)	110	217=	284	309=	310=
						312=	326=	492		
2-00000000	R*4	CMNU		80	(20)	134	242=	243	250(2)=	263A
8-00000008	R*4	COSRUM	COMM	56	(14)	130	207=	306	400	
2-00000050	R*4	CPT		44	(11)	134	263A	265A	317=	320A
2-0000007C	R*4	CPTNU		80	(20)	134	244=	245	263A	

3-0000029C	R*4	CTH	COMM	616	(14, 11)	110	179=	281	429	492
4-00000404	R*4	CUR	COMM	616	(14, 11)	117	467=	470=	474(2)=	475(2)=
						476(2)=	477(2)=	483	484	
3-00000C3C	R*4	DEN	COMM	616	(14, 11)	110	184=	187	196=	492
3-0000110C	R*4	DMF	COMM	44	(11)	110				
3-00000EA4	R*4	ENT	COMM	616	(14, 11)	110	185=	197=	441	
3-00000000	R*4	GS	COMM	44	(11)	110	216=	283	345=	
3-0000076C	R*4	H	COMM	616	(14, 11)	110	219=	286	328=	441
2-000000CC	R*4	PHID1		56	(14)	134	176=	208	277	483
2-00000104	R*4	PHIDN		56	(14)	134	176=	209	277	484
4-0000026C	R*4	PHIR	COMM	616	(14, 11)	117	208=	209=	306(2)	
6-0000003C	R*4	Q	COMM	616	(14, 11)	120	401=	403=	411	462
						492				
4-00000004	R*4	R	COMM	616	(14, 11)	117	218=	223(2)=	253(2)	255(2)
						260	261	285	296(2)=	301(2)
						302(2)	313=	327=	399	403
						429	467(4)	470(4)	492	
3-00000504	R*4	RCTH	COMM	616	(14, 11)	110	429=	441		
2-0000013C	R*4	RNU		80	(20)	134	240=	241	251(2)=	253(2)
						255(2)	260=	261=	263A	
6-000002A4	R*4	RUMR	COMM	616	(14, 11)	120				
6-00000000	R*4	RUMRQ	COMM	56	(14)	120	172=	207	275	306
						411	458			
8-00000040	R*4	SINRUM	COMM	56	(14)	130	459=			
4-0000073C	R*4	SM	COMM	616	(14, 11)	117	177(2)=	279(2)	483	484
5-00000000	R*4	TNA	COMM	616	(14, 11)	119	329=	336=	441	
5-00000268	R*4	WCTHD	COMM	616	(14, 11)	119	330=	337=	441	
4-000009A4	R*4	Z	COMM	616	(14, 11)	117	173=	276	411	457
						462=	467(6)	470(6)		
6-0000050C	R*4	ZQH	COMM	56	(14)	120				
6-00000544	R*4	ZQS	COMM	56	(14)	120				

PARAMETER CONSTANTS

Type	Name	References						
I*4	NCMMX	104#	134(3)	225	226			
I*4	NMMX	104#	110(10)	117(5)	119(2)	120(2)	134	
I*4	NQMX	104#	110(8)	117(5)	119(2)	120(5)	130(2)	134(2)

LABELS

Address	Label	References					
1-00000270	110'	152	153#				
0-0000052C	180	213	216#				
0-00000664	190	213	223#				
1-000002E8	192'	229	230#				
0-00000F50	200	213	294#				
0-00001208	210	213	325#				
0-000011B3	250	220	291	322	334#		
0-000013B8	300	331	342#				
1-0000034A	710'	350	351#				
1-00000465	740'	388	389#				
1-0000051A	910'	413	414#				
1-00000530	920'	417	418#				
0-000015B0	950	409	415	425#			
1-00000547	960'	433	434#				
1-000005CE	970'	446	447#				
1-00000683	990'	479	480#				
1-0000069F	1000'	486	488#				
1-000006C6	5905'	411	441	483	484	492	503#
1-000006CE	5965'	439	482	490	504#		

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References	
	ENTHAL	265	320
	FOR$CLOSE	289	
	FOR$OPEN	268	

```
R*4  MTH$COS                          207        306
R*4  MTH$SIN                          458

R*4  MTH$SQRT                         313
     NUF                              263

0001

+--------------------------------------------+
|          KEY TO REFERENCE FLAGS            |
|   =   - Value Modified                     |
|   #   - Defining Reference                 |
|   A   - Actual Argument, possibly modified |
|   D   - Data Initialization                |
|  (n)  - Number of occurrences on line      |
+--------------------------------------------+


COMMAND QUALIFIERS

  F/LIS/CHE/CRO NUFLOW

  /CHECK=(BOUNDS,OVERFLOW,UNDERFLOW)
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /STANDARD=(NOSYNTAX,NOSOURCE_FORM)
  /SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
  /WARNINGS=(GENERAL,NODECLARATIONS)
  /CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE
```

```
C
C++++++++++   SUBROUTINE QHIRSH ++++++++++
C
      SUBROUTINE QHIRSH (, IBFLOW, LG)                                IN

C (3037-QHITSH)
CGAS TO BE USED IF THE FLUID IS A GAS
CRUM SHOULD BE USED INSTEAD OF THE NEARBY STATEMENT IF Q-LINE IS NOT A
C        STRAIGHT LINE
C REFERENCED BY MAIN (SCM)
C REFERENCES AKIMAI, AKIMAO,DENSI, INTSPL (4 TIMES), MLINE (TWICE),
C        MTH$COS, MTH$SIGN, MTH$SIN (TWICE), MTH$SQRT, MTH$TAN,
C        QRZ (TWICE), & QPLINE (5 TIMES)
C TO BE COMPILED BY VAX FORTRAN V4.0-2.
C REVISED 19JUN86 FOR V3.1

      PARAMETER ( NMMX=11, NQMX=14)
      PARAMETER ( NWMX=MAX(NMMX,NQMX) )
      PARAMETER ( NVWMX=NWMX*7, NWSMX=NWMX*4-10)

CGAS      COMMON/GASCON/        SHR, GASC, CP, PREF, TREF
      COMMON/MFLOW/     GS(NMMX), G, OMG,
     +  CM(NQMX,NMMX), CTH(NQMX,NMMX),
     +  RCTH(NQMX,NMMX), H(NQMX,NMMX), A(NQMX,NMMX), DEN(NQMX,NMMX),
     +  ENT(NQMX,NMMX), DMF(NMMX), BLO(NQMX,NMMX)
CNOT     +       , DMBLO(NQMX,NMMX),
CNOT     +       DMRCTH(NQMX,NMMX), DMCM(NQMX,NMMX), WTH(NQMX,NMMX)
      COMMON/MLIN/      NM, R(NQMX,NMMX), PHIR(NQMX,NMMX),
     +  CUR(NQMX,NMMX), SM(NQMX,NMMX), Z(NQMX,NMMX)
      COMMON/NONAXI/    TNA(NQMX,NMMX), WCTHD(NQMX,NMMX)
      COMMON/QLIN/      RUMRQ(NQMX), NQ, Q(NQMX,NMMX)
CNOT     +       , RUMR(NQMX,NMMX), ZQH(NQMX), ZQS(NQMX), NQI, NQB
      COMMON/XY/        RAS, XM(NQMX,NMMX)

      COMMON/IQ/        ICM
      COMMON/IQS/       DMPCM, DMPG, DMPL, EPSCM, EPSG, EPSL
      COMMON/QS/        ITRL, MXITC, MXITG

      DIMENSION CMSML(NQMX), CMN(NMMX), COE(NMMX),
     +  DMBLO(NQMX,NMMX), DMCM(NQMX,NMMX),
     +  SAVE(NMMX), VW(NVWMX),
     +  WTH(NQMX,NMMX), WF(NWMX), WF1(NWMX), WF2(NWMX), WINT(NWMX),
     +  WL(NWMX), WORK(NWSMX), WX(NWMX)

C3100 CONSTANTS *****
      DATA      CMSMLC/.01/, MAXITC, MAXITG, MAXITL/ 50, 50, 200/,
     +  PI/3.14159265359/

C3105 INITIAL *****
      DATA      MXITC, MXITG/ 0, 0/

      IF (LG.LE.1)      CP=999.
      N1        =0
      N2        =NM
      N3        =NM+NM
      N4        =N3+NM
      N5        =N4+NM
      N6        =N5+NM
      N7        =N6+NM
      NM2P3     =N3+3
      NM7       =N7+NM
      NMM1      =NM-1
      NMM2      =NMM1-1
      NMS       =N5-10
      NQP3      =NQ+3
      NQS       =NQ*4-10
      PI2       =PI+PI

CINQUE      WRITE(6,3120)
CINQUE 3120 FORMAT(1H1//1H010X'+++++ WELCOME TO SUBROUTINE QHIRSH',' +++++'//)

C3200 *****
      TEM       =(G+G)*CMSMLC/PI
      DO        I=1,NQ
        CMSML(I)
     +  =TEM/ ( ( DEN(I,1)+DEN(I,NM) )*( R(I,1)+R(I,NM) )*Q(I,NM) )
        END DO
```

```
      ISTOPG    =0
      ISTOPCM   =0

      CALL    QRZ( NQ, NM, Q, R, Z, RUMRQ)
CRUM            CALL QRZ(NQ,NM,Q,R,Z,RUMR)

      ITRL      =0
C3300 ITERATE UNTIL ICONML=0
 3300 ITRL      =ITRL+1
      IF (ITRL.GT.MAXITL) THEN
        WRITE(6,3310) ITRL
 3310   FORMAT(1H0,10X,'***** CONVERGENCE OF THE MERIDIONAL STREAMLINE'
     +         ,' SOLUTION WAS NOT ACHIEVED *****'/' ITRL ='I4)
        PRINT *, 'NOT CONVERGED... MXITC, MXITG, ITRL =', MXITC, MXITG,
     +                ITRL
        ICON    =ITRL
        STOP 'QHIRTH.3315'
C       RETURN
        END IF

C3320
      CALL      MLINE ( 0, NQ, NM, RAS, Q, R, Z, SM, PHIR, CUR, XM,
     +                  ICON, WF, WL, WF1, WF2)

      LINE      =3320
      IF (ICON.NE.0) GOTO 9000
      DO  3350 J=1,NM
        DO      I=1,NQ
        WX(I)=SM(I,J)
        WF(I)=DEN(I,J)
          END DO

C3330
        CALL    SPLINE( WX, WF, NQ, WL, WF1, WF2, WORK, NQS, ICON)

        LINE    =3330
        IF (ICON.NE.0) GOTO 9000
        DO      I=1,NQ
          DMCM(I,J)=WF1(I)
          END DO
        DO      I=1,NQ
          WF(I)=BLO(I,J)
          END DO

C3340
        CALL    AKIMAO( WX, WF, NQ, WF1, WF2, WORK, NQP3, ICON)

        LINE    =3340
        IF (ICON.NE.0) GOTO 9000
        DO      I=1,NQ
          DMBLO(I,J)=WF1(I)
          END DO
 3350   CONTINUE

        SUMRES=0.
        ICONML=0
        DO 3700I=1,NQ
        CMSMLI  =CMSML(I)
        RUMRQI  =RUMRQ(I)
        DO 3420J=1,NM
          IF ( R(I,J).LT.1.E-6 .AND. RCTH(I,J).NE.0. ) THEN
            WRITE(6,3410) I, J, RCTH(I,J), I, J, R(I,J)
 3410       FORMAT(1H010X'***** ERROR IN SUBROUTINE QHIRSH ; RCTH('I3'.'
     +         I3')='E12.5' AT R('I3','I3')='E12.5' *****')
            ICON=-10
            STOP 'QHIRSH.3410'
C           RETURN
            END IF
          IF ( RCTH(I,J).EQ.0. .AND. R(I,J).LT.1.E-6 )
     *          CTH(I,J)=-WCTHD(I,J)
          IF ( R(I,J) .GE. 1.E-6)   CTH(I,J)=RCTH(I,J)/R(I,J)-WCTHD(I,J)
CSKIP     WTH(I,J)=CTH(I,J)-OMG*R(I,J)
          IF (LG.EQ.1 .OR. IBFLOW.GT.0) GOTO 3420
          DEN(I,J)        =DENSI( H(I,J), ENT(I,J), CM(I,J), CTH(I,J))
 3420   CONTINUE

        DO      J=1,NM
          WX(J) =G(I,J)
          WF(J) =H(I,J)
          END DO
```

```
C3430
         CALL    SPLINE( WX, WF, NM, WL, WF1, WF2, WORK, NMS, ICON)

         LINE    =3430
         IF(ICON.NE.0) GO TO 9000
         DO      J=1,NM
           VW(N1+J)       =WF1(J)
           WF(J) =ENT(I,J)
           END DO

C3440
         CALL    SPLINE( WX, WF, NM, WL, WF1, WF2, WORK, NMS, ICON)

         LINE    =3440
         IF(ICON.NE.0) GO TO 9000
         DO      J=1,NM
           VW(N2+J)       =WF1(J)
           WF(J) =CTH(I,J)*R(I,J)
           END DO

C3450
         CALL    SPLINE( WX, WF, NM, WL, WF1, WF2, WORK, NMS, ICON)

         LINE    =3450
         IF(ICON.NE.0) GO TO 9000
         DO      J=1,NM
           VW(N3+J)       =WF1(J)
           WF(J) =PHIR(I,J)
           END DO

C3460
         CALL    SPLINE( WX, WF, NM, WL, WF1, WF2, WORK, NMS, ICON)

         LINE    =3460
         IF(ICON.NE.0) GO TO 9000
         DO      J=1,NM
           VW(N4+J)       =WF1(J)
           END DO

         DO 3480J=1,NM
           IF ( R(I,J) .LT. 1.E-5) THEN
             VW(N5+J)     =0.
             VW(N6+J)     =0.
             GOTO 3480
             END IF
CRUM       EP     =PHIR(I,J)-RUMR(I,J)
           EP     =PHIR(I,J)-RUMRQI
           CE     =COS(EP)
           COE(J)=CE
           DMCM(I,J)      =-( DMBLO(I,J)/BLO(I,J) + VW(N4+J)/CE
     +            + CUR(I,J)*TAN(EP) + SIN( PHIR(I,J) )/R(I,J)
     +            + DMCM(I,J)/DEN(I,J) ) * CM(I,J)
           VW(N5+J)       =2.*( CUR(I,J)*CE - DMCM(I,J)*SIN(EP)/CM(I,J) )
     +            - VW(N2+J)/CP
           VW(N6+J)       =2.*( VW(N1+J) - CTH(I,J)*VW(N3+J)/R(I,J)
     +            - ( H(I,J) - ( CTH(I,J)*CTH(I,J) )*.5 )*VW(N2+J)/CP
     +            + TNA(I,J)/DEN(I,J) )
  3480     CONTINUE

C3500 *****
         MXITG   =MAX( MXITG, ITRG)
         ITRG    =0
         CMC     =CM(I,1)

  3510   ITRG    =ITRG+1
         IF (ITRG.GT.MAXITG) THEN
           WRITE(6,3520) I, ITRL
  3520     FORMAT(1H010X
     +            '***** OVERALL CONTINUITY WAS NOT SATISFIED AT I='I3
     +            ' , ITRL='I5)
           IF (IREV.EQ.0) THEN
             WRITE(6,3530)
  3530       FORMAT(1H+,74X,' *****')
            ELSE
             WRITE(6,3540)
  3540       FORMAT(1H+,74X,', BECAUSE REVERSE FLOW OCCURS. *****')
            END IF
           ISTOPG=ISTOPG+1
           GOTO 3700
```

```
          END IF
        IREV    =0
        MXITC   =MAX( MXITC, ITRCM)
        ITRCM   =0
 3600   ITRCM   =ITRCM+1
        IF (ITRCM.GT.MAXITC) THEN
          WRITE(6,3610) I, ITRL
 3610     FORMAT(1H010X'***** CM CONVERGENCE WAS NOT ACHIEVED AT I ='I3
     +          ' , ITRL='I5' *****')
          ISTOPCM       =ISTOPCM+1
          GOTO 3700
          END IF
        CMN(1)  =CMC
        TEM     =0.
        IF (ICM.EQ.2) THEN
          DO   J=1,NM
            WF(J)       =( VW(N6+J) - VW(N5+J)*CM(I,J)*CM(I,J) )*.5
            END DO
C3620
          CALL    INTSPL( WX, WF, NM, WINT, WL, WF1, WF2, WORK, NMS,
     +                      ICON)
          LINE  =3620
          IF (ICON.NE.0) GOTO 9000
          DO   J=2,NM
            TEM =WINT(J)+TEM
            CMN(J)      =SIGN( SQRT( ABS(TEM) ), TEM) + CMC
            END DO
         ELSE
          DO   J=1,NM
            WF(J)       =( VW(N6+J)/CM(I,J) - VW(N5+J)*CM(I,J) )*.5
            END DO
C3621
          CALL    INTSPL( WX, WF, NM, WINT, WL, WF1, WF2, WORK, NMS,
     +                      ICON)
          LINE  =3621
          IF (ICON.NE.0) GOTO 9000
          DO   J=2,NM
            TEM =WINT(J)+TEM
            CMN(J)      =TEM+CMC
            END DO
         END IF
        ICONCM  =0
        DO      J=1,NM
          IF ( ABS( 1. - CMN(J)/CM(I,J) ) .GE. EPSCM )   ICONCM=ICONCM+1
CDEGUG            PRINT *, 'I, J, CMN, CM =', I, J, CMN(J), CM(I,J)
          IF (CMN(J).LT.CMSMLI) THEN
            IREV=IREV+1
CCC         THE ORIGINAL PROGRAM DOES NOT PRINT & STOP HERE ---
             PRINT *,   'REVERSE FLOW?? I,J,CMN,CMAV,EPSCM,ICONCM,CM =',
     +                  I, J, CMN, CMSMLI/CMSMLC, EPSCM, ICONCM,
     +                  ( CM(I,K), K=1,NM)
             WRITE(6,*) 'REVERSE FLOW?? I,J,CMN,CMAV,EPSCM,ICONCM,CM =',
     +                  I, J, CMN, CMSMLI/CMSMLC, EPSCM, ICONCM,
     +                  ( CM(I,K), K=1,NM)
             STOP 'QHIRSH.3628'
           ELSE
            CM(I,J)     =CM(I,J) + DMPCM*( CMN(J)-CM(I,J) )
           END IF
          END DO
CDEBUG  PRINT *, 'I, ICONCM, ITRG, ITRCM, CM =', I, ICONCM, ITRG, ITRCM,
CDEBUG     +                     ( CM(I,J), J=1,NM)
        IF (IREV.EQ.0 .AND. ICONCM.NE.0) GOTO 3600
        DO      J=1,NM
          SAVE(J)       =PI2 * R(I,J) * DEN(I,J) * COE(J) * BLO(I,J)
          WF(J) =CM(I,J)*SAVE(J)
CRUM      WF(J) =PI2 * R(I,J) * DEN(I,J) * CM(I,J)
CRUM     +              * COS( PHIR(I,J)-RUMR(I.J) ) * BLO(I,J)
          END DO
```

```
C3640
        CALL    INTSPL( WX, WF, NM, WINT, WL, WF1, WF2, WORK, NMS, ICON)
        LINE    =3640
        IF(ICON.NE.0) GO TO 9000
        VW(N7+1)=0.0
        DO      J=2,NM
          VW(N7+J)       =VW(N7+J-1)+WINT(J)
          END DO
        IF ( ABS( 1. - VW(NM7)/G ) .LT. EPSG ) GOTO 3660
CDEBUG          PRINT *, ' I, ITRG, VW(NM7), G, EPSG =', I, ITRG, VW(NM7), G,
CDEBUG     +                    EPSG
CRUM    DO       J=1,NM
CRUM      SAVE(J) =PI2 * R(I,J) * DEN(I,J) * COS( PHIR(I,J)-RUMR(I,J) )
CRUM     +               *BLO(I,J)
CRUM      END DO
C3650
        CALL    INTSPL( WX, SAVE, NM, WINT, WL, WF1, WF2, WORK, NMS, ICON)
        IF (ICON.NE.0) GOTO 9000
        INTW=0.0
        DO      J=2,NM
          INTW  =INTW+WINT(J)
          END DO
        CMC     =CMC + DMPG*( G-VW(NM7) )/INTW
        GOTO 3510
 3660   DO      J=1,NM
          WINT(J)        =VW(N7+J)
          END DO
C3670
        CALL    AKIMAI( WINT, WX, NM, GS, NMM2, WL, WF1, WORK, NM2P3,
     *                  ICON)
C       LINE    =3670
C       IF(ICON.NE.0) GO TO 9000
        DO      J=1,NMM2
          WL(NM-J)       =WL(NMM1-J)
          END DO
        RES     =0.0
        DO      J=2,NMM1
          RES   =RES+ABS(WL(J)-Q(I,J))**2
          IF ( ABS( 1.-WL(J)/Q(I,J) ) .GE. EPSL)      ICONML=ICONML+1
          Q(I,J)=Q(I,J)+DMPL*(WL(J)-Q(I,J))
          END DO
        CALL    QRZ( NQ, NM, Q, R, Z, RUMRQ)
CRUM    CALL QRZ(NQ,NM,Q,R,Z,RUMR)
        SUMRES=SUMRES+RES
 3700   CONTINUE
      PRINT*, 'SUMRES, MXITC, MXITG, ITRL =', SUMRES, MXITC, MXITG, ITRL
      IF (ISTOP.NE.0) THEN
        ICON    =ISTOPCM+ISTOG
        PRINT *, 'ISTOPCM, ISTOPG =', ISTOPCM, ISTOPG
        STOP 'QHIRSH.3815'
C       RETURN
        END IF
      IF(ICONML.NE.0) GO TO 3300
C3820
      CALL      MLINE ( 1, NQ, NM, RAS, Q, R, Z, SM, PHIR, CUR, XM,
     *                  ICON, WF, WL, WF1, WF2)
      LINE      =3820
      IF (ICON.NE.0) GOTO 9000
CINOUE      WRITE(6,3830)
CINOUE 3830 FORMAT(1H010X'+++++ CONVERGENGE OF THE MERIDIONAL STREAMLINE'
CINOUE     +       ,' SOLUTION WAS ACHIEVED +++++')
      MXITC     =MAX( MXITC, ITERCM)
      MXITG     =MAX( MXITG, ITERG)
      ICON      =0
      RETURN
 9000 PRINT *, 'ICON = ', ICON,
```

```
0398          +  '    LINE, I, J, ITRL = ', LINE, I, J, ITRL
0399             STOP 'QHIRSH.9010'
0400    C         RETURN
0401
0402             END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	4763	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	614	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	3632	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3	MFLOW	5024	PIC OVR REL GBL SHR NOEXE RD WRT LONG
4	MLIN	3084	PIC OVR REL GBL SHR NOEXE RD WRT LONG
5	NONAXI	1232	PIC OVR REL GBL SHR NOEXE RD WRT LONG
6	QLIN	676	PIC OVR REL GBL SHR NOEXE RD WRT LONG
7	XY	620	PIC OVR REL GBL SHR NOEXE RD WRT LONG
8	IQ	4	PIC OVR REL GBL SHR NOEXE RD WRT LONG
9	IQS	24	PIC OVR REL GBL SHR NOEXE RD WRT LONG
10	QS	12	PIC OVR REL GBL SHR NOEXE RD WRT LONG
	Total Space Allocated	19685	

ENTRY POINTS

Address	Type	Name	References
0-00000000		QHIRSH	4#

VARIABLES

Address	Type	Name	Attributes	References
**	R*4	CE		206= 207 208 211
**	R*4	CMC		221= 252 268 285 345(2)=
**	R*4	CMSMLC		46D 73 297 300
2-00000BE0	R*4	CMSMLI		135= 294 297 300
2-00000B84	R*4	CP		52= 211 213
9-00000000	R*4	DMPCM	COMM	36 305
9-00000004	R*4	DMPG	COMM	36 345
9-00000008	R*4	DMPL	COMM	36 365
**	R*4	EP		205= 206 208 211
9-0000000C	R*4	EPSCM	COMM	36 292 297 300
9-00000010	R*4	EPSG	COMM	36 328
9-00000014	R*4	EPSL	COMM	36 364
3-0000002C	R*4	G	COMM	22 73(2) 328 345
2-00000B88	I*4	I		74= 75(6) 105= 106(2) 107(2) 115= 116(2) 118= 119(2) 127= 128(2) 134= 135 136 138(2) 139(4) 146(4) 148(5) 151(5) 155 156 166 176(2) 186 199 205 208(9) 211(3) 213(7) 221 225 245 257(2) 274(2) 292 297(2) 300(2) 305(3) 313(3) 314 363 364 365(3) 397
AP-00000004@	I*4	IBFLOW		4 150
8-00000000	I*4	ICM	COMM	35 254
2-00000BC4	I*4	ICON		93= 99A 103 111A 114 123A 126 142= 160A 163 170A 173 180A 183 190A 193 261A 265 278A 282 320A 323 338A 340 353A 376= 384A 388 394= 397
**	I*4	ICONCM		290= 292(2)= 297 300 310
2-00000BD8	I*4	ICONML		133= 364(2)= 381
**	I*4	INTW		341= 343(2)= 345
2-00000BEC	I*4	IREV		229 240= 295(2)= 310
**	I*4	ISTOG		376
2-00000C00	I*4	ISTOP		375
2-00000BC0	I*4	ISTOPCM		79= 248(2)= 376 377
2-00000B8C	I*4	ISTOPG		78= 236(2)= 377
2-00000C04	I*4	ITERCM		392
2-00000C08	I*4	ITERG		393
2-00000BF0	I*4	ITRCM		241 242= 243(2)= 244

2-00000BE8	I*4	ITRG		219	220=	223(2)=	224				
10-00000000	I*4	ITRL	COMM	37	84=	86(2)=	87	88	91	93	225
				245	374	397					
2-00000BCC	I*4	J		104=	106	107	116	119	128	137=	138(2)
				139(4)	146(4)	148(5)	151(5)	154=	155(2)	156(2)	164=
				165(2)	166(2)	174=	175(2)	176(3)	184=	185(2)	186(2)
				194=	195(2)	198=	199	200	201	205	207
				208(10)	211(5)	213(11)	256=	257(5)	266=	267	268
				273=	274(5)	283=	284	285	291=	292(2)	294
				297	300	305(4)	312=	313(5)	314(3)	325=	326(3)
				342=	343	348=	349(2)	358=	359(2)	362=	363(2)
				364(2)	365(4)	397					
**	I*4	K		297(2)=	300(2)=						
AP-00000008a	I*4	LG		4	52	150					
2-00000BC8	I*4	LINE		102=	113=	125=	162=	172=	182=	192=	264=
				281=	322=	387=	397				
**	I*4	MAXITC		46D	244						
**	I*4	MAXITG		46D	224						
**	I*4	MAXITL		46D	87						
10-00000004	I*4	MXITC	COMM	37	50D	91	241(2)=	374	392(2)=		
10-00000008	I*4	MXITG	COMM	37	50D	91	219(2)=	374	393(2)=		
**	I*4	N1		53=	165	213					
2-00000B88	I*4	N2		54=	175	211	213				
2-00000B8C	I*4	N3		55=	56	60	185	213			
2-00000B90	I*4	N4		56=	57	195	208				
2-00000B94	I*4	N5		57=	58	64	200	211	257	274	
2-00000B98	I*4	N6		58=	59	201	213	257	274		
2-00000B9C	I*4	N7		59=	61	324	326(2)	349			
4-00000000	I*4	NM	COMM	28	54	55(2)	56	57	58	59	61
				62	75(3)	81A	99A	104	137	154	160A
				164	170A	174	180A	184	190A	194	198
				256	261A	266	273	278A	283	291	297
				300	312	320A	325	338A	342	348	353A
				359	368A	384A					
2-00000BA0	I*4	NM2P3		60=	353A						
**	I*4	NM7		61=	328	345					
**	I*4	NMM1		62=	63	359	362				
2-00000BA4	I*4	NMM2		63=	353A	358					
2-00000BA8	I*4	NMS		64=	160A	170A	180A	190A	261A	278A	320A
				338A							
6-00000038	I*4	NQ	COMM	31	65	66	74	81A	99A	105	111A
				115	118	123A	127	134	368A	384A	
2-00000BAC	I*4	NQP3		65=	123A						
2-00000BB0	I*4	NQS		66=	111A						
3-00000030	R*4	OMG	COMM	22							
**	R*4	PI		46D	67(2)	73					
**	R*4	PI2		67=	313						
7-00000000	R*4	RAS	COMM	33	99A	384A					
**	R*4	RES		361=	363(2)=	371					
**	R*4	RUMRQI		136=	205						
2-00000BD4	R*4	SUMRES		132=	371(2)=	374					
2-00000BB4	R*4	TEM		73=	75	253=	267(2)=	268(2)	284(2)=	285	

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
3-00000904	R*4	A	COMM	616	(14, 11)	22				
3-00001138	R*4	BLO	COMM	616	(14, 11)	22	119	208	313	
3-00000034	R*4	CM	COMM	616	(14, 11)	22	151A	208	211	221
						257(2)	274(2)	292	297	300
						305(3)=	314			
2-00000038	R*4	CMN		44	(11)	39	252=	268=	285=	292
						294	297	300	305	
2-00000000	R*4	CMSML		56	(14)	39	75=	135		
2-00000064	R*4	COE		44	(11)	39	207=	313		
3-0000029C	R*4	CTH	COMM	616	(14, 11)	22	146=	148=	151A	176
						213(3)				
4-000004D4	R*4	CUR	COMM	616	(14, 11)	28	99A	208	211	384A
3-00000C3C	R*4	DEN	COMM	616	(14, 11)	22	75(2)	107	151=	208
						213	313			

2-00000090	R*4	DMBLO		616	(14, 11)	39	128=	208		
2-000002F8	R*4	DMCM		616	(14, 11)	39	116=	208(2)=	211	
3-0000110C	R*4	DMF	COMM	44	(11)	22				
3-00000EA4	R*4	ENT	COMM	616	(14, 11)	22	151A	166		
3-00000000	R*4	GS	COMM	44	(11)	22	353A			
3-0000076C	R*4	H	COMM	616	(14, 11)	22	151A	156	213	
4-0000026C	R*4	PHIR	COMM	616	(14, 11)	28	99A	186	205	208
						384A				
6-0000003C	R*4	Q	COMM	616	(14, 11)	31	75	81A	99A	155
						363	364	365(3)=	368A	384A
4-00000004	R*4	R	COMM	616	(14, 11)	28	75(2)	81A	99A	138
						139	146	148(2)	176	199
						208	213	313	368A	384A
3-00000504	R*4	RCTH	COMM	616	(14, 11)	22	138	139	146	148
6-00000000	R*4	RUMRQ	COMM	56	(14)	31	81A	136	368A	
2-00000560	R*4	SAVE		44	(11)	39	313=	314	338A	
4-0000073C	R*4	SM	COMM	616	(14, 11)	28	99A	106	384A	
5-00000000	R*4	TNA	COMM	616	(14, 11)	30	213			
2-0000058C	R*4	VW		392	(98)	39	165=	175=	185=	195=
						200=	201=	208	211(2)=	213(4)=
						257(2)	274(2)	324=	326(2)=	328
						345	349			
5-00000268	R*4	WCTHD	COMM	616	(14, 11)	30	146	148		
2-0000097C	R*4	WF		56	(14)	39	99A	107=	111A	119=
						123A	156=	160A	166=	170A
						176=	180A	186=	190A	257=
						261A	274=	278A	314=	320A
						384A				
2-00000984	R*4	WF1		56	(14)	39	99A	111A	116	123A
						128	160A	165	170A	175
						180A	185	190A	195	261A
						278A	320A	338A	353A	384A
2-000009EC	R*4	WF2		56	(14)	39	99A	111A	123A	160A
						170A	180A	190A	261A	278A
						320A	338A	384A		
2-00000A24	R*4	WINT		56	(14)	39	261A	267	278A	284
						320A	326	338A	343	349=
						353A				
2-00000A5C	R*4	WL		56	(14)	39	99A	111A	160A	170A
						180A	190A	261A	278A	320A
						338A	353A	359(2)=	363	364
						365	384A			
2-00000A94	R*4	WORK		184	(46)	39	111A	123A	160A	170A
						180A	190A	261A	278A	320A
						338A	353A			
2-00000714	R*4	WTH		616	(14, 11)	39				
2-00000B4C	R*4	WX		56	(14)	39	106=	111A	123A	155=
						160A	170A	180A	190A	261A
						278A	320A	338A	353A	
7-00000004	R*4	XM	COMM	616	(14, 11)	33	99A	384A		
4-00000[illegible]A4	R*4	Z	COMM	616	(14, 11)	28	81A	99A	368A	384A

PARAMETER CONSTANTS

Type	Name	References							
I*4	NMMX	17#	18	22(10)	28(5)	30(2)	31	33	39(6)
I*4	NQMX	17#	18	22(8)	28(5)	30(2)	31(2)	33	39(4)
I*4	NVWMX	19#	39						
I*4	NWMX	18#	19(2)	39(6)					
I*4	NWSMX	19#	39						

LABELS

Address	Label	References		
0-0000015C	3300	86#	381	
1-000000DC	3310'	88	89#	
**	3350	104	130#	
1-00000140	3410'	139	140#	
0-00000565	3420	137	150	152#
0-00000A32	3480	198	202	216#

```
0-00000A74  3510                     223#    346
1-0000019C  3520'                    225     226#
1-000001E2  3530'                    230     231#
1-000001F0  3540'                    233     234#

0-00000B04  3600                     243#    310
1-0000021C  3610'                    245     246#
0-00001034  3660                     328     348#
0-00001148  3700                     134     237     249     372#
0-00001234  9000                     103     114     126     163     173     183     193     265
                                     282     323     340     388     397#
```

FUNCTIONS AND SUBROUTINES REFERENCED

```
Type  Name                        References

      AKIMAI                       353
      AKIMAO                       123
 R*4  DENSI                        151
      INTSPL                       261     278     320     338
      MLINE                         99     384

 R*4  MTH$COS                      206
 R*4  MTH$SIGN                     268
 R*4  MTH$SIN                      208     211
 R*4  MTH$SQRT                     268
 R*4  MTH$TAN                      208

      QRZ                           81     368
      SPLINE                       111     160     170     180     190
```

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|   =   - Value Modified                        |
|   #   - Defining Reference                    |
|   A   - Actual Argument, possibly modified    |
|   D   - Data Initialization                   |
|  (n)  - Number of occurrences on line         |
+-----------------------------------------------+
```

COMMAND QUALIFIERS

```
F/LIS/CHE/CRO QHIRSH

/CHECK=(BOUNDS,OVERFLOW,UNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE
```

```
0001   C
0002   C+++++++++      SUBROUTINE QRZ         +++++++++
0003   C
0004         SUBROUTINE QRZ( NQ, NM, Q, R, Z, RUMRQ)
0005
0006   C REFERENCED BT QHIRSH (TWICE)
0007   C TO BE COMPILED BY VAX FORTRAN V4.0-2
0008   C REVISED 15MAY86
0009
0010         PARAMETER ( NQMX=14)
0011
0012         COMMON/NU/        NMM1, PI, COSRUM(NQMX), SINRUM(NQMX)
0013
0014   CRUM      DIMENSION     Q(NQMX,NM), R(NQMX,NM), RUMR(NQMX,NM), Z(NQMX,NM)
0015         DIMENSION Q(NQMX,NM), R(NQMX,NM), RUMRQ(NQMX), Z(NQMX,NM)
0016
0017         DO       I=1,NQ
0018           CO      =COSRUM(I)
0019           SI      =SINRUM(I)
0020           DO     J=2,NMM1
0021   CRUM    RUMR(I,J)=RUMR(I,1)
0022   CRUM    Z(I,J)=Z(I,1)-Q(I,J)*SIN(RUMR(I,J))
0023   CRUM    R(I,J)=R(I,1)+Q(I,J)*COS(RUMR(I,J))
0024             Z(I,J)=Z(I,1) - Q(I,J)*SI
0025             R(I,J)=R(I,1) + Q(I,J)*CO
0026             END DO
0027           END DO
0028
0029           RETURN
0030           END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes								
0	$CODE	298	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
2	$LOCAL	188	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	LONG
3	NU	120	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG

Total Space Allocated 606

ENTRY POINTS

Address	Type	Name	References
0-00000000		QRZ	4#

VARIABLES

Address	Type	Name	Attributes	References				
**	R*4	CO		18=	25			
2-00000000	I*4	I		17=	18	19	24(3)	25(3)
**	I*4	J		20=	24(2)	25(2)		
AP-00000008@	I*4	NM		4	15(3)			
3-00000000	I*4	NMM1	COMM	12	20			
AP-00000004@	I*4	NQ		4	17			
3-00000004	R*4	PI	COMM	12				
**	R*4	SI		19=	24			

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References			
3-00000008	R*4	COSRUM	COMM	56	(14)	12	18		
AP-0000000C@	R*4	Q		**	(14, *)	4	15	24	25
AP-00000010@	R*4	R		**	(14, *)	4	15	25(2)=	
AP-00000018@	R*4	RUMRQ		56	(14)	4	15		
3-00000040	R*4	SINRUM	COMM	56	(14)	12	19		
AP-00000014@	R*4	Z		**	(14, *)	4	15	24(2)=	

PARAMETER CONSTANTS

Type	Name	References		
I*4	NQMX	10#	12(2)	15(4)

```
C
C+++++++++++++     SUBROUTINE SOLOUT     +++++++++++++++
C
      SUBROUTINE SOLOUT( !DSN3, ITRB, LG, TODAY)                  IN

C REFERENCED BY MAIN(SCM); REFERENCES none
C TO BE COMPILED BY VAX FORTRAN V4.0-2
C REVISED 19OCT87

      CHARACTER TODAY*9

      PARAMETER ( NMMX=11, NQMX=14)

        COMMON /DESIGN/BSBE,BE,RT,RPM,UT,EFFI,AVIL,VELR,HEADAD,FLOWC
     + ,FLOWR,CONST,POW
        COMMON /GASCON/SHR,GASC,CP,POSABS,TOSABS
      COMMON/MFLOW/     GS(NMMX), G, OMG,
     + CM(NQMX,NMMX), CTH(NQMX,NMMX),
     * RCTH(NQMX,NMMX), H(NQMX,NMMX), A(NQMX,NMMX), DEN(NQMX,NMMX),
     * ENT(NQMX,NMMX), DMF(NMMX), BLO(NQMX,NMMX), DMBLO(NQMX,NMMX),
     * DMRCTH(NQMX,NMMX), DMCM(NQMX,NMMX), WTH(NQMX,NMMX)
      COMMON/MLIN/      NM, R(NQMX,NMMX), PHIR(NQMX,NMMX),
     * CUR(NQMX,NMMX), SM(NQMX,NMMX), Z(NQMX,NMMX)
      COMMON/QLIN/      RUMRQ(NQMX), NQ, Q(NQMX,NMMX), RUMR(NQMX,NMMX),
     * ZQH(NQMX), ZQS(NQMX), NQI, NQB
        COMMON /STAN/STANP,STANT,STANDE
        COMMON /SCROLL/ALPSC,DHU,BS
      COMMON/ROTO/      ZLE(2), ZTE(2), NB, IPRF(NMMX), TMPC(NMMX),
     * AXIM(NMMX), ZHAXI, RUMAXI, IAXI, TTE
      COMMON/XY/        RAS, XM(NQMX,NMMX)
      COMMON/LOAD/      PCON(NMMX), IPCON(NMMX)
      COMMON/NONAXI/    TNA(NQMX,NMMX), WCTHD(NQMX,NMMX)
CCC     COMMON /BFLOW/IQ1,IQ2,INQ,NIN,RB(51,15),SMB(51,15),XB(51,15)
CCC     + ,BLOB(51,15),DTH1R(51,15),PHIB(51,15),THEMB(51,15)
      COMMON/IQS/       DMPCM, DMPG, DMPL, EPSCM, EPSG, EPSL
      COMMON/QS/        ITRL, MXITC, MXITG

      DATA      PI/3.14159265359/

      DEGGRA    =180./PI
      WRITE(6,610) MAX(ITRB,0)+1
  610 FORMAT('1'11X'+++++ 'I2
     + '-TH THROUGH FLOW SOLUTION BY USE OF THE EXTENDED NOVAK''S ',
     + 'METHOD  +++++')
      WRITE(6,615) EPSCM, EPSG, EPSL, DMPCM, DMPG, DMPL, MXITC, MXITG,
     +         ITRL
  615 FORMAT(//20X'ITERATION CONDITION: EPSCM='F10.7/41X'EPSG ='F10.7/
     + 41X'EPSL ='F10.7/41X'DMPCM='F5.2/41X'DMPG ='F5.2/41X'DMPL ='
     + F5.2/20X'ITERATION TIMES:    MXITC ='I4/40X'MXITG ='I4/41X
     + 'ITRL ='I4)

CCC      WRITE(6,670)
CCC  670 FORMAT(/////11X
CCC     +          '+++++ DESIGN CONDITIONS AT THE WORKING STATIONS +++++'///
CCC     + 11X'J      RCTH'12X'H'10X'ENT'10X'BLO'10X'TNA'8X'WCTHD'/)
CCC      DO       I=1,NQ
CCC             WRITE(6,671) I
CCC             DO      J=1,NM
CCC               WRITE(6,672) J, RCTH(I,J), H(I,J), ENT(I,J), BLO(I,J),
CCC     +                    TNA(I,J), WCTHD(I,J)
CCC             END DO
CCC      END DO

        WRITE(6,611)
  611 FORMAT(/////11X'+++++ MERIDIONAL STREAMLINE OUTPUT +++++'///
     + 11X'J'8X'Q'12X'Z'12X'R'10X'PHI'10X'CUR'12X'M'10X'DMF'/)
        DO 10 I=1,NQ
        WRITE(6,671) I
        DO 10 J=1,NM
        PHI     =PHIR(I,J) * DEGGRA
          WRITE(6,672) J, Q(I,J), Z(I,J), R(I,J), PHI, CUR(I,J),
     +                 SM(I,J), DMF(J)
 10     CONTINUE
        WRITE(6,630) RAS
  630 FORMAT(/////11X'+++++ FLOW FIELD OUTPUT +++++'T82,'( RAS='E12.5')'
     + ///11X'J       CTH'11X'CR'11X'CZ'11X'CM'12X'C'10X'ALP'10X'DEN'9X
     + 'MACH'12X'X'/)
```

```
          DO 20 I=1,NQ
          WRITE(6,671) I
          DO 20 J=1,NM
          CZ=CM(I,J)*COS(PHIR(I,J))
          CR=CM(I,J)*SIN(PHIR(I,J))
          CTHAV=RCTH(I,J)/R(I,J)
          C=SQRT(CM(I,J)**2+CTHAV**2)
          ALP     =ATAN( CTHAV/CM(I,J) ) * DEGGRA

        IF (LG.EQ.2) THEN
          A0      =SQRT((SHR-1.)*H(I,J))
          A(I,J)  =A0
     *            *SQRT(1.-(SHR-1.)*(CTH(I,J)**2+CM(I,J)**2)/(2.*A0**2))
        ELSE
          A(I,J)  =C*10.
        END IF

          RMACH=C/A(I,J)
            WRITE(6,672) J, CTHAV, CR, CZ, CM(I,J), C, ALP, DEN(I,J),
     +                    RMACH, XM(I,J)
   20     CONTINUE

        IF (IDSN3.GT.0) THEN
          OPEN( 2, FILE='DSN3ZI', STATUS='NEW')                           OPEN2
          WRITE(2,640) TODAY
  640     FORMAT(' OUTPUT FROM SCM, FOR INPUT TO DSN3'T63,A9)
          WRITE(2,*) NQB, NM
          NQI1    =NQI+1
          NQIB    =NQI+NQB
          WRITE(2,*)   ( RCTH(NQI1,J)/R(NQI1,J), J=1,NM)
          WRITE(2,*)   ( RCTH(NQIB,J)/R(NQIB,J), J=1,NM)
          WRITE(2,*) ( (   Z(I,J), I=NQI1,NQIB), J=1,NM)
          WRITE(2,*) ( (   R(I,J), I=NQI1,NQIB), J=1,NM)
          WRITE(2,*) ( (  SM(I,J), I=NQI1,NQIB), J=1,NM)
          WRITE(2,*) ( (  CM(I,J), I=NQI1,NQIB), J=1,NM)
          WRITE(2,*) ( (   Q(I,J), I=NQI1,NQIB), J=1,NM)
          WRITE(2,*) ( ( BLO(I,J), I=NQI1,NQIB), J=1,NM)
          WRITE(2,*) ( ( DEN(I,J), I=NQI1,NQIB), J=1,NM)
          CLOSE(2)
        END IF

          WRITE(6,650)
  650 FORMAT(/////11X'+++++ RELATIVE FLOW FIELD OUTPUT +++++'///
     +  11X'J'8X'M'11X'WM'10X,'WTH'10X'BET'12X'W'12X'U'/)
          DO 40 I=1,NQ
          WRITE(6,671) I
          DO 40 J=1,NM
          WM=CM(I,J)
          U=OMG*R(I,J)
          AM=SM(I,J)-SM(NQI+1,J)
          TEM     =CTH(I,J)-U
          WTH(I,J)=TEM
          BET     =ATAN( TEM/WM ) * DEGGRA
          W       =SQRT(WM*WM+TEM*TEM)
            WRITE(6,672) J, AM, WM, WTH(I,J), BET, W, U
   40     CONTINUE
          WRITE(6,660)
  660 FORMAT(/////11X'+++++ DIMENSIONLESS VALUE ++++'///
     +  11X'J    Q/QMAX'9X'C/UT'8X'CM/UT'7X'CTH/UT'9X'W/UT'7X
     +  'WTH/UT       RCTH/(RT*UT)'/)
          UT=OMG*RAS
          DO 50 I=1,NQ
          WRITE(6,671) I
          DO 50 J=1,NM
          DQ=Q(I,J)/Q(I,NM)
          DCM=CM(I,J)/UT
          DCTH=RCTH(I,J)/R(I,J)/UT
          DC=SQRT(DCM**2-DCTH**2)
          DWTH=WTH(I,J)/UT
          DW=SQRT(DCM**2+DWTH**2)
          DRCTH=RCTH(I,J)/(RAS*UT)
            WRITE(6,672) J, DQ, DC, DCM, DCTH, DW, DWTH, DRCTH
   50     CONTINUE

  671 FORMAT('0     I ='I3)
  672 FORMAT(I12,9(E13.5))
          RETURN
          END
```


PROGRAM SECTIONS

	Name	Bytes	Attributes								
0	$CODE	2830	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
1	$PDATA	767	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT	LONG
2	$LOCAL	92	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	LONG
3	DESIGN	52	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
4	GASCON	20	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
5	MFLOW	7488	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
6	MLIN	3084	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
7	QLIN	1412	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
8	STAN	12	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
9	SCROLL	12	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
10	ROTO	168	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
11	XY	620	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
12	LOAD	88	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
13	NONAXI	1232	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
14	IQS	24	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
15	QS	12	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
	Total Space Allocated	17913									

ENTRY POINTS

Address	Type	Name	References
0-00000000		SOLOUT	4#

VARIABLES

Address	Type	Name	Attributes	References							
**	R*4	A0		88=	89(2)						
**	R*4	ALP		85=	96						
9-00000000	R*4	ALPSC	COMM	27							
**	R*4	AM		127=	132						
3-00000018	R*4	AVIL	COMM	14							
3-00000004	R*4	BE	COMM	14							
**	R*4	BET		130=	132						
9-00000008	R*4	BS	COMM	27							
3-00000000	R*4	BSBE	COMM	14							
**	R*4	C		84=	92	95	96				
3-0000002C	R*4	CONST	COMM	14							
4-00000008	R*4	CP	COMM	16							
**	R*4	CR		82=	96						
**	R*4	CTHAV		83=	84	85	96				
**	R*4	CZ		81=	96						
**	R*4	DC		145=	149						
**	R*4	DCM		143=	145	147	149				
**	R*4	DCTH		144=	145	149					
**	R*4	DEGGRA		40=	70	85	130				
9-00000004	R*4	DHU	COMM	27							
14-00000000	R*4	DMPCM	COMM	35	45						
14-00000004	R*4	DMPG	COMM	35	45						
14-00000008	R*4	DMPL	COMM	35	45						
**	R*4	DQ		142=	149						
**	R*4	DRCTH		148=	149						
**	R*4	DW		147=	149						
**	R*4	DWTH		146=	147	149					
3-00000014	R*4	EFFI	COMM	14							
14-0000000C	R*4	EPSCM	COMM	35	45						
14-00000010	R*4	EPSG	COMM	35	45						
14-00000014	R*4	EPSL	COMM	35	45						
3-00000024	R*4	FLOWC	COMM	14							
3-00000028	R*4	FLOWR	COMM	14							
5-0000002C	R*4	G	COMM	17							
4-00000004	R*4	GASC	COMM	16							
3-00000020	R*4	HEADAD	COMM	14							
2-00000004	I*4	I		67=	68	70	71(5)	78=	79	81(2)	82(2)
				83(2)	84	85	88	89(3)	92	95	96(3)
				109(2)=	110(2)=	111(2)=	112(2)=	113(2)=	114(2)=	115(2)=	122=

```
                                          123       125       126       127       128       129       132       139=
                                          140       142(2)    143       144(2)    146       148
10-000000A0   I*4   IAXI       COMM       28
AP-00000004@  I*4   IDSN3                 4         100
AP-00000008@  I*4   ITRB                  4         41

15-00000000   I*4   ITRL       COMM       36        45
 2-00000008   I*4   J                     69=       70        71(7)     80=       81(2)     82(2)     83(2)     84
                                          85        88        89(3)     92        95        96(4)     107(3)=   108(3)=
                                          109(2)=   110(2)=   111(2)=   112(2)=   113(2)=   114(2)=   115(2)=   124=
                                          125       126       127(2)    128       129       132(2)    141=      142
                                          143       144(2)    146       148       149
AP-0000000C@  I*4   LG                    4         87
15-00000004   I*4   MXITC      COMM       36        45
15-00000008   I*4   MXITG      COMM       36        45

10-00000010   I*4   NB         COMM       28
 6-00000000   I*4   NM         COMM       22        69        80        104       107       108       109       110
                                          111       112       113       114       115       124       141       142
 7-00000038   I*4   NQ         COMM       24        67        78        122       139
 7-00000580   I*4   NQB        COMM       24        104       106
 7-0000057C   I*4   NQI        COMM       24        105       106       127

     **       I*4   NQI1                  105=      107(2)    109       110       111       112       113       114
                                          115
 2-0000000C   I*4   NQIB                  106=      108(2)    109       110       111       112       113       114
                                          115
 5-00000030   R*4   OMG        COMM       17        126       138
 4-0000000C   R*4   POSABS     COMM       16
     **       R*4   PHI                   70=       71

 2-00000000   R*4   PI                    38D       40
 3-00000030   R*4   POW        COMM       14
11-00000000   R*4   RAS        COMM       30        74        138       148
     **       R*4   RMACH                 95=       96
 3-0000000C   R*4   RPM        COMM       14

 3-00000008   R*4   RT         COMM       14
10-0000009C   R*4   RUMAXI     COMM       28
 4-00000000   R*4   SHR        COMM       16        88        89
 8-00000008   R*4   STANDE     COMM       26
 8-00000000   R*4   STANP      COMM       26

 8-00000004   R*4   STANT      COMM       26
 4-00000010   R*4   TOSABS     COMM       16
     **       R*4   TEM                   128=      129       130       131(2)
AP-00000010@  CHAR  TODAY                 4         10        102
10-000000A4   R*4   TTE        COMM       28

     **       R*4   U                     126=      128       132
 3-00000010   R*4   UT         COMM       14        138=      143       144       146       148
 3-0000001C   R*4   VELR       COMM       14
     **       R*4   W                     131=      132
     **       R*4   WM                    125=      130       131(2)    132

10-00000098   R*4   ZHAXI      COMM       28
```

ARRAYS

```
   Address    Type  Name       Attributes       Bytes  Dimensions      References

 5-000009D4   R*4   A          COMM              616   (14, 11)         17        89=       92=       95
10-0000006C   R*4   AXIM       COMM               44   (11)             28
 5-00001138   R*4   BLO        COMM              616   (14, 11)         17        114
 5-00000034   R*4   CM         COMM              616   (14, 11)         17        81        82        84        85
                                                                        89        96        112       125       143
 5-0000029C   R*4   CTH        COMM              616   (14, 11)         17        89        128

 6-000004D4   R*4   CUR        COMM              616   (14, 11)         22        71
 5-00000C3C   R*4   DEN        COMM              616   (14, 11)         17        96        115
 5-000013A0   R*4   DMBLO      COMM              616   (14, 11)         17
 5-00001870   R*4   DMCM       COMM              616   (14, 11)         17
 5-0000110C   R*4   DMF        COMM               44   (11)             17        71

 5-00001608   R*4   DMRCTH     COMM              616   (14, 11)         17
 5-00000EA4   R*4   ENT        COMM              616   (14, 11)         17
 5-00000000   R*4   GS         COMM               44   (11)             17
 5-0000076C   R*4   H          COMM              616   (14, 11)         17        88
12-0000002C   I*4   IPCON      COMM               44   (11)             31
```

10-00000014	I*4	IPRF	COMM	44	(11)	28				
12-00000000	R*4	PCON	COMM	44	(11)	31				
6-0000026C	R*4	PHIR	COMM	616	(14, 11)	22	70	81	82	
7-0000003C	R*4	Q	COMM	616	(14, 11)	24	71	113	142(2)	
6-00000004	R*4	R	COMM	616	(14, 11)	22	71	83	107	108
						110	126	144		
5-00000504	R*4	RCTH	COMM	616	(14, 11)	17	83	107	108	144
						148				
7-000002A4	R*4	RUMR	COMM	616	(14, 11)	24				
7-00000000	R*4	RUMRQ	COMM	56	(14)	24				
6-0000073C	R*4	SM	COMM	616	(14, 11)	22	71	111	127(2)	
10-00000040	R*4	TMPC	COMM	44	(11)	28				
13-00000000	R*4	TNA	COMM	616	(14, 11)	32				
13-00000268	R*4	WCTHD	COMM	616	(14, 11)	32				
5-00001AD8	R*4	WTH	COMM	616	(14, 11)	17	129=	132	146	
11-00000004	R*4	XM	COMM	616	(14, 11)	30	96			
6-000009A4	R*4	Z	COMM	616	(14, 11)	22	71	109		
10-00000000	R*4	ZLE	COMM	8	(2)	28				
7-0000050C	R*4	ZQH	COMM	56	(14)	24				
7-00000544	R*4	ZQS	COMM	56	(14)	24				
10-00000008	R*4	ZTE	COMM	8	(2)	23				

PARAMETER CONSTANTS

Type	Name	References							
I*4	NMMX	12#	17(14)	22(5)	24(2)	28(3)	30	31(2)	32(2)
I*4	NQMX	12#	17(12)	22(5)	24(5)	30	32(2)		

LABELS

Address	Label	References				
**	10	67	69	73#		
**	20	78	80	98#		
**	40	122	124	133#		
**	50	139	141	150#		
1-00000007	610'	41	42#			
1-00000109	611'	64	65#			
1-00000061	615'	45	47#			
1-0000016D	630'	74	75#			
1-000001E8	640'	102	103#			
1-00000212	650'	119	120#			
1-0000026E	660'	134	135#			
1-000002E6	671'	68	79	123	140	152#
1-000002F4	672'	71	96	132	149	153#

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References					
	FOR$CLOSE	116					
	FOR$OPEN	101					
R*4	MTH$ATAN	85	130				
R*4	MTH$COS	81					
R*4	MTH$SIN	82					
R*4	MTH$SQRT	84	88	89	131	145	147

```
+-----------------------------------------------+
|            KEY TO REFERENCE FLAGS             |
|   =   - Value Modified                        |
|   #   - Defining Reference                    |
|   A   - Actual Argument, possibly modified    |
|   D   - Data Initialization                   |
|  (n)  - Number of occurrences on line         |
+-----------------------------------------------+
```

```
      SUBROUTINE SPLIN3( N, X, Y, NARG, DOMAIN, FUNC, DERIV, SUMS, SEND,
     *                        EPS)

C (3037-SPLIN3) CUBIC SPLINE INTERPOLATION (AND INTEGRATION)
C V.3 - INTEGRATION TO EVERY PT.
C SEE ALSO V.1 - BASIC SPLINE SUBROUTINE
C SEE ALSO V.2 - INCLUDING SINGLE ARGUMENT INTERPOLATION &
C                       EXTRAPOLATION
C ADAPTED FROM (8005-SPLIN1), 19MAY86; TOUCHED 06JUN86
C REFERENCED BY NUFLOW
C REFERENCES MTH$SQRT
C INPUT EPS, DOMAIN(*), N, NARG, SEND, X(*), AND Y(*)
C OUTPUT DERIV(*), FUNC(*), AND SUMS(*)...
C TO BE COMPILED BY VAX FORTRAN V4.0-2

C DERIV OUTPUT VECTOR (SIZE NARG) CONTAINING DERIVATIVE VALUES FOR
C             DOMAIN(*)
C DOMAIN  INPUT VECTOR (SIZE NARG, IN ASCENDING ORDER) CONTAINING DOMAIN
C             VALUES FOR WHICH THE DERIVATIVE OR FUCTIONAL VALUE IS TO BE
C             COMPUTED
C EPS   ERROR TOLERANCE IN ITERATIVE SOL. OF SIMUL. EQS.
C FUNC  OUTPUT VECTOR (SIZE NARG) CONTAINING INTERPOLATED FUNCTION
C             VALUES FOR DOMAIN(*)
C N           # OF DATA PTS.
C NARG  # OF ARGUMENTS FOR WHICH FUNC(*) &/OR DERIV(*) ARE REQ'D.;
C             POSITIVE IF THE INTEGRAL IS NEEDED; NEG. IF THE INTEGRAL IS NOT
C             NEEDED; 0 IF ONLY THE INTEGRAL IS NEEDED (NEITHER FUNC NOR DERIV
C             IS NEEDED)
C NQ    # OF Q-LINES USED IN THE SCM METHOD
C OMEGA RELAXATION FACTOR FOR SUCCESSIVE OVER-RELAXATION
C SEND  A FACTOR TO BE APPLIED TO S(2) & S(N-1) TO OBTAIN S(1) & S(N),
C             RESPECTIVELY, S BEING CURVATURE; NORMALLY 0, .5, OR 1
C SUMS  INTEGRALS
C X           VECTOR (SIZE N, IN ASCENDING ORDER) CONTAINING DOMAIN VALUES OF
C             THE DATA PTS.
C Y           VECTOR (SIZE N) CONTAINING RANGE VALUES OF THE DATA PTS.

      PARAMETER ( MXN=103)

      DIMENSION DERIV(1), DOMAIN(1), DX(MXN), DYDX(MXN),
     *  FUNC(1), G(MXN), S(MXN), SUMS(1), WORK(MXN), X(N), Y(N)

      DATA     MNITER/3/, MXITER/50/

      IF (N.GT.MXN .OR. N.LT.3) THEN
        PRINT *, 'N EXCEEDED DIM or < 3...', N, MXN
        STOP 'SPLIN3.1'
        END IF

      OMEGA    =8.-4.*SQRT(3.)

      NM1      =N-1
      DO       I=1,NM1
        DX(I)   =X(I+1)-X(I)
        DYDX(I) =( Y(I+1)-Y(I) )/DX(I)
        END DO

      DO       I=2,NM1
        DX2     =X(I+1)-X(I-1)
        WORK(I) =DX(I-1)*.5/DX2
        DYDX2H  =( DYDX(I)-DYDX(I-1) )/DX2
        S(I)    =DYDX2H+DYDX2H
        G(I)    =S(I)+DYDX2H
        END DO
      S(1)     =S(2)*SEND
      S(N)     =S(N-1)*SEND

      ITER     =0
   30 ETA      =0.
      ITER     =ITER+1
      DO       I=2,NM1
        TEM     =( G(I) - WORK(I)*S(I-1) - ( .5-WORK(I) )*S(I+1)
     *                 - S(I) ) * OMEGA
        ETA     =AMAX1( ABS(TEM), ETA)
        S(I)    =S(I)+TEM
        END DO

```

```
      IF (ITER.LT.MNITER) GOTO 30
      IF (ETA.GT.EPS) THEN
        IF (ITER.LT.MXITER) GOTO 30
        DO   I=1,N
          WRITE(6,*) I, X(I), Y(I), G(I), S(I), WORK(I)
          END DO
        PRINT *, 'NOT CONVERGED... ', MXITER, ETA, EPS
        STOP 'SPLIN3.85'
        END IF

      IF (NARG.EQ.0) GOTO 210

      DO       I=1,NM1
        G(I)     =( S(I+1)-S(I) )/DX(I)
        END DO

C 100 CALC. FUNCTION VALUES AND DERIVATIVES
      I        =2
      DO       J=1,ABS(NARG)
        DOM      =DOMAIN(J)
        IF ( X(1).GT.DOM .OR. X(N).LT.DOM) THEN
          PRINT *, 'ARGUMENT OUT OF RANGE... ', J, N, DOM, X(1), X(N)
          STOP 'SPLIN3.115'
          END IF
        DO WHILE ( X(I).LT.DOM )
          I      =I+1
          END DO
        IM       =I-1
        H        =DOM-X(IM)
        T        =DOM-X(I)
        HT       =H*T
        DSQS     =( G(IM)*H+S(IM)+S(IM)+S(I) )/6.
        FUNC(J) =DYDX(IM)*H + HT*DSQS + Y(IM)
        DERIV(J)=(H+T)*DSQS + G(IM)*HT/6. + DYDX(IM)
        END DO

C 200 INTEGRATE FROM X(1) TO X(N)
      IF (NARG.LT.0)      RETURN
  210 SUMS(1)    =0.
      DO       J=1,NM1
        H        =DX(J)
        SUMS(J+1)
     +  =( (Y(J)+Y(J+1))*.5 - ( S(J) + S(J+1) )*H*H/24. )*H + SUMS(J)
        END DO

      RETURN
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1587	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	97	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	2304	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	3988	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SPLIN3	1#

VARIABLES

Address	Type	Name	Attributes	References					
**	R*4	DOM		97=	98(2)	99	102	106	107
**	R*4	DSQS		109=	110	111			
**	R*4	DX2		59=	60	61			
**	R*4	DYDX2H		61=	62(2)	63			
AP-00000028@	R*4	EPS		1	79	84			
**	R*4	ETA		69=	74(2)=	79	84		

Address	Type	Name	References							
**	R*4	H	106=	108	109	110	111	118=	119(3)	
**	R*4	HT	108=	110	111					
**	I*4	I	53=	54(3)	55(4)	58=	59(2)	60(2)	61(2)	62
			63(2)	71=	72(6)	75(2)	81=	82(6)	90=	91(4)
			95=	102	103(2)=	105	107	109		
**	I*4	IM	105=	106	109(3)	110(2)	111(2)			
**	I*4	ITER	68=	70(2)=	78	80				
**	I*4	J	96=	97	99	110	111	117=	118	119(6)
2-0000080C	I*4	MNITER	43D	78						
2-00000810	I*4	MXITER	43D	80	84					
AP-00000004@	I*4	N	1	40(2)	45(2)	46	52	66(2)	81	98
			99(2)							
AP-00000010@	I*4	NARG	1	88	96	115				
2-00000814	I*4	NM1	52=	53	58	71	90	117		
**	R*4	OMEGA	50=	72						
AP-00000024@	R*4	SEND	1	65	66					
**	R*4	T	107=	108	111					
**	R*4	TEM	72=	74	75					

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
AP-0000001C@	R*4	DERIV		4	(1)	1	40	111=		
AP-00000014@	R*4	DOMAIN		4	(1)	1	40	97		
2-00000000	R*4	DX		412	(103)	40	54=	55	60	91
						118				
2-0000019C	R*4	DYDX		412	(103)	40	55=	61(2)	110	111
AP-00000018@	R*4	FUNC		4	(1)	1	40	110=		
2-00000338	R*4	G		412	(103)	40	63=	72	82	91=
						109	111			
2-000004D4	R*4	S		412	(103)	40	62=	63	65(2)=	66(2)=
						72(3)	75(2)=	82	91(2)	109(3)
						119(2)				
AP-00000020@	R*4	SUMS		4	(1)	1	40	116=	119(2)=	
2-00000670	R*4	WORK		412	(103)	40	60=	72(2)	82	
AP-00000008@	R*4	X		**	(*)	1	40	54(2)	59(2)	82
						98(2)	99(2)	102	106	107
AP-0000000C@	R*4	Y		**	(*)	1	40	55(2)	82	110
						119(2)				

PARAMETER CONSTANTS

Type	Name	References			
I*4	MXN	38#	40(5)	45	46

LABELS

Address	Label	References		
0-00000230	30	69#	78	80
0-000005B0	210	88	116#	

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
R*4	MTH$SQRT	50

```
+------------------------------------------+
|          KEY TO REFERENCE FLAGS          |
|    =   - Value Modified                  |
|    #   - Defining Reference              |
|    A   - Actual Argument, possibly modified |
|    D   - Data Initialization             |
|   (n)  - Number of occurrences on line   |
+------------------------------------------+
```

```
C
C+++++++++   SUBROUTINE SPLINE    +++++++++
C
          SUBROUTINE SPLINE(X,Y,N,L,Y1,Y2,WK,NW,ICON)

C FIND Y2=dY/dX by CUBIC SPLINE METHOD
C INPUT N, NW, X(N), Y(N); OUTPUT L(N),WK(NW), Y1(N), Y2(N)
C DEBUGGED 29JUN86 for v.2.5

          REAL X(N),Y(N),Y1(N),Y2(N),L(N),WK(NW)
          NF=N-2
          NA=NF+(N-3)
          NB=NA+(N-2)
          NC=NB+(N-3)
          IF(NW.GE.NC) GO TO 11
          ICON=-1
          GOTO 9999
11        CONTINUE
          DO 10 I=1,N-1
10        L(I)=X(I+1)-X(I)
          DO 20 I=1,N-2
          I1=I+1
20        WK(I)=(Y(I1+1)-Y(I1))/L(I1)-(Y(I+1)-Y(I))/L(I)
          DO 30 I=1,N-4
          WK(NF+I)=L(I+1)/6.
          WK(NA+I+1)=(L(I+1)+L(I+2))/3.
30        WK(NB+I+1)=L(I+2)/6.
          WK(NF+N-3)=(L(N-2)**2-L(N-1)**2)/(6.*L(N-2))
          WK(NA+1)=(L(1)**2+3*L(1)*L(2)+2*L(2)**2)/(6.*L(2))
          WK(NA+N-2)=(L(N-1)**2+3*L(N-1)*L(N-2)+2*L(N-2)**2)/(6.*L(N-2))
          WK(NB+1)=(L(2)**2-L(1)**2)/(6.*L(2))
          NTR=N-2
          EPS=WK(NA+2)*1.E-5
          IF(NTR.LT.3.OR.EPS.LE.0.0) GO TO 999
          NN=NTR-1
          DO 1 I=1,NN
          IF(ABS(WK(NA+I)).LT.ABS(WK(NF+I))) GO TO 2
          IF(ABS(WK(NA+I)).LT.EPS) GO TO 888
          WK(NB+I)=WK(NB+I)/WK(NA+I)
          WK(I)=WK(I)/WK(NA+I)
          WK(NA+I+1)=WK(NA+I+1)-WK(NF+I)*WK(NB+I)
          WK(I+1)=WK(I+1)-WK(NF+I)*WK(I)
          WK(NF+I)=0.0
          GO TO 1
2         IF(ABS(WK(NF+I)).LT.EPS) GO TO 888
          W=WK(NA+I+1)/WK(NF+I)
          WK(NA+I+1)=WK(NB+I)-WK(NA+I)*W
          WK(NB+I)=W
          W=WK(I+1)/WK(NF+I)
          WK(I+1)=WK(I)-WK(NA+I)*W
          WK(I)=W
          IF(I.GE.NTR-1) GO TO 1
          WK(NF+I)=WK(NB+I+1)/WK(NF+I)
          WK(NB+I+1)=-WK(NA+I)*WK(NF+I)
1         CONTINUE
          IF(ABS(WK(NA+I)).LT.EPS) GO TO 888
          WK(NTR)=WK(NTR)/WK(NA+NTR)
          WK(NTR-1)=WK(NTR-1)-WK(NB+NTR-1)*WK(NTR)
          DO 3 I=2,NN
          II=NTR-I
3         WK(II)=WK(II)-WK(NB+II)*WK(II+1)-WK(NF+II)*WK(II+2)
          GO TO 5000
888       ICON=I
          GOTO 9999
999       ICON=30000
          GOTO 9999
5000      CONTINUE
          DO 40 I=2,N-1
40        Y2(I)=WK(I-1)
          Y2(1)=(1.+L(1)/L(2))*Y2(2)-L(1)/L(2)*Y2(3)
          Y2(N)=(1.+L(N-1)/L(N-2))*Y2(N-1)-L(N-1)/L(N-2)*Y2(N-2)
          DO 50 I=1,N-1
          F1=(Y(I+1)-Y(I))/L(I)
50        Y1(I)=F1-(Y2(I+1)-Y2(I))*L(I)/6.-Y2(I)*L(I)/2.
          F1=(Y(N)-Y(N-1))/L(N-1)
          Y1(N)=Y2(N)*L(N-1)/2.+F1-(Y2(N)-Y2(N-1))*L(N-1)/6.
          ICON=0
9999      RETURN
```

0079 END

PROGRAM SECTIONS

Name	Bytes	Attributes
0 $CODE	2459	PIC CON REL LCL SHR EXE RD NOWRT LONG
2 $LOCAL	284	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
Total Space Allocated	2743	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SPLINE	4#

VARIABLES

Address	Type	Name	Attributes	References							
**	R*4	EPS		33=	34	38	45	56			
**	R*4	F1		73=	74	75=	76				
2-0000000C	I*4	I		19=	20(3)	21=	22	23(4)	24=	25(2)	26(3)
				27(2)	29	36=	37(2)	38	39(3)	40(3)	41(4)
				42(4)	43	45	46(2)	47(3)	48	49(2)	50(3)
				51	52	53(3)	54(3)	56	59=	60	63
				68=	69(2)	72=	73(3)	74(6)			
**	I*4	I1		22=	23(3)						
AP-00000024@	I*4	ICON		4	16=	63=	65=	77=			
**	I*4	II		60=	61(6)						
AP-0000000C@	I*4	N		4	10(5)	11	12	13	14	19	21
				24	28(4)	30(6)	32	68	71(7)	72	75(3)
				76(6)							
2-00000004	I*4	NA		12=	13	26	29	30	33	37	38
				39	40	41(2)	46	47(2)	50	54	56
				57							
2-00000008	I*4	NB		13=	14	27	31	39(2)	41	47	48
				53	54	58	61				
**	I*4	NC		14=	15						
2-00000000	I*4	NF		11=	12	25	28	37	41	42	43
				45	46	49	53(2)	54	61		
2-00000018	I*4	NN		35=	36	59					
2-00000014	I*4	NTR		32=	34	35	52	57(3)	58(4)	60	
AP-00000020@	I*4	NW		4	10	15					
**	R*4	W		46=	47	48	49=	50	51		

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
AP-00000010@	R*4	L		**	(*)	4	10	20=	23(2)	25
						26(2)	27	28(3)	29(5)	30(5)
						31(3)	70(4)	71(4)	73	74(2)
						75	76(2)			
AP-0000001C@	R*4	WK		**	(*)	4	10	23=	25=	26=
						27=	28=	29=	30=	31=
						33	37(2)	38	39(3)=	40(3)=
						41(4)=	42(4)=	43=	45	46(2)
						47(3)=	48=	49(2)	50(3)=	51=
						53(3)=	54(3)=	56	57(3)=	58(4)=
						61(6)=	69			
AP-00000004@	R*4	X		**	(*)	4	10	20(2)		
AP-00000008@	R*4	Y		**	(*)	4	10	23(4)	73(2)	75(2)
AP-00000014@	R*4	Y1		**	(*)	4	10	74=	76=	
AP-00000018@	R*4	Y2		**	(*)	4	10	69=	70(3)=	71(3)=
						74(3)	76(3)			

LABELS

Address	Label	References			
0-00000676	1	36	44	52	55#
0-00000540	2	37	45#		
**	3	59	61#		
**	10	19	20#		
0-000000BC	11	15	18#		
**	20	21	23#		
**	30	24	27#		
**	40	68	69#		
**	50	72	74#		
0-0000098C	888	38	45	56	63#
0-00000994	999	34	65#		
**	5000	62	67#		
0-0000099A	9999	17	64	66	78#

B.2 PROGRAM RIS

```
      PROGRAM   RIS

C (3037-RIS) Calc. cascade influence functions R & I (Millor, 1959, p.370)
C            & store
C Adapted from 9845-3037-RI, 18NOV85... copied from 9845-3725-R&I2...
C       adapted from 9845-3725-R&I... coded by W. Chiang, 06JUL83...
C       revised 27MAY88 for format of file 1 RI.DAT
C Input RISI.DAT, output RIS.DAT & RISO.DAT
C References no subroutines
C Fct. of XS=XMXC*CS=(x0-x)/s & AMDA=Lambda
C Using symbols AI for I and AMDA for Lambda

C AI    I function
C AII   I function
C AMDA  Lambda, in deg.
C AMDAA Starting Lambda, in deg.
C AMDAI Increment of Lambda, in deg.
C AMDAK AMDA(KA)
C AMDAZ Ending Lambda, in deg.
C CO    cos(Lambda)
C EPS   Converging criterion
C EQUAL Equal sign
C K           A dummy counter
C KA    Dummy counter, for AMDA
C KAMAX Max. allowable KA
C KAZ   Final KA
C KX    Dummy counter, for XS
C KXMAX Max. allowable KX
C KXZ   Final KX
C LENGTH  # of characters for the length of the horizontal lines in table
C N           A dummy variable, 1 to NMAX
C NMAX  Max. # of terms (either pos. or neg. side) in the series
C NMAXMX  Max. N used
C Q           A temporary variable
C Q1    A temporary variable
C Q2    A temporary variable
C Q3    A temporary variable
C Q9    A temporary variable
C R           R function
C RANDI A string contains R & I
C RPI   1/(PI)
C RR    R function
C SI    sin(Lambda)
C UL    Underline sign
C XS    (x0-x)/s = (x0-x)/c * c/s
C XSA   Starting XS
C XSI   Increment of XS
C XSK   XS(KX)
C XSZ   Ending XS

C   0 *** Dimensions ********************
      CHARACTER*1       EQUAL, RANDI*15, TODAY*9, UL, VS*3

      DIMENSION AI(51,7), AMDA(7), R(51,7), XS(51)

      DATA      VS/'1.0'/

      KAMAX     =7
      KXMAX     =51

C  10 *** File ********************
      OPEN ( 1, FILE='RI.DAT',   STATUS='NEW')
      OPEN ( 5, FILE='RISI.DAT', STATUS='OLD')
      OPEN ( 6, FILE='RISO.DAT', STATUS='NEW')

C  50 *** Data ********************
C  60 *** Constant ***************
      DATA      EQUAL, RANDI, UL/ '=', 'R      I        ', '_'/
      RPI       =1./3.14159265359
C  70 *** Input ***************

      CALL      DATE(TODAY)

      READ (5,*)
      WRITE(6,75) VS, TODAY

   75 FORMAT(' (3037-RISO.DAT) OUTPUT FROM RIS.FOR... VERSION 'A3,
```

```
     +             T61,A9//)
      READ (5,*) AMDAA, AMDAZ, AMDAI, XSA, XSZ, XSI
      WRITE(6,*) AMDAA, AMDAZ, AMDAI, XSA, XSZ, XSI
      READ (5,*) NMAX, EPS
      WRITE(6,*) NMAX, EPS
      WRITE(6,79) EPS
   79 FORMAT(/' CONVERGENCE CRITERION ='1PE8.1////)
C  80 *** Constant ***************
      KAZ          =(AMDAZ-AMDAA)/AMDAI+1
      KXZ          =(XSZ-XSA)/XSI+1
      IF (KAZ.GT.KAMAX .OR. KXZ.GT.KXMAX) THEN
        TYPE *, 'KAZ.GT.KAMAX .OR. KXZ.GT.KXMAX...', KAZ, KAMAX, KXZ,
     *          KXMAX
        STOP 'IRS.104'
        END IF
      IF (KAZ.GT.8) TYPE *, ' MAY NEED CHANGE FORMAT FOR LISTING...'
      LENGTH       =5+15*KAZ
      DO      KA=1,KAZ
        AMDA(KA)      =(KA-1)*AMDAI+AMDAA
        END DO
      DO      KX=1,KXZ
        XS(KX)  =(KX-1)*XSI+XSA
        END DO
C  90 *** Initial ***************
      NMAXMX       =0
C 100 *** Loop thru Lambda *******************
      DO  400 KA=1,KAZ
        AMDAK    =AMDA(KA)
        TYPE *, '***** LAMBDA =', AMDAK
        IF ( AMDAK.EQ.0. ) GOTO 350
C 110   *** Loop thru XS (Lambda .NE. 0) ***************
        SI       =SIND( AMDAK )
        CO       =COSD( AMDAK )
        DO       300   KX=1,KXZ
          XSK    =XS(KX)
          IF (AMDAK.EQ.90. .AND. XSK.GE.1.) THEN
            DO         K=KX,KXZ
              AI(K,KA)  =0.
              R(K,KA)    =-9.999
              END DO
            GOTO 400
            END IF
          AII    =0.
          RR     =0.
C 150     *** Summation over -N to N except 0 *****
          DO     200  N=1, NMAX
            Q1   =XSK-SI*N
            Q2   =XSK+SI*N
            Q          =CO*CO*N*N
            Q3   =1/(Q1*Q1+Q)
            Q          =1/(Q2*Q2+Q)
            Q9   =CO*N*(Q3-Q)
            Q          =Q1*Q3+Q2*Q
                IF ( ABS(Q).LT.EPS .AND. ABS(Q9).LT.EPS ) GOTO 210
            AII =AII+Q9
  200       RR  =RR+Q
          TYPE *, 'N exceeded...', NMAX, AMDAK, XSK
          STOP 'RIS.202'
  210     AI(KX,KA)     =(AII+Q)*RPI
                R(KX,KA)          =(RR+Q9)*RPI
CTYPE       TYPE *, XSK, R(KX,KA), AI(KX,KA)
  300     NMAXMX      =MAX(NMAXMX,N)
          GOTO 400
C 340   *** Loop thru XS (for Lambda = 0) **********
  350   DO  390   KX=1,KXZ
C 360     *** Summation over -N to N except 0 *****
          RR     =0.
          XSK    =XS(KX)
          DO     370  N=1, NMAX
            Q    =(XSK+XSK) / (XSK*XSK+N*N)
                IF ( ABS(Q).LT.EPS ) GOTO 380
  370         RR=RR+Q
          TYPE *, ' N exceeded...', NMAX, 0, XSK
          STOP 'RI.372'
  380     AI(KX,KA)      =0.
                R(KX,KA)          =(RR+Q9)*RPI
          TYPE *, XSK, R(KX,KA)
```

```
 390       NMAXMX=MAX(NMAXMX,N)
 400   CONTINUE

C 500 *** List data ********************
      WRITE(6,5520) ( EQUAL, K=1,LENGTH)
      WRITE(6,530) ( AMDA(KA), KA=1,KAZ)
  530 FORMAT(/' Lambda'F10.0,7F15.0)
      WRITE(6,5520) ( UL, K=1,LENGTH)
      WRITE(6,540) ( RANDI, KA=1,KAZ)
  540 FORMAT(/'   X/S     '9A15)
      WRITE(6,5520) ( UL, K=1,LENGTH)
      WRITE(6,*)
      DO      KX=1,KXZ
        WRITE(6,5550) XS(KX), ( R(KX,KA), AI(KX,KA), KA=1,KAZ)
        IF ( MOD(KX,5).EQ.1 ) THEN
          WRITE(6,*)
          END IF
        END DO
      WRITE(6,5520) ( EQUAL, K=1,LENGTH)
      WRITE(6,5003)
      WRITE(6,*) 'Maximum number of positive or negative terms used =',
     1                   NMAXMX

C 600 *** Store data ********************
      WRITE(1,*) '(3037-RI.DAT) GENERATED FROM RIS.FOR FOR MELLOR.FOR'
      WRITE(1,*) EPS, AMDAA, AMDAZ, AMDAI, XSA, XSZ, XSI
      WRITE(1,*) ( (  R(KX,KA), KA=1,KAZ), KX=1,KXZ),
     *           ( ( AI(KX,KA), KA=1,KAZ), KX=1,KXZ), 999

C 900 ********************
      STOP

 5003 FORMAT(///)
 5008 FORMAT(A20)
 5520 FORMAT(1H 125A1)
 5550 FORMAT(F6.2,8(F8.3,F7.3))
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes								
0	$CODE	2325	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
1	$PDATA	428	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT	LONG
2	$LOCAL	3496	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	LONG
	Total Space Allocated	6249									

ENTRY POINTS

Address	Type	Name	References
0-00000000		RIS	1#

VARIABLES

Address	Type	Name	Attributes	References							
**	R*4	AII		122=	134(2)=	138					
2-00000C30	R*4	AMDAA		79=	80	86	96	183			
2-00000C38	R*4	AMDAI		79=	80	86	96	183			
2-00000C70	R*4	AMDAK		107=	108	109	111	112	115	136	
2-00000C34	R*4	AMDAZ		79=	80	86	183				
2-00000C78	R*4	CO		112=	128(2)	131					
2-00000C4C	R*4	EPS		81=	82	83	133(2)	151	183		
2-00000C10	CHAR	EQUAL		52	68D	162	176				
**	I*4	K		116=	117	118	162=	165=	168=	176=	
2-00000C5C	I*4	KA		95=	96(2)	106=	107	117	118	138	139
				140(2)	155	156	157	163(2)=	166=	171(3)=	184(4)=
**	I*4	KAMAX		58=	88	89					
2-00000C50	I*4	KAZ		86=	88	89	93	94	95	106	163
				166	171	184(2)					
**	I*4	KX		98=	99(2)	113=	114	116	138	139	140(2)

```
                                  145=     148      155      156      157      170=     171(3)   172
                                  184(4)=
      **         I*4  KXMAX        59=      88       89
2-00000C54       I*4  KXZ          87=      88       89       98      113      116      145      170
                                  184(2)

2-00000C58       I*4  LENGTH       94=     162      165      168      176
      **         I*4  N           125=     126      127      128(2)   131      141      149=     150(2)
                                  158
2-00000C48       I*4  NMAX         81=      82      125      136      149      153
2-00000C68       I*4  NMAXMX      103=     141(2)=  158(2)=  178
      **         R*4  Q           128=     129      130(2)=  131      132(2)=  133      135      138
                                  150=     151      152

      **         R*4  Q1          126=     129(2)   132
      **         R*4  Q2          127=     130(2)   132
      **         R*4  Q3          129=     131      132
2-00000C8C       R*4  Q9          131=     133      134      139      156
2-00000C11       CHAR RANDI        52       680     166

      **         R*4  RPI          69=     138      139      156
      **         R*4  RR          123=     135(2)=  139      147=     152(2)=  156
2-00000C74       R*4  SI          111=     126      127
2-00000C20       CHAR TODAY        52       72A      75
2-00000C29       CHAR UL           52       680     165      168

2-00000C2A       CHAR VS           52       560      75
2-00000C3C       R*4  XSA          79=      80       87       99      183
2-00000C44       R*4  XSI          79=      80       87       99      183
2-00000C80       R*4  XSK         114=     115      126      127      136      140      148=     150(4)
                                  153      157
2-00000C40       R*4  XSZ          79=      80       87      183


ARRAYS

   Address   Type  Name        Attributes       Bytes  Dimensions      References

2-00000000   R*4   AI                            1428  (51, 7)          54      117=     138=     140      155=
                                                                       171      184
2-00000594   R*4   AMDA                            28  (7)              54       96=     107      163
2-000005B0   R*4   R                             1428  (51, 7)          54      118=     139=     140      156=
                                                                       157      171      184
2-00000B44   R*4   XS                             204  (51)             54       99=     114      148      171

LABELS

   Address   Label                        References

1-0000010B   75'                           75       77#
1-00000146   79'                           83       84#
     **      200                          125      135#
0-000003D0   210                          133      138#
     **      300                          113      141#

0-0000048C   350                          109      145#
     **      370                          149      152#
0-0000051C   380                          151      155#
     **      390                          145      158#
0-000005E0   400                          106      120      142      159#

1-00000168   530'                         163      164#
1-0000017E   540'                         166      167#
1-00000191   5003'                        177      190#
     **      5008'                        191#
1-00000195   5520'                        162      165      168      176      1,2#

1-0000019D   5550'                        171      193#


FUNCTIONS AND SUBROUTINES REFERENCED

   Type  Name                             References

         FOR$DATE_T_DS                     72
         FOR$OPEN                          62       63       64
   R*4   MTH$COSD                         112
   R*4   MTH$SIND                         111
```

B.3 PROGRAM MELLOR

Provided in this section is a listing of the computer program MELLOR and some of its subroutines which include

DGNK
FFCP
GNK
INPUT
LOOP
and RESULT.

The subroutines FILONC, SIMUL1, and SPLIN1 are sharable by programs Mellor and DSN3 and are listed together with program DSN3 in Section B.4.

B.3 PROGRAM MELLOR

Provided in this section is a listing of the computer program MELLOR and some of its subroutines which include

DGNK
FFCP
GNK
INPUT
LOOP
and RESULT.

The subroutines FILONC, SIMUL1, and SPLIN1 are sharable by programs Mellor and DSN3 and are listed together with program DSN3 in Section B.4.

```
      PROGRAM   MELLOR
C (3037-MELLOR) CALC. LIFT COEF. ETC. USING CASCADE THEORY (MILLOR, 1959)
C REQ. SUBS (D) DGNK, (.) FFCP, (.) FILONC, (G) GNK, (I) INPUT, (L) LOOP,
C            (R) RESULT, (.) SIMUL1, & (.) SPLIN1
C REQ. INPUT DATA FILES (5) MELI, (7) ICL (IF MH=2), (9) RI (GENERATED
C       FROM (3725[9845]-R&I)), & (11) BLADE
C REFERENCES DGNK, INPUT, LOOP, READRI, & SRI, & FOR$DATE_T_DS & FOR$OPEN
C ADAPTED FROM (3037-CASCAD[HP9845])
C CODED BY W. CHIANG, 14JULY83... REVISED 11NOV85
C TRANSFERED TO VAX BY S. HSU, 10OCT86
C TO BE COMPILED BY VAX-11 FORTRAN, V.4.0
C REVISED 24OCT86, CHIANG; TOUCHED 16JUN88

COMC (3037-MELLOR-MELCOM)                                              01APR87
COM
COM      PARAMETER     ( KAMX=8, KXMX=76, NALMX=13, NAMX=19, NBMX=4, NCMX=9,
COM     +                NFRMX=15, NSMX=5, NSECMX=500, NYMAX=100, VS='1.0')
COM      PARAMETER     ( NYSMX=MAX( NSECMX*2, MAX( NYMAX, NSECMX+1+2)),
COM     +                  NYSMX2=MAX( NYMAX+1, NSECMX*2) + 2)
COM
COM      COMMON/M  IL / TDATE
COM      COMMON/ DG   / DG(NFRMX,NFRMX,NSECMX),
COM     +                       DH(NFRMX,NFRMX,NSECMX),
COM     +                       DB(NFRMX,NSECMX), DT(NFRMX,NSECMX)
COM      COMMON/ DGI  / DX, NSEC
COM      COMMON/ DGILR/ LISTO, NFR
COM      COMMON/ DG L / C(NFRMX)
COM      COMMON/ D I  / FCP(NYSMX2), FTP(NYSMX),             NSEC2
COM      COMMON/ D IL / PI
COM      COMMON/  GI  / AI(KXMX,KAMX), R(KXMX,KAMX),         XSA, XSI
COM      COMMON/  GILR/ CSA
COM      COMMON/  G LR/ G(NFRMX,NFRMX), H(NFRMX,NFRMX),
COM     +                       B(NFRMX), T(NFRMX),           AMDAD, CS
COM      COMMON/   IL / ICL(NAMX,NCMX,NSMX,NBMX),
COM     +                       BETA1(NBMX), CB1(NSMX), V(NFRMX),
COM     +                       AMDA, AMDZ, AMDI, CA, CC, CSI, CSZ, SALPHA,
COM     +                       SEND5, TOC,           ISCL, MH, NA, NB, NC, NS
COM      COMMON/   ILR/ ALPHAM, CALPHA, CB, PI2
COM      COMMON/   I R/ IRES,          SA2PI, TOC2PI
COM      COMMON/    LR/ A(NFRMX),             CBO

      CHARACTER TDATE*9, VS*3

      PARAMETER ( VS='1.0')

      COMMON/ML/ TDATE

      OPEN(  5, FILE='MELI',   STATUS='OLD', READONLY)
      OPEN(  6, FILE='MELO',   STATUS='NEW')
C     OPEN(  7, FILE='ICL',    STATUS='OLD', READONLY) ... IN INPUT
C     OPEN(  8, FILE='MELO8',  STATUS='NEW') ... IN LOOP
      OPEN(  9, FILE='RI',     STATUS='OLD', READONLY)
C     OPEN( 10, FILE='MELO10', STATUS='NEW') ... IN INPUT
      OPEN( 11, FILE='BLADE',  STATUS='OLD', READONLY)

      CALL      DATE(TDATE)

      WRITE(6,110) VS, TDATE
  110 FORMAT(' LIFT COEFFICIENT CALCULATION USING MELLOR''S ALGORITHM'/
     +  ' VERSION 'A3,T51,A9//)

      CALL      INPUT

      CALL      DGNK

      CALL      LOOP

      WRITE(6,900)
  900 FORMAT(///' END OF OUTPUT "3037-MELLOR-MELO"')
      STOP 'Done...'
      END
```

```
    Name                                     Bytes   Attributes

  0 $CODE                                      146   PIC CON REL LCL   SHR   EXE   RD NOWRT LONG
  1 $PDATA                                     145   PIC CON REL LCL   SHR NOEXE   RD NOWRT LONG
  2 $LOCAL                                     172   PIC CON REL LCL NOSHR NOEXE   RD   WRT LONG
  3 ML                                           9   PIC OVR REL GBL   SHR NOEXE   RD   WRT LONG

    Total Space Allocated                      472


ENTRY POINTS

    Address  Type  Name                        References

  0-00000000       MELLOR                          1#


VARIABLES

    Address  Type  Name         Attributes  References

  3-00000000  CHAR TDATE        COMM           42        46        56A       58


PARAMETER CONSTANTS

  Type  Name                                   References

   CHAR VS                                     42        44#       58

LABELS

    Address    Label                           References

  1-0000001D  110'                             58        59#
  1-0000006A  900'                             68        69#


FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name                                   References

        DGNK                                   64
        FOR$DATE_T_DS                          56
        FOR$OPEN                               48        49        52        54
        INPUT                                  62
        LOOP                                   66


  +------------------------------------------------------+
  |            KEY TO REFERENCE FLAGS                    |
  |     =    - Value Modified                            |
  |     #    - Defining Reference                        |
  |     A    - Actual Argument, possibly modified        |
  |     D    - Data Initialization                       |
  |    (n)   - Number of occurrences on line             |
  +------------------------------------------------------+


COMMAND QUALIFIERS

  F/LIS/CHE/CRO MELLORM

  /CHECK=(BOUNDS,OVERFLOW,UNDERFLOW)
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /STANDARD=(NOSYNTAX,NOSOURCE_FORM)
  /SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
  /WARNINGS=(GENERAL,NODECLARATIONS)
  /CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE
```

```
      SUBROUTINE DGNK

C (3037-MELLOR-DGNK) (D) CALC. DELTA G(N,K)(THETA)-G(N,K)(PI-THETA)
C                        (& OTHERS), CF. MILLOR (1959)
C REFERENCED BY MAIN (MELLOR); REFERENCES MTH$SQRT (TWICE)
C INPUT DX, FCP, FTP, LISTO, NFR, NSEC, NSEC2, & PI
C INDECES N & K, STARTING 0, ARE REPLACED BY N & K STARTING 1
C CODED BY W. CHIANG, 1983
C 10OCT86, S. HSU TRANSLATED FROM (3037-CASCAD[9845]-DGNK) WHICH WAS
C       ADAPTED FROM (3725-CASCAD[HP9845]-DGNK2)
C REVISED 12JAN87, CHIANG; TOUCHED 25MAR87

      PARAMETER ( NFRMX=15, NSECMX=500, NYMAX=100)
      PARAMETER ( NYSMX=MAX( NSECMX*2, MAX( NYMAX, NSECMX+1+2)),
     +            NYSMX2=MAX( NYMAX+1, NSECMX*2) + 2)

      COMMON/DG    /    DG(NFRMX,NFRMX,NSECMX),
     +                  DH(NFRMX,NFRMX,NSECMX),
     +                  DB(NFRMX,NSECMX), DT(NFRMX,NSECMX)
      COMMON/DGI   /    DX, NSEC
      COMMON/DGIL R/    LISTO, NFR
      COMMON/DG L  /    C(NFRMX)
      COMMON/D I   /    FCP(NYSMX2), FTP(NYSMX),              NSEC2
      COMMON/D IL  /    PI

      DIMENSION COA(NFRMX), SIB(NFRMX)

C 100 CONSTANT **********
      CON=DX*DX*2./PI
      CON4=CON*.25
      NSEC2P=NSEC2+1
      OT=DX/3.

C 200 INITIALIZATION **********
      DO      I=1,NFR
        DO      J=1,NFR
          DB(I,J)       =0.
          DT(I,J)       =0.
          DO    K=1,NSEC
            DG(I,J,K)   =0.
            DH(I,J,K)   =0.
            END DO
          END DO
        END DO

      IF (LISTO.EQ.1)      WRITE(6,210)
  210 FORMAT(//'  c/s   dG00   dG02   dG11   dG20   dG22   dH00   ',
     +          'dH01   dH10   dB0   dB1   dT0   dT1'/)

C 300 LOOP THRU SECTIONS **********
      DO750 ISEC=1,NSEC
        Y       =-OT
        X       =DX*ISEC-OT

C 400   LOOP THRU SEGMENTS *****
        IX      =ISEC+ISEC
        IY      =1
        DO 650 ISEG=1,(NSEC-ISEC)*2+1
          DXX   =( MOD(ISEG,2)+1 )*OT
          Y     =Y+DXX
          YM    =1.-Y
          XM    =1.-X

C         DEFINE COS & SIN FOR BOTH ANGLES OF K*THETA & K*THETAO
          COSB  =YM-Y
          COSB2 =COSB+COSB
          SIB(1)=0.
          SINB  =SQRT(YM*Y)*2.
          SIB(2)=SINB
          COA(1)=1.
          COA(2)=XM-X
          COSA2 =COA(2)+COA(2)
          SINA  =SQRT(XM*X)*2.
          DO    N=3,NFR
            COA(N)      =COA(N-1)*COSA2 - COA(N-2)
            SIB(N)      =SIB(N-1)*COSB2 - SIB(N-2)
            END DO
```

```
           FCPX  =FCP(IX)
           T8    =(1+COSB)*FCPX
           T5    =FTP(IY)/SINA
           T3    =FCPX*T5
           FCPXC =FCP(NSEC2P-IX)
           T9    =(1-COSB)*FCPXC
           T6    =FTP(NSEC2P-IY)/SINA
           T4    =FCPXC*T6
           RSIA2 =2./SINA
           RSS   =1./( SINA*SINB )
           RSSH  =RSS*.5
           COBRSS=COSB*RSS
           FCCD  =(T8-T9)*RSSH
           FCCS  =(T8+T9)*RSSH
           FCD   =(FCPX-FCPXC)/SINA
           FCS   =(FCPX+FCPXC)/SINA
           Q3    =T3+T4
           Q4    =T3-T4
           Q5    =T5+T6
           Q6    =T5-T6
           T2    =COSB-COA(2)
           T1    =Q3/T2
           T2    =Q4/T2

C 500      LOOP THRU K & N ***
           DO520K=1,NFR,2
             COAK=COA(K)
             C(K)=T2*COAK+C(K)
             DB(K,ISEC)  =Q4*COAK+DB(K,ISEC)
             DT(K,ISEC)  =Q6*COAK+DT(K,ISEC)
             DG(1,K,ISEC)=COAK*COBRSS+DG(1,K,ISEC)
             DH(1,K,ISEC)=COAK*FCCD  +DH(1,K,ISEC)
             T9  =COAK*RSIA2
             T7  =COAK*FCD
             T8  =COAK*FCS
             DO N=2,NFR,2
               DH(N,K,ISEC)      =SIB(N)*T7+DH(N,K,ISEC)
               END DO
             DO N=3,NFR,2
               DG(N,K,ISEC)      =SIB(N)*T9+DG(N,K,ISEC)
               DH(N,K,ISEC)      =SIB(N)*T8+DH(N,K,ISEC)
               END DO
   520       CONTINUE
           DO540K=2,NFR,2
             COAK=COA(K)
             C(K)=T1*COAK+C(K)
             DB(K,ISEC)=Q3*COAK+DB(K,ISEC)
             DT(K,ISEC)=Q5*COAK+DT(K,ISEC)
             DG(1,K,ISEC)=COAK*RSS +DG(1,K,ISEC)
             DH(1,K,ISEC)=COAK*FCCS+DH(1,K,ISEC)
             T9  =COAK*RSIA2
             T7  =COAK*FCD
             T8  =COAK*FCS
             DO N=2,NFR,2
               DG(N,K,ISEC)      =SIB(N)*T9+DG(N,K,ISEC)
               DH(N,K,ISEC)      =SIB(N)*T8+DH(N,K,ISEC)
               END DO
             DO N=3,NFR,2
               DH(N,K,ISEC)      =SIB(N)*T7+DH(N,K,ISEC)
               END DO
   540       CONTINUE
           IX    =IX+1
           IY    =IY+1
           X     =X+DXX
   650   CONTINUE
         IF (LISTO.EQ.1)
     +          WRITE(6,5660) (ISEC-.5)*DX, DG(1,1,ISEC),
     +          DG(1,3,ISEC), DG(2,2,ISEC), DG(3,1,ISEC), DG(3,3,ISEC),
     +          DH(1,1,ISEC), DH(1,2,ISEC), DH(2,1,ISEC),
     +          DB(1,ISEC)*.25, DB(2,ISEC)*.25,
     +          DT(1,ISEC)*.25, DT(2,ISEC)*.25
   750 CONTINUE
       DO      I=1,NFR
         DO      J=1,NFR
           DO   K=1,NSEC
             DG(I,J,K)   =DG(I,J,K)*CON
             DH(I,J,K)   =DH(I,J,K)*CON
             END DO
           END DO
         END DO
```

```
0158          Q          =CON/(PI+PI)
0159          DO         I=1,NFR
0160            C(I)     =C(I)*Q
0161            DO       J=1,NSEC
0162              DB(I,J)        =DB(I,J)*CON4
0163              DT(I,J)        =DT(I,J)*CON4
0164              END DO
0165            END DO
0166          RETURN
0167
0168     5660 FORMAT(F5.2,12F7.3)
0169          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes								
0	$CODE	2132	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
1	$PDATA	106	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT	LONG
2	$LOCAL	168	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	LONG
3	DG	960000	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
4	DGI	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
5	DGILR	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
6	DGL	60	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
7	DI	8012	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
8	DIL	4	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
	Total Space Allocated	970498									

ENTRY POINTS

Address	Type	Name	References
0-00000000		DGNK	1#

VARIABLES

Address	Type	Name	Attributes	References							
**	R*4	COAK		104=	105	106	107	108	109	110	111
				112	122=	123	124	125	126	127	128
				129	130						
**	R*4	COBRSS		89=	108						
**	R*4	CON		29=	30	153	154	158			
**	R*4	CON4		30=	162	163					
**	R*4	COSA2		72=	75						
**	R*4	COSB		65=	66(2)	79	83	89	98		
**	R*4	COSB2		66=	76						
4-00000000	R*4	DX	COMM	20	29(2)	32	53	143			
2-00000090	R*4	DXX		59=	60	141					
**	R*4	FCCD		90=	109						
**	R*4	FCCS		91=	127						
2-00000098	R*4	FCD		92=	111	129					
**	R*4	FCPX		78=	79	81	92	93			
**	R*4	FCPXC		82=	83	85	92	93			
2-0000009C	R*4	FCS		93=	112	130					
**	I*4	I		35=	37	38	40	41	150=	153(2)	154(2)
				159=	160(2)	162(2)	163(2)				
2-00000078	I*4	ISEC		51=	53	56(2)	58	106(2)	107(2)	108(2)	109(2)
				114(2)	117(2)	118(2)	124(2)	125(2)	126(2)	127(2)	132(2)
				133(2)	136(2)	143(13)					
2-0000008C	I*4	ISEG		58=	59						
2-00000084	I*4	IX		56=	78	82	139(2)=				
2-00000088	I*4	IY		57=	80	84	140(2)=				
**	I*4	J		36=	37	38	40	41	151=	153(2)	154(2)
				161=	162(2)	163(2)					
**	I*4	K		39=	40	41	103=	104	105(2)	106(2)	107(2)
				108(2)	109(2)	114(2)	117(2)	118(2)	121=	122	123(2)
				124(2)	125(2)	126(2)	127(2)	132(2)	133(2)	136(2)	152=
				153(2)	154(2)						
5-00000000	I*4	LISTO	COMM	21	46	143					
**	I*4	N		74=	75(3)	76(3)	113=	114(3)	116=	117(3)	118(3)
				131=	132(3)	133(3)	135=	136(3)			
5-00000004	I*4	NFR	COMM	21	35	36	74	103	113	116	121

				131	135	150	151	159		
4-00000004	I*4	NSEC	COMM	20	39	51	58	152	161	
7-00001F48	I*4	NSEC2	COMM	23	31					
**	I*4	NSEC2P		31=	82	84				
**	R*4	OT		32=	52	53	59			
8-00000000	R*4	PI	COMM	24	29	158(2)				
**	R*4	Q		158=	160					
2-000000A0	R*4	Q3		94=	99	124				
**	R*4	Q4		95=	100	106				
**	R*4	Q5		96=	125					
**	R*4	Q6		97=	107					
2-00000094	R*4	RSIA2		86=	110	128				
**	R*4	RSS		87=	88	89	126			
**	R*4	RSSH		88=	90	91				
**	R*4	SINA		73=	80	84	86	87	92	93
**	R*4	SINB		68=	69	87				
**	R*4	T1		99=	123					
2-000000A4	R*4	T2		98=	99	100(2)=	105			
**	R*4	T3		81=	94	95				
**	R*4	T4		85=	94	95				
**	R*4	T5		80=	81	96	97			
**	R*4	T6		84=	85	96	97			
**	R*4	T7		111=	114	129=	136			
**	R*4	T8		79=	90	91	112=	118	130=	133
**	R*4	T9		83=	90	91	110=	117	128=	132
2-00000080	R*4	X		53=	62	71	73	141(2)=		
**	R*4	XM		62=	71	73				
2-0000007C	R*4	Y		52=	60(2)=	61	65	68		
**	R*4	YM		61=	65	68				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
6-00000000	R*4	C	COMM	60	(15)	22	105(2)=	123(2)=	160(2)=	
2-00000000	R*4	COA		60	(15)	26	70=	71=	72(2)	75(3)=
						98	104	122		
3-000DBBA0	R*4	DB	COMM	30000	(15, 500)	17	37=	106(2)=	124(2)=	143(2)
						162(2)=				
3-00000000	R*4	DG	COMM	450000	(15, 15, 500)	17	40=	108(2)=	117(2)=	126(2)=
						132(2)=	143(5)	153(2)=		
3-0006DDD0	R*4	DH	COMM	450000	(15, 15, 500)	17	41=	109(2)=	114(2)=	118(2)=
						127(2)=	133(2)=	136(2)=	143(3)	154(2)=
3-000E3000	R*4	DT	COMM	30000	(15, 500)	17	38=	107(2)=	125(2)=	143(2)
						163(2)=				
7-00000000	R*4	FCP	COMM	4008	(1002)	23	78	82		
7-00000FA8	R*4	FTP	COMM	4000	(1000)	23	80	84		
2-0000003C	R*4	SIB		60	(15)	26	67=	69=	76(3)=	114
						117	118	132	133	136

PARAMETER CONSTANTS

Type	Name	References			
I*4	NFRMX	13#	17(6)	22	26(2)
I*4	NSECMX	13#	14(3)	17(4)	
I*4	NYMAX	13#	14(2)		
I*4	NYSMX	14#	23		
I*4	NYSMX2	14#	23		

LABELS

Address	Label	References	
1-00000000	210'	46	47#
**	520	103	120#
**	540	121	138#
**	650	58	142#
**	750	51	149#

```
1-00000061  5660'                        143       168#

FUNCTIONS AND SUBROUTINES REFERENCED

 Type  Name                             References

  R*4  MTH$SQRT                           68        73

 +----------------------------------------------+
 |          KEY TO REFERENCE FLAGS              |
 |    =    - Value Modified                     |
 |    #    - Defining Reference                 |
 |    A    - Actual Argument, possibly modified |
 |    D    - Data Initialization                |
 |   (n)   - Number of occurrences on line      |
 +----------------------------------------------+
```

```
      SUBROUTINE FFCP( N, XCC, FCP)
C (3037-MELLOR-FFCP) FIND Fc' USING A FORMULA ( FOR NACA a=1.0)
C REF. ABBOTT & VON DOENHOFF, 1959, EQ. 4.25
C INPUT N & XCC; OUTPUT FCP
C REFERENCED BY INPUT (TWICE); REFERENCES MTH$ALOG & MTH$ATAN
C ADAPTED FROM (3037-MELLOR-INPUT); CODED BY W. CHIANG, 25MAR87; TOUCHED 31MAR87

C N     # OF PTS.; N>2
C XCC   DISTANCE ALONG CHORD, RANGES FROM 0 TO 100, SIZE N, XCC IN
C       ASCENDING ORDER EXCEPT XCC(N-1) & XCC(N) COULD BE ANY NUMBER
C       FROM 0 TO 100
C FCP   dY/dX, SIZE N

      DIMENSION FCP(N), XCC(N)

      IF ( XCC(1).LT.0.) THEN
        GOTO 1100
       ELSE
        IF ( XCC(1).LT..001) THEN
         N1    =2
        ELSE
         N1    =1
        END IF
       END IF

      PI4       =ATAN(1.) * 16.
      DO     I=N1,N
        IF ( XCC(I).GT.99.999 ) THEN
          IF ( I.LT.N1+2 .OR. XCC(I).GT.100. ) THEN
            PRINT *, 'I<N1+2 or XCC(I)>100... ', I, N1, XCC(I)
            STOP 'FFCP.210'
           ELSE
            FCP(I)=FCP(I-1)*2.-FCP(I-2)
           END IF
         ELSE
          FCP(I)=ALOG( 100./XCC(I)-1. ) / PI4
         END IF
       END DO
      IF (N1.EQ.2)     FCP(1)=FCP(2)+FCP(2)-FCP(3)

      RETURN

 1100 PRINT *, 'CHECK 0<=XCC=>100...', XCC(1), XCC(N), N
      STOP 'FFCP.1100'
      END
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 $CODE	433	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 $PDATA	61	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 $LOCAL	120	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
Total Space Allocated	614	

ENTRY POINTS

Address	Type	Name	References
0-00000000		FFCP	1#

VARIABLES

Address	Type	Name	Attributes	References					
**	I*4	I		27=	28	29(2)	30(2)	33(3)	36(2)
AP-00000004@	I*4	N		1	14(2)	27	43(2)		
2-00000000	I*4	N1		20=	22=	27	29	30	39
**	R*4	PI4		26=	36				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References

```
AP-0000000C@ R*4  FCP                    ** (*)          1     14     33(3)=   36=     39(4)=
AP-00000008@ R*4  XCC                    ** (*)          1     14     16       19      28
                                                         29    30     36       43(2)
```

LABELS

```
  Address   Label                    References

    **      1100                       17       43#
```

FUNCTIONS AND SUBROUTINES REFERENCED

```
  Type  Name                         References

   R*4  MTH$ALOG                       36
   R*4  MTH$ATAN                       26

+----------------------------------------------+
|           KEY TO REFERENCE FLAGS             |
|    =   - Value Modified                      |
|    #   - Defining Reference                  |
|    A   - Actual Argument, possibly modified  |
|    D   - Data Initialization                 |
|   (n)  - Number of occurrences on line       |
+----------------------------------------------+
```

COMMAND QUALIFIERS

```
  F/LIS/CHE/CRO FFCP

  /CHECK=(BOUNDS,OVERFLOW,UNDERFLOW)
  /DEBUG=(NOSYMBOLS,TRACEBACK)
  /STANDARD=(NOSYNTAX,NOSOURCE_FORM)
  /SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
  /WARNINGS=(GENERAL,NODECLARATIONS)
  /CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
  /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE
```

```
      SUBROUTINE GNK(KA)

C (3037-MELLOR-GNK) (G) CALC. DOUBLE FOURIER INTEGRALS
C                         (CF. MILLOR, 1959, TABLE 3)
C REFERENCED BY LOOP, REFERENCES NONE
C INPUT AI, C, CS, DB, DG, DH, DT, DX, LISTO, NFR, NSEC, KA, R, XSA, & XSI
C OUTPUT B, G, H, & T
C REQUIRES DATA GENERATED FROM (3037-RI)
C INDECES N & K, STARTING 0, ARE REPLACED BY N & K STARTING 1
C CODED BY W. CHIANG, 13JUL83, REVISED 30AUG83
C TRANSLATED FROM (3037-CASCAD[9845]-GNK), S. HSU, 10OCT86
C REVISED 30JAN87, CHIANG; TOUCHED 05MAR87

      PARAMETER ( KAMX=8, KXMX=76, NFRMX=15, NSECMX=500)

      COMMON/DG   /     DG(NFRMX,NFRMX,NSECMX),
     +                  DH(NFRMX,NFRMX,NSECMX),
     +                  DB(NFRMX,NSECMX), DT(NFRMX,NSECMX)
      COMMON/DGI  /     DX, NSEC
      COMMON/DGILR/     LISTO, NFR
      COMMON/DG L /     C(NFRMX)
      COMMON/ GI  /     AI(KXMX,KAMX), R(KXMX,KAMX),            XSA, XSI
      COMMON/ GILR/     CSA
      COMMON/ G LR/     G(NFRMX,NFRMX), H(NFRMX,NFRMX),
     +                  B(NFRMX), T(NFRMX),             AMDAD, CS

C 100 CONSTANT ***
      DXCS=DX*CS

C 200 INITIALIZATION ***
      DO      K=1,NFR
        B(K)    =0.
        T(K)    =0.
        DO      N=1,NFR
          G(N,K)=0.
          H(N,K)=0.
          END DO
        END DO

C 300 LOOP THRU SECTIONS ***
      DO    ISEC=1,NSEC
        XK      =( (ISEC-.5)*DXCS-XSA )/XSI + 1.
        KX1=INT(XK)
        KX2=KX1+1
        RV=(XK-KX1)*R(KX2,KA)+(KX2-XK)*R(KX1,KA)
        AIV=(XK-KX1)*AI(KX2,KA)+(KX2-XK)*AI(KX1,KA)

        DO      K=1,NFR
          B(K)  =DB(K,ISEC)*RV  + B(K)
          T(K)  =DT(K,ISEC)*AIV + T(K)
          DO    N=1,NFR
            G(N,K)      =DG(N,K,ISEC)*RV  + G(N,K)
            H(N,K)      =DH(N,K,ISEC)*AIV + H(N,K)
            END DO
          END DO
        END DO

      DO      K=1,NFR
        B(K)    =B(K)*CS + C(K)
        T(K)    =T(K)*CS
        DO      N=1,NFR
          G(N,K)=G(N,K)*CS
          H(N,K)=H(N,K)*CS
          END DO
        END DO
      G(1,1)=G(1,1)+1.
      DO      N=2,NFR
        G(N,N)  =G(N,N)-1.
        END DO

      IF (LISTO.EQ.1) THEN
        IF (CS.EQ.CSA) THEN
          WRITE(6,5400) AMDAD, CS, G(1,1), G(1,3), G(2,2), G(3,1),
     +          G(3,3), H(1,1), H(1,2), H(2,1), B(1), B(2), T(1), T(2)
         ELSE
          WRITE(6,5410)        CS, G(1,1), G(1,3), G(2,2), G(3,1),
     +          G(3,3), H(1,1), H(1,2), H(2,1), B(1), B(2), T(1), T(2)
```

```
            END IF
          END IF

        IF (LISTO.EQ.11) THEN
          WRITE(6,5420) AMDAD, CS, G(1,1), G(1,3), G(2,2), G(3,1),
     +           G(3,3), H(1,1), H(1,2), H(2,1), B(1), B(2), T(1), T(2)
          WRITE(10,*) ( ( G(N,K), K=1,NFR), N=1,NFR),
     +                ( ( H(N,K), K=1,NFR), N=1,NFR), ( B(N), N=1,NFR),
     +                ( T(N), N=1,NFR)
          END IF

        RETURN

 5400   FORMAT(/F4.0,F6.1,12F6.3)
 5410   FORMAT(      F10.1,12F6.3)
 5420   FORMAT( F4.0,F6.1,12F6.3)
        END
```

PROGRAM SECTIONS

Name	Bytes	Attributes								
0 $CODE	1495	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
1 $PDATA	34	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT	LONG
2 $LOCAL	8	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	LONG
3 DG	960000	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
4 DGI	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
5 DGILR	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
6 DGL	60	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
7 GI	4872	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
8 GILR	4	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
9 GLR	1928	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
Total Space Allocated	968417									

ENTRY POINTS

Address	Type	Name	References
0-00000000		GNK	1#

VARIABLES

Address	Type	Name	Attributes	References							
**	R*4	AIV		46=	50	53					
9-00000780	R*4	AMDAD	COMM	24	73	82					
9-00000784	R*4	CS	COMM	24	28	59	60	62	63	72	73
				76	82						
8-00000000	R*4	CSA	COMM	23	72						
4-00000000	R*4	DX	COMM	19	28						
2-00000000	R*4	DXCS		28=	42						
2-00000004	I*4	ISEC		41=	42	49	50	52	53		
**	I*4	K		31=	32	33	35	36	48=	49(3)	50(3)
				52(3)	53(3)	58=	59(3)	60(2)	62(2)	63(2)	84(4)=
AP-00000004@	I*4	KA		1	45(2)	46(2)					
**	I*4	KX1		43=	44	45(2)	46(2)				
**	I*4	KX2		44=	45(2)	46(2)					
5-00000000	I*4	LISTO	COMM	20	71	81					
**	I*4	N		34=	35	36	51=	52(3)	53(3)	61=	62(2)
				63(2)	67=	68(4)	84(8)=				
5-00000004	I*4	NFR	COMM	20	31	34	48	51	58	61	67
				84(6)							
4-00000004	I*4	NSEC	COMM	19	41						
**	R*4	RV		45=	49	52					
**	R*4	XK		42=	43	45(2)	46(2)				
7-00001300	R*4	XSA	COMM	22	42						
7-00001304	R*4	XSI	COMM	22	42						

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
7-00000000	R*4	AI	COMM	2432	(76, 8)	22	46(2)			
9-00000708	R*4	B	COMM	60	(15)	24	32=	49(2)=	59(2)=	73(2)

						76(2)	82(2)	84		
6-00000000	R*4	C	COMM	60	(15)	21	59			
3-0000BBA0	R*4	DB	COMM	30000	(15, 500)	16	49			
3-00000000	R*4	DG	COMM	450000	(15, 15, 500)	16	52			
3-0006DDD0	R*4	DH	COMM	450000	(15, 15, 500)	16	53			
3-000E3DD0	R*4	DT	COMM	30000	(15, 500)	16	50			
9-00000000	R*4	G	COMM	900	(15, 15)	24	35=	52(2)=	62(2)=	66(2)=
						68(2)=	73(5)	76(5)	82(5)	84
9-00000384	R*4	H	COMM	900	(15, 15)	24	36=	53(2)=	63(2)=	73(3)
						76(3)	82(3)	84		
7-00000980	R*4	R	COMM	2432	(76, 8)	22	45(2)			
9-00000744	R*4	T	COMM	60	(15)	24	33=	50(2)=	60(2)=	73(2)
						76(2)	82(2)	84		

PARAMETER CONSTANTS

Type	Name	References			
I*4	KAMX	14#	22(2)		
I*4	KXMX	14#	22(2)		
I*4	NFRMX	14#	16(6)	21	24(6)
I*4	NSECMX	14#	16(4)		

LABELS

Address	Label	References	
1-00000000	5400'	73	91#
1-0000000D	5410'	76	92#
1-00000016	5420'	82	93#

```
+----------------------------------------------+
|           KEY TO REFERENCE FLAGS             |
|   =   - Value Modified                       |
|   #   - Defining Reference                   |
|   A   - Actual Argument, possibly modified   |
|   D   - Data Initialization                  |
|  (n)  - Number of occurrences on line        |
+----------------------------------------------+
```

```
      SUBROUTINE INPUT

C (3037-MELLOR-INPUT) (I) INPUT DATA
C REFERENCED BY MAIN (MELLOR)
C REFERENCES FFCP (TWICE), FILONC & SPLIN1 (4 TIMES) & FOR$CLOSE (TWICE),
C        FOR$OPEN, MTH$COSD (TWICE), & MTH$SIND (TWICE)
C INPUT AI, CA, CBI, CC, CIN, CSA, CSI, CSZ, EPS, F$, ICL, IFLAT,
C        INMAX, LAMDA, LAMDI, LAMDZ, LIST, MH, NB, NFR, NFRMX, NSEC,
C        NSECMX, NYC, NYT, R, TTL$, XC, XT, XSA, XSI, XSZ, YC, & YT
C CODED BY W. CHIANG, 14JUL83
C 10OCT86, S. HSU TRANSLATED FROM (3037-CASCAD[9845]-INPUT) WHICH WAS
C        ADAPTED FROM (3725-CASCAD[HP9845]-INPUT)
C REVISED 31MAR87; SLIGHTLY REVISED 01JUN88; CHIANG

C
C READ1 TITLE (WILL NOT SHOW IN OUTPUT)
C
C READ2 TITLE (TO BE SHOWN IN OUTPUT)
C
C READ3
C MH    = 1  TO CALCULATE THE CASE SHOWN IN MELLOR (1959);
C       = 2  FOR THE CASE FOLLOWS HERRIG, et al. (1951)
C       =11  TO PREPARE A TABLE OF G, H, B, & T FUNCTIONS (OF STAGGER
C                ANGLE) FOR A SPEIFIED SOLIDITY
C INCAM = 1  IF THE CAMBER IS A CIRCULAR ARC;
C       = 2  IF THE BLADE IS NACA 65 W/ a=1.0;
C       = 3* IF THE CAMBER IS TO BE CALCULATED FROM A FORMULA
C IFLAT (NOT USED IF INCAM=1)
C       = 0* IF THE INPUT CAMBER IS NOT TO BE MODIFIED;
C       = 1  IF THE SLOPES OF INPUT CAMBER  ARE TO BE KEPT CONSTANT
C                WITHIN 5% OF BOTH ENDS (SEE MELLOR, 1959)
C NSEC  # OF SEGMENTS ALONG THE CHORD, e.g., 50;
C        SET TO 10 IF MH=1 TO COMPARE W/ TABLES IN MELLOR (1959)
C NFR   # OF FOURIER SERIES TERMS TO BE USED, e.g., NFR=3 TO HAVE A0,
C        A1, & A2
C LIST  = 0  IF CAMBER & THICKNESS DATA ARE NOT TO BE LISTED;
C       = 1* TO LIST
C
C READ4
C CBID  INPUT IDEAL CB (KNOWN) FOR THE INPUT CAMBER, IF THE CALCULATED CB
C        IS TO BE NEGLECTED; OTHERWISE, INPUT A NUMBER >=9 TO USE THE
C        CALCULATED CB
C SEND1 A FACTOR TO BE USED BY 4 SUB SPLIN1 IN THIS ROUTINE.... IT IS TO
C        BE APPLIED TO S(2) & S(N-1) TO OBTAIN S(1) & S(N), S BEING
C        CURVATURE... NORMALLY 0, .5, OR 1... e.g., 1 FOR CAMBER &
C        THICKNESS DISTRIBUTIONS
C
C ***** SKIP READ5 & READ6, IF MH.NE.1 *****
C READ5 (IIF MH=1)
C LISTO = 1  TO MAKE DG00-DT1 & G00-T1 TABLES (MELLOR, 1959);
C       = 2  TO MAKE A01-AT TABLE;
C       = 3  TO MAKE CL-AT TABLE
C
C READ6 (IIF MH=1)
C ALPHAMMEAN ANGLE OF INCIDENCE, IN deg. (NOT USED IF LISTO=1)
C
C ***** SKIP READ7 & READ8, IF MH.NE.2 *****
C READ7 (IIF MH=2)
C IRES  = 1  TO FIND LIFT COEF. BY CL=2*PI*(A1+A2);
C       = 2  TO CALC. CL AS ABOVE PLUS BETA1 & CL1
C ISCL  = 0  TO DO NOTHING
C       = 1  TO STORE CALCULATED SET OF LIFT COEFFICIENTS TO BE COMPARED
C            W/ HERRIG, et al., 1951
C
C READ8 (IIF MH=2)
C CC     FACTOR FOR EFFECTIVE CB, <=1, 1 FOR THEO. VALUES, MELLOR HAS .725
C CA     FACTOR FOR EFFECTIVE ALPHA(M), <=1, 1 FOR THEORETICAL VALUES
C CSA    STARTING C/S (SOLIDITY) FOR CASES TO BE CALCULATED, e.g., 0.
C CSI    INCREMENT OF C/S TO BE CALCULATED, e.g., 2.0
C CSZ    ENDING C/S, e.g., .5
C SEND5 A FACTOR TO BE USED BY 4 SUB SPLIN1 IN SUB LOOP... IT IS TO
C        BE APPLIED TO S(2) & S(N-1) TO OBTAIN S(1) & S(N), S BEING
C        CURVATURE... NORMALLY 0, .5, OR 1... e.g., .5 FOR SOME
C        DISTRIBUTIONS
C
C ***** SKIP READ9 IF MH.NE.11 *****
C READ9 (IIF MH=11)
```

```
C CSA   C/S (SOLIDITY) FOR THE GENERATED TABLE OF G, H, B, & T FUNCTIONS
C
C READ10
C IDUM  INTEGER 999, TO CHECK THE END OF INPUT FROM UNIT 5

C ***** SKIP READ701 THRU READ706, IF MH.NE.2 *****
C READ701 (IIF MH=2)
C NB    # OF ELEMENTS IN ARRAY BETA1
C NS    # OF ELEMENTS IN ARRAY SIG
C NC    # OF ELEMENTS IN ARRAY CB1
C NA    # OF ELEMENTS IN ARRAY ALPHA1
C
C READ702 (IIF MH=2)
C BETA1 BETA AT INLET, IN DEG, SIZE NB
C
C READ703 (IIF MH=2)
C SIG   SOLIDITY, SIZE NS
C
C READ704 (IIF MH=2)
C CB1   Cb, SIZE NC
C
C READ705 (IIF MH=2)
C ALPHA1INCIDENT ANGLE AT INLET, IN DEG, SIZE NA
C
C READ706 (IIF MH=2)
C ICL   INTEGER VALUES OF 10000 TIMES OF LIFT COEFFICIENT, EXPERIMENTAL
C       VALUES WHEN INPUT, CALCULATED VALUES WHEN OUTPUT, SIZE (NA,NC,NS,NB)...
C       THOSE NOT USED ARE ASSIGNED A VALUE HIGHER THAN 32600

C READ901
C TITL  TITLE FOR THIS R & I FUNCTION FILE
C
C READ902
C EPS   CLOSING CRITERION USED IN PROGRAM RI, NO USE HERE
C AMDA  STARTING STAGGER ANGLE, IN DEG
C AMDI  INTERVAL OF STAGGER ANGLE, IN DEG
C AMDZ  ENDING STAGGER ANGLE, IN DEG
C XSA   STARTING XS ( [XO-X]/S, SEE MELLOR, 1959)
C XSI   INTERVAL OF XS
C XSZ   ENDING XS
C
C READ903
C R     R FUNCTION, SEE MELLOR, 1959
C AI    I FUNCTION, SEE MELLOR, 1959
C JACK  A DUMMY NUMBER, SHOULD BE 999 IF DATA FILE IS CORRECT

C ***** SKIP READ1101 THRU READ1104 IF INCAM.NE.2 *****
C READ1101 (IIF INCAM=2)
C TITL  TITLE FOR THIS CAMBER DATA FILE
C
C READ1102 (IIF INCAM=2)
C NYC   # OF DATA SETS FOR CAMBER COORDINATES
C
C READ1103 (IIF INCAM=2)
C XC    X-COORDINATES FOR CAMBER YC, SIZE NYC
C
C READ1104 (IIF INCAM=2)
C YC    CAMBER AS FUNCTION OF XC, SIZE NYC
C
C READ1105
C TITL  TITLE FOR THIS BLADE THICKNESS FILE
C
C READ1106
C NYT   # OF DATA SETS FOR BLADE THICKNESS
C
C READ1107
C XT    X-COORDINATES FOR DATA OF THICKNESS YT, SIZE NYT
C
C READ1108
C YT    HALF-THICKNESS OF BLADE AT CORRESPONDING XT, SIZE BYT
C
C READ1109
C JACK  = 999 AS A CHECK
C

      CHARACTER TDATE*9, TITL*90

      PARAMETER ( KAMX=8, KXMX=76, NAMX=19, NBMX=4, NCMX=9, NFRMX=15,
     +                      NSMX=5, NSECMX=500, NYMAX=100)
      PARAMETER ( NYSMX=MAX( NSECMX*2, MAX( NYMAX, NSECMX+1+2)),
```

```
     +             NYSMX2=MAX( NYMAX+1, NSECMX*2) + 2)

      COMMON/M  IL /    TDATE
      COMMON/ DGI  /    DX, NSEC
      COMMON/ DGILR/    LISTO, NFR
      COMMON/ D I  /    FCP(NYSMX2), FTP(NYSMX),            NSEC2
      COMMON/ D IL /    PI
      COMMON/  GI  /    AI(KXMX,KAMX), R(KXMX,KAMX),        XSA, XSI
      COMMON/  GILR/    CSA
      COMMON/   IL /    ICL(NAMX,NCMX,NSMX,NBMX),
     +                  BETA1(NBMX), CB1(NSMX), V(NFRMX),
     +                  AMDA, AMDZ, AMDI, CA, CC, CSI, CSZ, SALPHA,
     +                  SEND5, TOC,            ISCL, MH, NA, NB, NC, NS
      COMMON/   ILR/    ALPHAM, CALPHA, CB, PI2
      COMMON/   I R/    IRES,           SA2PI, TOC2PI

      DIMENSION ALPHA1(NAMX), DUMMY(NYSMX2), SIG(NSMX),
     +          XC(NYMAX), XCC(NYSMX2), XT(NYMAX), YC(NYMAX), YT(NYMAX)

      DATA      PI/3.14159265359/

C 100
      READ (5,5101) TITL                                                READ1
      READ (5,5101) TITL                                                READ2
      WRITE(6,5103) TITL
      READ (5,*)  MH, INCAM, IFLAT, NSEC, NFR, LIST                     READ3
      READ (5,*)                                    CBID, SEND1         READ4
      WRITE(6,120) MH, INCAM, IFLAT, NSEC, NFR, LIST, CBID, SEND1
CC120 FORMAT('     .   MH     INCAM     IFLAT     NSEC'/1X5I10//
  120 FORMAT(' MH, INCAM, IFLAT:'T40,3I5/
     + ' # OF SEGMENT ON A BLADE:'T40,I5/
     + ' # OF FOURIER SERIES TERMS:'T40,I5/' LIST:'T40,I5/
     + ' CBI, SEND1:'T40,2F8.2)
      IF (MH.LT.1 .OR. NSEC.GT.NSECMX .OR. NFR.LT.3 .OR. NFR.GT.NFRMX)
     + THEN
        PRINT *, 'CHECK MH, NSEC, NFR...', MH, NSEC, NFR, NSECMX, NFRMX
        STOP 'INPUT.122'
        END IF

      IF (MH-2) 130, 140, 170

  130 READ (5,*) LISTO                                                  READ5
      READ (5,*) ALPHAM                                                 READ6
      WRITE(6,135) LISTO, ALPHAM
  135 FORMAT(' LISTO:'T40,I5/
     +          ' MEAN AIR ANGLE (deg) RELATIVE TO CHORD:'T40,F7.1/)
      CSA       =.5
      CSI       =.5
      CSZ       =2.
      IRES      =3
      GOTO 210

  140 READ (5,*)   IRES, ISCL                                           READ7
      READ (5,*)                 CC, CA, CSA, CSZ, CSI, SEND5           READ8
      WRITE(6,150) IRES, ISCL, CC, CA, CSA, CSZ, CSI, SEND5
  150 FORMAT(' IRES, ISCL:'T40,2I5/
     +          ' CC, CA, CSA, CSZ, CSI, SEND5:'T40,6F8.2/)
      OPEN( 7, FILE='ICL', STATUS='OLD', READONLY)
      READ (7,*)   NB, NS, NC, NA                                       READ701
      WRITE(6,160) NB, NS, NC, NA
  160 FORMAT(    ' NB, NS, NC, NA:'T40,4I5)
      IF (NB.GT.NBMX .OR. NS.GT.NSMX .OR. NC.GT.NCMX .OR. NA.GT.NAMX)
     +          THEN
        PRINT *, 'CHECK NB, NS, NC, NA...', NB, NS, NC, NA, NBMX,
     +                  NSMX, NCMX, NAMX
        END IF
      READ (7,*) (  BETA1(I), I=1,NB)                                   READ702
      READ (7,*) (     SIG(I), I=1,NS)                                  READ703
      READ (7,*) (     CB1(I), I=1,NC)                                  READ704
      READ (7,*) ( ALPHA1(I), I=1,NA)                                   READ705
      READ (7,*) ( ( ( ( ICL(IA,IC,IS,IB), IA=1,NA), IC=1,NC),
     +                                     IS=1,NS), IB=1,NB), JACK     READ706
       IF (JACK.NE.999) THEN
        PRINT *, 'CHECK TAPE7 DATA...', JACK
        STOP 'INPUT.168'
        END IF
      CLOSE(7)
      GOTO 210

  170 READ (5,*) CSA                                                    READ9
```

```
      WRITE(6,*) 'SOLIDITY = ', CSA
      CSI       =1.
      CSZ       =CSA
      LISTO     =11

C 200 READ R & I DATA GENERATED BY (3037-RIS) *****
  210 READ (9,5101) TITL                                               READ901
      WRITE(6,5103) TITL
      READ (9,*) EPS, AMDA, AMDZ, AMDI, XSA, XSZ, XSI                  READ902
      KAZ       =(AMDZ-AMDA)/AMDI + 1
      KXZ       =(XSZ-XSA)/XSI + 1
      IF (KAZ.GT.KAMX .OR. KXZ.GT.KXMX) THEN
        PRINT *, 'KAZ OR KXZ EXCEEDED DIM...', KAZ, KXZ, KAMX, KXMX
        STOP 'INPUT.210'
        END IF
      READ (9,*) ( (  R(KX,KA), KA=1,KAZ), KX=1,KXZ),
     +           ( ( AI(KX,KA), KA=1,KAZ), KX=1,KXZ), JACK             READ903
      IF (JACK.NE.999) THEN
        PRINT *, 'CHECK R & I DATA...', JACK
        STOP 'INPUT.220'
        END IF
      CLOSE(9)

C 250 PREPARE DATA FILE MELO10.DAT (TO BE USED BY, e.g., 3037-DSN3-MELLOR2)
      IF (LISTO.EQ.11) THEN
        IF (CSA.NE.CSZ) THEN
          PRINT *, 'CHECK CSA, CSZ FOR LISTO=11...', CSA, CSZ, LISTO
          STOP 'INPUT.3002'
          END IF
        OPEN( 10, FILE='MELO10', STATUS='NEW')
        WRITE(10,260) TDATE                                           W10.1
  260   FORMAT(
     +  ' (3037-MELLOR-MELO10) G & H FUNCTIONS (MELLOR, 1959, TABLE 3)'
     +  T71,A9)
        WRITE(10,*) 'NFR'                                             W10.2
        WRITE(10,*)  NFR                                              W10.3
        WRITE(10,*) 'CS, AMDA, AMDI, AMDZ'                            W10.4
        WRITE(10,*) CSA, AMDA, AMDI, AMDZ                             W10.5
        IF (INCAM.LT.2 .OR. INCAM.GT.3) THEN
          PRINT *, 'CHECK INCAM WHILE LISTO=11...', INCAM, LISTO
          STOP 'INPUT.265'
          END IF
        END IF

C 300 READ CAMBER & THICKNESS DATA *****
      NYC       =0
      PI2       =PI+PI
      GOTO ( 1110, 1210, 1310), INCAM

 1110 PRINT *, 'SUB. CIRARC NOT EXISTING...'
CCC     CALL    CIRARC( NYC, XC, YC)
      STOP 'INPUT.1110'

C1200 INPUT CAMBER DATA
 1210 READ (11,5101) TITL                                              READ1101
      WRITE(6,5103)  TITL
      READ (11,*) NYC                                                  READ1102
      IF (NYC.GT.NYMAX) THEN
        PRINT *, 'NYCN.GT.NYMAX...', NYC, NYMAX
        STOP 'INPUT.1215'
        END IF
      READ (11,*) ( XC(I), I=1,NYC)                                    READ1103
      READ (11,*) ( YC(I), I=1,NYC)                                    READ1104

      IF (LIST.GT.0) THEN
        CALL    SPLIN1( NYC, XC, YC, NYC, XC, YC, FCP, DUM, SEND1,
     +                  1.E-9)

        WRITE(6,1220)
 1220   FORMAT(/21X'X         Yc        dYc/dX'/)
        DC      I=1,NYC
          WRITE(6,6222) XC(I), YC(I), FCP(I)
          IF ( MOD(I,5).EQ.0 ) WRITE(6,6222)
          END DO
        END IF

C1300 YC'
 1310 READ (5,*) JACK                                                  READ10
      IF (JACK.NE.999) THEN
        PRINT *, 'CHECK DATA IN UNIT 5...', JACK
```

```
        END IF
      CLOSE(5)

      NYS        =MAX( NYC, NSEC+1)
      IF ( MOD(NYS,2).EQ.0 )       NYS=NYS+1
      IF (NYS.GE.NYSMX2) THEN
        PRINT *, 'NYS.GE.NYSMX2', NYS, NYSMX2
        END IF

      DTHETA    =PI/(NYS-1)
      DO        I=1,NYS
        XCC(I)  =(I-1)*DTHETA
        END DO
      DO        I=1,NYS
        XCC(I)  =COS( XCC(I) )
        END DO
      DO        I=1,NYS
        XCC(I)  =( 1.-XCC(I) ) * 50.
        END DO
C1320
      IF (IFLAT.EQ.1) THEN
        NYSP1   =NYS+1
        NYSP2   =NYSP1+1
        XCC(NYSP1)      =5.
        XCC(NYSP2)      =95.
       ELSE
        NYSP2   =NYS
       END IF
      IF (NYSP2.GT.NYSMX) THEN
        PRINT *, 'NYSP2.GT.NYSMX...', NYSP2, NYSMX
        STOP 'INPUT.1329'
        END IF

      IF (INCAM.EQ.3) THEN

        CALL    FFCP( NYSP2, XCC, FCP)

       ELSE

        CALL    SPLIN1( NYC, XC, YC, NYSP2, XCC, DUMMY, FCP, DUM,
     +                  SEND1, 1.E-9)

       END IF

C1330 MAKE CAMBER SLOPE CONSTANT FOR PORTIONS < 5% & > 95 %
      IF (IFLAT.EQ.1) THEN
        ITEM    =NYS*.3
        FCPA    =FCP(NYSP1)
        FCPZ    =FCP(NYSP2)
        DO      I=1,ITEM
          FCP(I)=MIN( FCP(I), FCPA)
          END DO
        DO      I=NYS-ITEM,NYS
          FCP(I)=MAX( FCP(I), FCPZ)
          END DO
        WRITE(6,1340)
 1340   FORMAT(//' CAMBER SLOPE ARE FORCED TO BE CONSTANT FOR THE 1ST ',
     +         '& LAST 5% REGIONS...'//)
        END IF

C1360 PREPARE DATA FILE MELO10.DAT (CONT.)
      IF (LISTO.EQ.11) THEN

        CALL    FILONC( NYS-1, FCP, NFR, V)

        WRITE(6,1365) ( V(N), N=1,NFR)
 1365   FORMAT(//' VECTOR OF INTEGRAING Fc''*cos(theta) IS:'/(10F12.5))
        WRITE(10,*) 'NYS'                                               W10.6
        WRITE(10,*)  NYS                                                W10.7
        WRITE(10,*) '( XCC(I),      I=1,NYS) & ( FCP(I), I=1,NYS)'      W10.8
        WRITE(10,*)  ( XCC(I)*.01, I=1,NYS)                             W10.9
        WRITE(10,*)  ( FCP(I),      I=1,NYS)                            W10.10
        END IF

C1370 PREPARE FC' OF SIZE NSEC2 (ASSUMING TRIANGLE ELEMENTS) *****
      DX        =1./NSEC
      DXH       =DX*50.
      OT        =DX/.03
      NSEC2     =NSEC+NSEC
      DO    ISEG=1,NSEC2,2
```

```
           X        =(ISEG-1)*DXH+OT
           XCC(ISEG)       =X
           XCC(ISEG+1)     =X+OT
           END DO

         IF (INCAM.EQ.3) THEN
C1380      USE FORMULA TO FIND FC'
           WRITE(6,1382)
 1382      FORMAT(//' Fc'' OBTAINED FROM FORMULA FOR NACA a=1.0')

           CALL     FFCP( NSEC2, XCC, FCP)

           DO     N=2,NFR,2
             V(N-1)=0.
             V(N)  =.5/(N-1)
             END DO
           IF ( MOD(NFR,2).NE.0 )      V(NFR)=0.

           CB       =1.

          ELSE
C1390      INCAM = 1 OR 2
C1392      FIND CB

           CALL     FILONC( NYS-1, FCP, NFR, V)

           CB       =V(2)+V(2)
           WRITE(6,1395) CB
 1395      FORMAT(' CB ='F8.5)
C1396      PREPARE FCP

           CALL     SPLIN1( NYC, XC, YC, NSEC2, XCC, DUMMY, FCP, DUM,
     +                      SEND1, 1.E-9)

          END IF

         IF (IFLAT.EQ.1) THEN
           ITEM     =NSEC2*.15
           DO   I=1,ITEM
             FCP(I)         =MIN( FCP(I), FCPA)
             END DO
           DO   I=NSEC2-ITEM,NSEC2
             FCP(I)         =MAX( FCP(I), FCPZ)
             END DO
           END IF

         IF (INCAM.NE.3) THEN
           DO     I=1,NSEC2
             FCP(I)=FCP(I)/CB
             END DO
           DO     N=1,NFR
             V(N)  =V(N)/CB
             END DO

CCC        IF (INCAM.EQ.1) GOTO ??? ??? ???

           IF (CBID.LT.5) THEN
             Q      =CBID/CB
             CB     =CB*Q
             WRITE(6,1398) CB, Q
 1398        FORMAT(/' ... SYMMETRIC CAMBER ...    Cb ='F6.3
     +             '   MULTIPLE FACTOR ='F6.2)
             END IF
          END IF

         IF (MH.EQ.2) THEN
CCC ??? TO BE CHECKED   ???
           Q        =-CB/PI
           DO     N=1,NFR
             V(N)  =V(N)*Q
             END DO
           V(1)     =SIND(ALPHAM)+V(1)
           END IF

C1400 THICKNESS DATA *****
         IF (INCAM.EQ.1) THEN
           DO     I=1,NSEC2
             FTP(I)=0.
             END DO
           TOC      =0.
```

```
            TOC2PI  =0.
            WRITE(6,1410)
     1410   FORMAT(//' THICKNESS = 0'//)
           ELSE
            READ (11,5101) TITL                                      READ1105
            WRITE(6,5103)  TITL
            READ (11,*) NYT                                          READ1106
            IF (NYT.GT.NYMAX) THEN
              PRINT *, 'NYT.GT.NYMAX...', NYT, NYMAX
              STOP 'INPUT.1415'
              END IF
            READ (11,*) ( XT(I), I=1,NYT)                            READ1107
            READ (11,*) ( YT(I), I=1,NYT)                            READ1108
            IF (LIST.GT.0) THEN
              WRITE(6,1440)
     1440     FORMAT(/21X'X        Yt'/)
              DO   I=1,NYT
                WRITE(6,6222) XT(I), YT(I)
                IF ( MOD(I,5).EQ.0 )      WRITE(6,6222)
                END DO
              END IF
            TOC     =0.
            DO      I=1,NYT
              TOC   =MAX( YT(I), TOC)
              END DO
            TOC     =TOC*.01
            CALL    SPLIN1( NYT, XT, YT, NSEC2, XCC, DUMMY, FTP, DUM,
     +                      SEND1, 1.E-9)
            DO      I=1,NSEC2
              FTP(I)=FTP(I)/TOC
              END DO
            TOC     =TOC+TOC
    C       TOTAL THICKNESS IS TWICE THE INPUT DATA
            WRITE(6,1430)TOC
     1430   FORMAT(/' MAXIMUM BLADE THICKNESS ='F8.5//)
            TOC2PI  =TOC*PI2
           END IF
          READ (11,*) JACK                                           READ1109
          IF (JACK.NE.999) THEN
            PRINT *, 'CHECK BLADE DATA...', JACK
            STOP 'INPUT.1500'
            END IF
          CLOSE(11)
    C2000 CONSTANT
          CALPHA    =COSD(ALPHAM)
          SALPHA    =SIND(ALPHAM)
          SA2PI     =PI2*SALPHA
    C3000 PREPARE DATA FILE MELO10.DAT (CONT.)
          IF (LISTO.EQ.11) THEN
            WRITE(10,*) 'TOC'                                        W10.11
            WRITE(10,*)  TOC                                         W10.12
            WRITE(10,3120)                                           W10.13
     3120   FORMAT(' FOLLOWING ARE ( ( G(N,K), K=1,NFR), N=1,NFR),'/
         +  ' ( ( H(N,K), K=1,NFR), N=1,NFR), ( B(N), N=1,NFR), &'/
         +  ' ( T(N), N=1,NFR), FOR AMDAD=AMDA,AMDZ,AMDI')
            WRITE(6,3130)                                            W10.14
     3130   FORMAT(//
         +  ' A TABLE OF G, H, B, & T FUNCTIONS IS CREATED IN "MELO10.DAT"'/
         +  /)
            END IF
          RETURN
    C5000 FORMAT
     5101 FORMAT(A90)
     5103 FORMAT(/1XA90)
     6222 FORMAT(F23.2,F9.3,F11.5)
          END
```

PROGRAM SECTIONS

Name   Bytes   Attributes

0	$CODE	5506	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
1	$PDATA	1488	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT	LONG
2	$LOCAL	10492	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	LONG
3	MIL	9	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
4	DGI	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
5	DGILR	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
6	DI	8012	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
7	DIL	4	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
8	GI	4872	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
9	GILR	4	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
10	IL	13840	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
11	ILR	16	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
12	IR	12	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
	Total Space Allocated	44271									

ENTRY POINTS

Address	Type	Name	References
0-00000000		INPUT	1#

VARIABLES

Address	Type	Name	Attributes	References							
11-00000000	R*4	ALPHAM	COMM	171	200=	201	469	528	529		
10-000035D0	R*4	AMDA	COMM	167	246=	247	275				
10-000035D8	R*4	AMDI	COMM	167	246=	247	275				
10-000035D4	R*4	AMDZ	COMM	167	246=	247	275				
10-000035DC	R*4	CA	COMM	167	211=	212					
11-00000004	R*4	CALPHA	COMM	171	528=						
11-00000008	R*4	CB	COMM	171	416=	424=	425	446	449	455	456(2)=
				457	465						
2-00002658	R*4	CBID		184=	185	454	455				
10-000035E0	R*4	CC	COMM	167	211=	212					
9-00000000	R*4	CSA	COMM	166	204=	211=	212	237=	238	240	263
				264	275						
10-000035E4	R*4	CSI	COMM	167	205=	211=	212	239=			
10-000035E8	R*4	CSZ	COMM	167	206=	211=	212	240=	263	264	
**	R*4	DTHETA		327=	329						
2-00002690	R*4	DUM		303A	357A	429A	507A				
4-00000000	R*4	DX	COMM	161	393=	394	395				
**	R*4	DXH		394=	398						
2-0000266C	R*4	EPS		246=							
2-0000269C	R*4	FCPA		365=	368	437					
2-000026A0	R*4	FCPZ		366=	371	440					
**	I*4	I		224(2)=	225(2)=	226(2)=	227(2)=	299(2)=	300(2)=	308=	309(3)
				310	328=	329(2)	331=	332(2)	334=	335(2)	367=
				368(2)	370=	371(2)	388(2)=	389(2)=	436=	437(2)	439=
				440(2)	445=	446(2)	474=	475	489(2)=	490(2)=	495=
				496(2)	497	502=	503	510=	511(2)		
**	I*4	IA		228(2)=							
2-00002664	I*4	IB		228(2)=							
**	I*4	IC		228(2)=							
2-00002650	I*4	IFLAT		183=	185	338	363	434			
2-0000264C	I*4	INCAM		183=	185	276(2)	277	285	351	403	444
				473							
12-00000000	I*4	IRES	COMM	172	207=	210=	212				
2-00002660	I*4	IS		228(2)=							
10-000035F8	I*4	ISCL	COMM	167	210=	212					
**	I*4	ISEG		397=	398	399	400				
**	I*4	ITEM		364=	367	370	435=	436	439		
2-00002668	I*4	JACK		228=	230	231	253=	255	256	315=	316
				317	520=	521	522				
**	I*4	KA		253(4)=							
2-00002674	I*4	KAZ		247=	249	250	253(2)				
**	I*4	KX		253(4)=							
**	I*4	KXZ		248=	249	250	253(2)				
2-00002654	I*4	LIST		183=	185	302	492				
5-00000000	I*4	LISTO	COMM	162	199=	201	241=	262	264	277	379

```
                                                     533
10-000035FC  I*4  MH          COMM          167      183=      185      191      193      197      463
     **      I*4  N                         383(2)=  410=      411      412(2)   448=     449(2)   466=     467(2)
10-00003600  I*4  NA          COMM          167      216=      217      219      221      227      228

10-00003604  I*4  NB          COMM          167      216=      217      219      221      224      228
10-00003608  I*4  NC          COMM          167      216=      217      219      221      226      228
 5-00000004  I*4  NFR         COMM          162      183=      185      191(2)   193      273      381A     383
                                            410      414(2)    422A     448      466
10-0000360C  I*4  NS          COMM          167      216=      217      219      221      225      228
 4-00000004  I*4  NSEC        COMM          161      183=      185      191      193      321      393      396(2)

 6-00001F48  I*4  NSEC2       COMM          163      396=      397      408A     429A     435      439(2)   445
                                            474      507A      510
 2-0000268C  I*4  NYC                       283=     294=      295      296      299      300      303(2)A  308
                                            321      357A      429A
     **      I*4  NYS                       321=     322(3)=   323      324      327      328      331      334
                                            339      344       364      370(2)   381      386      388      389
                                            422
 2-00002694  I*4  NYSP1                     339=     340       341      365
 2-00002698  I*4  NYSP2                     340=     342       344=     346      347      353A     357A     366

 2-00002680  I*4  NYT                       484=     485       486      489      490      495      502      507A
     **      R*4  OT                        395=     398       400
 7-00000000  R*4  PI          COMM          164      177D      284(2)   327      465
11-0000000C  R*4  PI2         COMM          171      284=      517      530
     **      R*4  Q                         455=     456       457      465=     467

12-00000004  R*4  SA2PI       COMM          172      530=
10-000035EC  R*4  SALPHA      COMM          167      529=      530
 2-0000265C  R*4  SEND1                     184=     185       303A     357A     429A     507A
10-000035F0  R*4  SEND5       COMM          167      211=      212
 3-00000000  CHAR TDATE       COMM          153      160

 2-00002688  R*4  TDATEW                    268
 2-000025F0  CHAR TITL                      153      180=      181=     182      244=     245      292=     293
                                            482=     483
10-000035F4  R*4  TOC         COMM          167      477=      501=     503(2)=  505(2)=  511      513(3)=  515
                                            517      535
12-00000008  R*4  TOC2PI      COMM          172      478=      517=
     **      R*4  X                         398=     399       400

 8-00001300  R*4  XSA         COMM          165      246=      248
 8-00001304  R*4  XSI         COMM          165      246=      248
 2-00002670  R*4  XSZ                       246=     248
```

ARRAYS

```
   Address   Type Name        Attributes       Bytes  Dimensions      References

 8-00000000  R*4  AI          COMM              2432  (76, 8)          165      253=
 2-00000000  R*4  ALPHA1                          76  (19)             174      227=
10-00003570  R*4  BETA1       COMM                16  (4)              167      224=
10-00003580  R*4  CB1         COMM                20  (5)              167      226=
 2-0000004C  R*4  DUMMY                         4008  (1002)           174      357A     429A     507A

 6-00000000  R*4  FCP         COMM              4008  (1002)           163      303A     309      353A     357A
                                                                       365      366      368(2)=  371(2)=  381A
                                                                       389      408A     422A     429A     437(2)=
                                                                       440(2)=  446(2)=
 6-00000FA8  R*4  FTP         COMM              4000  (1000)           163      475=     507A     511(2)=
10-00000000  I*4  ICL         COMM             13680  (19, 9, 5, 4)    167      228=
 8-00000980  R*4  R           COMM              2432  (76, 8)          165      253=
 2-00000FF4  R*4  SIG                             20  (5)              174      225=

10-00003594  R*4  V           COMM                60  (15)             167      381A     383      411=     412=
                                                                       414=     422A     424(2)   449(2)=  467(2)=
                                                                       469(2)=
 2-00001008  R*4  XC                             400  (100)            174      299=     303(2)A  309      357A
                                                                       429A
 2-00001198  R*4  XCC                           4008  (1002)           174      329=     332(2)=  335(2)=  341=
                                                                       342=     353A     357A     388      399=
                                                                       400=     408A     429A     507A
 2-00002140  R*4  XT                             400  (100)            174      489=     496      507A
 2-000022D0  R*4  YC                             400  (100)            174      300=     303(2)A  309      357A
                                                                       429A

 2-00002460  R*4  YT                             400  (100)            174      490=     496      503      507A
```

PARAMETER CONSTANTS

Type	Name	References						
I*4	KAMX	155#	165(2)	249	250			
I*4	KXMX	155#	165(2)	249	250			
I*4	NAMX	155#	167	174	219	221		
I*4	NBMX	155#	167(2)	219	221			
I*4	NCMX	155#	167	219	221			
I*4	NFRMX	155#	167	191	193			
I*4	NSECMX	155#	157(3)	191	193			
I*4	NSMX	155#	167(2)	174	219	221		
I*4	NYMAX	155#	157(2)	174(4)	295	296	485	486
I*4	NYSMX	157#	163	346	347			
I*4	NYSMX2	157#	163	174(2)	323	324		

LABELS

Address	Label	References					
1-00000221	120'	185	187#				
**	130	197	199#				
1-000002A1	135'	201	202#				
0-00000278	140	197	210#				
1-000002E0	150'	212	213#				
1-0000031E	160'	217	218#				
0-00000634	170	197	237#				
0-0000068E	210	208	235	244#			
1-00000337	260'	268	269#				
0-00000A2D	1110	285	287#				
0-00000A54	1210	285	292#				
1-0000037B	1220'	306	307#				
0-00000C1D	1310	285	315#				
1-00000399	1340'	373	374#				
1-000003EA	1365'	383	384#				
1-00000420	1382'	405	406#				
1-0000044E	1395'	425	426#				
1-00000459	1398'	457	458#				
1-0000049A	1410'	479	480#				
1-000004C0	1430'	515	516#				
1-000004AF	1440'	493	494#				
1-000004E3	3120'	536	537#				
1-00000579	3130'	540	541#				
1-0000058D	5101'	180	181	244	292	482	549#
1-000005C0	5103'	182	245	293	483	550#	
1-000005C6	6222'	309	310	496	497	551#	

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References			
	FFCP	353	408		
	FILONC	381	422		
	FOR$CLOSE	234	259	319	525
	FOR$OPEN	215	267		
R*4	MTH$COS	332			
R*4	MTH$COSD	528			
R*4	MTH$SIND	469	529		
	SPLIN1	303	357	429	507

```
+--------------------------------------------+
|          KEY TO REFERENCE FLAGS            |
|   =   - Value Modified                     |
|   #   - Defining Reference                 |
|   A   - Actual Argument, possibly modified |
|   D   - Data Initialization                |
|  (n)  - Number of occurrences on line      |
+--------------------------------------------+
```

```
      SUBROUTINE LOOP

C (3037-MELLOR-LOOP) (L) LOOP THRU VARIOUS PARAMETERS TO FIND SOLUTIONS
C REFERENCED BY MAIN (MELLOR); REFERENCES GNK (TWICE), RESULT (TWICE),
C       SIMUL1 (TWICE), & SPLIN1 (4 TIMES), & FORSOPEN, MTH$AMOD,
C       MTH$COSD (TWICE), & MTH$SIND (TWICE)
C INPUT AI, AMDA, AMDI, AMDZ, BETA1, C, CA, CALPHA, CB, CB1, CBEFF,
C       CC, CSA, CSI, CSZ, ICL, LISTO, MH, NB, NC, NFR, PI, PI2, R,
C       SA2PI, SALPHA, TOC & V
C CODED BY W. CHIANG
C 10OCT86, S. HSU TRANSLATED FROM (3037-CASCAD[HP9845]-LOOP) WHICH WAS
C       SEPARATED FROM MAIN OF (3037-CASCAD[HP9845]), 26AUG85, AND
C       REVISED NOV85
C REVISED 27FEB87, CHIANG; TOUCHED 01APR87

      CHARACTER TDATE*9

      PARAMETER ( NALMX=13, NAMX=19, NBMX=4, NCMX=9, NFRMX=15, NSMX=5)

      COMMON/M  IL /    TDATE
      COMMON/ DGILR/    LISTO, NFR
      COMMON/ DG L /    C(NFRMX)
      COMMON/ D IL /    PI
      COMMON/  GILR/    CSA
      COMMON/  G LR/    G(NFRMX,NFRMX), H(NFRMX,NFRMX),
     +                  B(NFRMX), T(NFRMX),             AMDAD, CS
      COMMON/   IL /    ICL(NAMX,NCMX,NSMX,NBMX),
     +                  BETA1(NBMX), CB1(NSMX), V(NFRMX),
     +                  AMDA, AMDZ, AMDI, CA, CC, CSI, CSZ, SALPHA,
     +                  SEND5, TOC,          ISCL, MH, NA, NB, NC, NS
      COMMON/   ILR/    ALPHAM, CALPHA, CB, PI2
      COMMON/    LR/    A(NFRMX),                CB0

      DIMENSION GHV( NFRMX*(NFRMX+1) ),
     + ALFAM(NALMX), B1(NALMX), BCTTT(NFRMX), BT(NFRMX), CL(NALMX),
     + CL1(NALMX), DTH(NALMX), DUMMY(NALMX), TT(NFRMX), VE(NFRMX),
     + VEC(NFRMX)

      NFR1      =NFR+1
      NFRSQ     =NFR*NFR
      NFRSQP    =NFRSQ+1
      Q         =-1./PI
      DO      N=1,NFR
        VE(N)   =V(N)*Q
        END DO
      IF (LISTO.GT.0) THEN
        DO      N=1,NFR
          VE(N) =VE(N)*CALPHA
          END DO
        END IF

      IF (MH.EQ.2) GOTO 1310

      GOTO ( 110, 120, 130) LISTO

  110 WRITE(6,112) C(1), C(2)
  112 FORMAT(//
     + ' LAMDA C/S   G00   G02   G11   G20   G22   H00   H01   H10',
     + '    B0    B1    T0    T1'/' (DEG)'/
     + '           0 1.000     0-1.000     0-1.000     0     0     0'
     + 2F6.3'     0     0')
      GOTO 310

  120 WRITE(6,122)
  122 FORMAT(//' LAMDA  CB    C/S    A0A    A0C    A0T    A1A    A1C  ',
     + '  A1T     AA     AC     AT'/' (DEG)'/)
      GOTO 310

  130 WRITE(6,132)
  132 FORMAT(//' LAMDA  CB    C/S      CL      (CL)      A0      A1 ',
     + '        A2        A3        A4'/' (DEG)'/)

C 200 LOOP TO PRODUCE MELLOR'S TABLE+OPTIONAL USING ALPHAM AS A PARAMETER ******

C 300 LOOP THRU STAGER ANGLES *****
  310 DO750AMDAD=AMDA,AMDZ,AMDI
        KA      =INT( (AMDAD-AMDA)/AMDI+.000001 ) + 1
```

```
C 400 LOOP THRU CB *****
        IF (LISTO.EQ.2 .OR. LISTO.EQ.3) THEN
          CBO   =0.
         ELSE
          CBO   =1.
         END IF
        DO 750CB=CBO,1.,1.
          CBTOC =-CB*TOC
          DO   N=1,NFR
            VEC(N)      =VE(N)*CB
            END DO
          IF (LISTO.NE.0)      VEC(1)=VEC(1)+SALPHA
C 500     LOOP THRU SOLIDITIES *****
          DO   750    CS=CSA,CSZ,CSI
            CALL        GNK(KA)
            IF ( MOD(LISTO,10) .EQ.1 ) GOTO 750
            KN  =0
            DO N=1,NFR
              BCTTT(N)  =B(N)*CBTOC-T(N)*TOC
              DO        K=1,NFR
                GHV(KN+K)       =G(N,K)-H(N,K)*CB
                END DO
              KN=KN+NFR
              END DO
CCC         IF (LISTO.EQ.0) THEN
CCC           DO 720ALPHAM= 15.-AMDAD, 70.-AMDAD, 5.
CCC             SALPHA  =SIND(ALPHAM)
CCC             SA2PI   =SALPHA*PI2
CCC             CALPHA  =COSD(ALPHAM)
CCC           DO      N=1,NFR
CCC             GHV(N+NFRSQ)    =VEC(N)*CALPHA + BCT(N) + TT(N)
CCC             END DO
CCC           GHV(NFRSQP)       =GHV(NFRSQP)+SALPHA
CCC          ELSE
              DO        N=1,NFR
                GHV(N+NFRSQ)    =VEC(N)+BCTTT(N)
                END DO
CCC          END IF
C 600         CALC. FOURIER COEFS. **********
              CALL      SIMUL1( 3, NFR, GHV, A, 1.E-9, DETER)
              IF (DETER.LT..1) PRINT *, 'DETER < .1 ???'
C 700         CALC. VORTICITY GAMMA & LIFT COEF. CL **********
              CALL      RESULT( DUM, CL, DUM, DUM)
CCC           IF (LISTO.EQ.0) THEN
CCC720          CONTINUE
CCC             END IF
  750       CONTINUE
C 750     CONTINUE
C 750   CONTINUE
       IF (LISTO.EQ.11) THEN
         WRITE(10,*) 999
         END IF
       GO TO 3900
C1200 LOOP TO PRODUCE DATA TO BE COMPARED W/HERRIG'G **********
C1300 LOOP THRU SOLIDITIES *****
 1310 DO 1850 CS=CSA,CSZ,CSI
        IS      =INT( (CS-CSA)/CSI + 1.1 )
        WRITE(6,1320)
 1320   FORMAT(//'  C/S LAMDA    CB ALPHA(M)    CL(M)   DTHETA  BETA(M)',
     +          '  BETA(1) ALPHA(1)    CL(1)'/'       (DEG)'/)
        WRITE(6,6330) CS
C1400   LOOP THRU LAMDA *****
        DO  1850   AMDAD=AMDA,AMDZ,AMD
```

```
          IF ( AMDAD-AMDA .GT. 0.)      WRITE(6,6410) AMDAD

          CALL  GNK( INT( (AMDAD-AMDA)/AMDI+.000001 ) + 1 )

          DO   N=1,NFR
            BT(N)        =-B(N)*TOC
            TT(N)        =-T(N)*TOC
            END DO

C1500     LOOP THRU CB *****
          DO   1850   IC=1,NC
            IF (IC.GT.1)        WRITE(6,6510) CB1(IC)
            CB  =CB1(IC)*CC
            KN  =0
            DO N=1,NFR
              BCTTT(N)  =BT(N)*CB+TT(N)
              VEC(N)     =VE(N)*CB
              DO        K=1,NFR
                GHV(KN+K)        =G(N,K)-H(N,K)*CB
                END DO
              KN         =KN+NFR
              END DO

C1600       LOOP THRU ALPHAM *****
            ALFAM(1)     =20.-AMDAD
            KIA =0
            IA  =1
 1610       ALPHAM       =ALFAM(IA)
            ALPHA        =ALPHAM*CA
            IF (ABS(ALPHA).GT.90.) THEN
              PRINT *, 'ERROR... ABS(ALPHA).GT.90...', ALPHA
              STOP 'LOOP.1610'
              END IF

            SALPHA       =SIND(ALPHA)
            CALPHA       =COSD(ALPHA)
            DO N=1,NFR
              GHV(N+NFRSQ)        =VEC(N)*CALPHA+BCTTT(N)
              END DO
            GHV(NFRSQP) =GHV(NFRSQP)+SALPHA

C1700       CALC. FOURIER COEFS. **********

            CALL         SIMUL1( 3, NFR, GHV, A, 1.E-9, DETER)

            IF (DETER.LT..1) PRINT *, 'DETER < .1 ???'

C1800       CALC. VORTICITY GAMMA & LIFT COEF. CL **********

            CALL         RESULT( B1, CL, CL1, DTH)

            IF (IA.GT.1) THEN
              IF (B1(IA).GT.70.) GOTO 1840
              IF (IA.NE.3) GOTO 1830
              IF (B1(3).GT.30.) GOTO 1830
              ALFAM(3)  =ALFAM(3) - INT( B1(3) ) + 32.
              GOTO 1610
              END IF
            IF (B1(1).LT.30. .AND. B1(1).GT.24.) GOTO 1830
            KIA =KIA+1
            IF (KIA.LE.9) THEN
              ALFAM(1)  =ALFAM(1) - INT( B1(1) ) + 28.
              GOTO 1610
              END IF
CCC  ??  ERROR BELOW
            IF (B1(1).GT.24.)       ALFAM(1)=ALFAM(1)-1
            IF (B1(1).LT.30.)       ALFAM(1)=ALFAM(1)+1
            GOTO 1610

 1830       ALFAM(IA+1) =ALFAM(IA)+5.
            IA  =IA+1
            GOTO 1610

 1840       CALL         SPLIN1( IA, B1, ALFAM, NB, BETA1, ALFAM, DUMMY,
     +                          DUM, SEND5, 1.E-6)

            CALL         SPLIN1( IA, B1,    CL, NB, BETA1,    CL, DUMMY,
     +                          DUM, SEND5, 1.E-6)

            CALL         SPLIN1( IA, B1,   CL1, NB, BETA1,   CL1, DUMMY,
```

```
      +                         DUM, SEND5, 1.E-6)

            CALL        SPLIN1( IA, B1,   DTH, NB, BETA1,   DTH, DUMMY,
      +                         DUM, SEND5, 1.E-6)

CCC            KIB      =0
            DO  1850  IB=1,NB
              ALPHA1    =BETA1(IB)-AMDAD
              IF ( AMOD( ALPHA1, 2.) .NE. 0.) GOTO 1850
              IA=ALPHA1/2+3
              IF (IA.GT.19 .OR. IA.LT.1) GOTO 1850
CCC              IF (KIB.NE.0)      WRITE(6,6850)
CCC              KIB    =1
              WRITE(6,6860) ALFAM(IB), CL(IB), DTH(IB),
      +         ALFAM(IB)+AMDAD, BETA1(IB), ALPHA1, CL1(IB)
              IF ( ICL(IA,IC,IS,IB).LT.32760 ) THEN
                PRINT *, 'ERROR... ICL(IA,IC,IS,IB).LT.32760...',
      +                 ICL(IA,IC,IS,IB), IA, IC, IS, IB
                STOP 'LOOP.6870'
                END IF
              ICL(IA,IC,IS,IB)  =CL1(IB)*10000
 1850         CONTINUE
C1850       CONTINUE
C1850     CONTINUE
C1850   CONTINUE

        IF (ISCL.GT.0) THEN
          OPEN( 8, FILE='MELO8', STATUS='NEW')
          WRITE(8,1910) TDATE, NB, NS, NC, NA
 1910     FORMAT(' (3037-MELLOR-MELO8) CALCULATED CL TO BE COMPARED W/ ',
      +   'HERRIG, ET AL, 1951',T71,A9/'   NB,  NS,  NC,  NA  :'/415/
      +   ' ( ( ( ( ICL(IA,IC,IS,IB), IA=1,NA), IC=1,NC), IS=1,NS), ',
      +   'IB=1,NB), 999  :')
          WRITE(8,*) ( ( ( ( ICL(IA,IC,IS,IB), IA=1,NA), IC=1,NC),
      +                 IS=1,NS), IB=1,NB), 999
          END IF

 3900 RETURN

C5000 FORMAT
 6330 FORMAT(1XF4.2)
 6410 FORMAT(/F11.0)
 6510 FORMAT(F16.2)
 6860 FORMAT(4(F8.3)2(F9.2)F8.3)
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	2808	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	821	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	2000	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3	MIL	9	PIC OVR REL GBL SHR NOEXE RD WRT LONG
4	DGILR	8	PIC OVR REL GBL SHR NOEXE RD WRT LONG
5	DGL	60	PIC OVR REL GBL SHR NOEXE RD WRT LONG
6	DIL	4	PIC OVR REL GBL SHR NOEXE RD WRT LONG
7	GILR	4	PIC OVR REL GBL SHR NOEXE RD WRT LONG
8	GLR	1928	PIC OVR REL GBL SHR NOEXE RD WRT LONG
9	IL	13840	PIC OVR REL GBL SHR NOEXE RD WRT LONG
10	ILR	16	PIC OVR REL GBL SHR NOEXE RD WRT LONG
11	LR	64	PIC OVR REL GBL SHR NOEXE RD WRT LONG
	Total Space Allocated	21562	

ENTRY POINTS

Address	Type	Name	References
0-00000000		LOOP	1#

VARIABLES

Address	Type	Name	Attributes	References				
**	R*4	ALPHA		186=	187	188	192	193
2-0000066C	R*4	ALPHA1		245=	246	247	251	
10-00000000	R*4	ALPHAM	COMM	31	185=	186		

9-000035D0	R*4	AMDA	COMM	27	76	77	157	158	160		
8-00000780	R*4	AMDAD	COMM	25	76=	77	157=	158(2)	160	182	245
				251							
9-000035D8	R*4	AMDI	COMM	27	76	77	157	160			
9-000035D4	R*4	AMDZ	COMM	27	76	157					
9-000035DC	R*4	CA	COMM	27	186						
10-00000004	R*4	CALPHA	COMM	31	48	193=	195				
10-00000008	R*4	CB	COMM	31	85=	86	88	102	170=	173	174
				176							
11-0000003C	R*4	CBO	COMM	32	81=	83=	85				
**	R*4	CBTOC		86=	100						
9-000035E0	R*4	CC	COMM	27	170						
8-00000784	R*4	CS	COMM	25	93=	149=	150	154			
7-00000000	R*4	CSA	COMM	24	93	149	150				
9-000035E4	R*4	CSI	COMM	27	93	149	150				
9-000035E8	R*4	CSZ	COMM	27	93	149					
2-00000644	R*4	DETER		124A	126	201A	203				
2-00000648	R*4	DUM		130(3)A	231A	234A	237A	240A			
2-00000664	I*4	IA		184=	185	209	210	211	227(2)	228(2)=	231A
				234A	237A	240A	247=	248(2)	253	254(2)	258
				271(2)=							
**	I*4	IB		244=	245	251(6)	253	254(2)	258(2)	271(2)=	
**	I*4	IC		168=	169(2)	170	253	254(2)	258	271(2)=	
2-00000654	I*4	IS		150=	253	254(2)	258	271(2)=			
9-000035F8	I*4	ISCL	COMM	27	264						
**	I*4	K		101=	102(3)	175=	176(3)				
2-00000634	I*4	KA		77=	95A						
**	I*4	KIA		183=	217(2)=	218					
**	I*4	KN		98=	102	104(2)=	171=	176	178(2)=		
4-00000000	I*4	LISTO	COMM	21	46	54	80(2)	90	97	140	
9-000035FC	I*4	MH	COMM	27	52						
**	I*4	N		43=	44(2)	47=	48(2)	87=	88(2)	99=	100(3)
				102(2)	117=	118(3)	162=	163(2)	164(2)	172=	173(3)
				174(2)	176(2)	194=	195(3)				
9-00003600	I*4	NA	COMM	27	266	271					
9-00003604	I*4	NB	COMM	27	231A	234A	237A	240A	244	266	271
9-00003608	I*4	NC	COMM	27	168	266	271				
4-00000004	I*4	NFR	COMM	21	39	40(2)	43	47	87	99	101
				104	117	124A	162	172	175	178	194
				201A							
**	I*4	NFR1		39=							
2-00000624	I*4	NFRSQ		40=	41	118	195				
2-00000628	I*4	NFRSQP		41=	197(2)						
9-0000360C	I*4	NS	COMM	27	266	271					
6-00000000	R*4	PI	COMM	23	42						
10-0000000C	R*4	PI2	COMM	31							
**	R*4	Q		42=	44						
9-000035EC	R*4	SALPHA	COMM	27	90	192=	197				
9-000035F0	R*4	SEND5	COMM	27	231A	234A	237A	240A			
3-00000000	CHAR	TDATE	COMM	16	20	266					
9-000035F4	R*4	TOC	COMM	27	86	100	163	164			

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
11-00000000	R*4	A	COMM	60	(15)	32	124A	201A		
2-000003C0	R*4	ALFAM		52	(13)	34	182=	185	213(2)=	219(2)=
						223(2)=	224(2)=	227(2)=	231(2)A	251(2)
8-00000708	R*4	B	COMM	60	(15)	25	100	163		
2-000003F4	R*4	B1		52	(13)	34	207A	210	212	213
						216(2)	219	223	224	231A
						234A	237A	240A		
2-00000428	R*4	BCTTT		60	(15)	34	100=	118	173=	195
9-00003570	R*4	BETA1	COMM	16	(4)	27	231A	234A	237A	240A
						245	251			
2-00000464	R*4	BT		60	(15)	34	163=	173		
5-00000000	R*4	C	COMM	60	(15)	22	56(2)			
9-00003580	R*4	CB1	COMM	20	(5)	27	169	170		

2-000004A0	R*4	CL		52	(13)	34	130A	207A	234(2)A	251
2-000004D4	R*4	CL1		52	(13)	34	207A	237(2)A	251	258
2-00000508	R*4	DTH		52	(13)	34	207A	240(2)A	251	
2-0000053C	R*4	DUMMY		52	(13)	34	231A	234A	237A	240A
8-00000000	R*4	G	COMM	900	(15, 15)	25	102	176		
2-00000000	R*4	GHV		960	(240)	34	102=	118=	124A	176=
						195=	197(2)=	201A		
8-00000384	R*4	H	COMM	900	(15, 15)	25	102	176		
9-00000000	I*4	ICL	COMM	13680	(19, 9, 5, 4)	27	253	254	258=	271
8-00000744	R*4	T	COMM	60	(15)	25	100	164		
2-00000570	R*4	TT		60	(15)	34	164=	173		
9-00003594	R*4	V	COMM	60	(15)	27	44			
2-000005AC	R*4	VE		60	(15)	34	44=	48(2)=	88	174
2-000005E8	R*4	VEC		60	(15)	34	88=	90(2)=	118	174=
						195				

PARAMETER CONSTANTS

Type	Name	References					
I*4	NALMX	18#	34(6)				
I*4	NAMX	18#	27				
I*4	NBMX	18#	27(2)				
I*4	NCMX	18#	27				
I*4	NFRMX	18#	22	25(6)	27	32	34(7)
I*4	NSMX	18#	27(2)				

LABELS

Address	Label	References						
0-0000009E	110	54	56#					
1-0000007B	112'	56	57#					
0-000000D0	120	54	64#					
1-0000012D	122'	64	65#					
0-000000E8	130	54	69#					
1-0000018E	132'	69	70#					
0-000000FE	310	62	67	76#				
0-0000035B	750	76	85	93	97	136#		
0-000003B0	1310	52	149#					
1-000001EF	1320'	151	152#					
0-00000660	1610	185#	214	220	225	229		
0-000009C0	1830	211	212	216	227#			
0-00000794	1840	210	231#					
0-00000967	1850	149	157	168	244	246	248	259#
1-00000254	1910'	266	267#					
0-00000AF7	3900	144	275#					
1-00000312	6330'	154	278#					
1-00000318	6410'	158	279#					
1-0000031D	6510'	169	280#					
1-00000321	6860'	251	281#					

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References			
	FOR$OPEN	265			
	GNK	95	160		
R*4	MTH$AMOD	246			
R*4	MTH$COSD	193			
R*4	MTH$SIND	192			
	RESULT	130	207		
	SIMUL1	124	201		
	SPLIN1	231	234	237	240

```
      SUBROUTINE RESULT( BETA1, CL, CL1, DTH)

C (3037-MELLOR-RESULT) (R) CALC. RESULTS FROM CASCADE THEORY, SUCH AS
C                           VORTICITY(GAMMA), LIFT COEF.(MELLOR,1959)
C REFERENCED BY LOOP
C REFERENCES MTH$ATAND (THRICE), MTH$COSD (TWICE), MTH$SIND (TWICE), &
C       MTH$TAND
C INPUT A, IRES,  & PI2, AND OUTPUT CL, IF IRES=1
C INPUT A, ALPHAM, AMDAD, CS, IRES, & PI2, AND OUTPUT BETA1, CL, CL1, &
C       DTH, IF IRES=2
C INPUT A, AMDAD, B, CALPHA, CB, CS, G, H, IRES, LISTO, PI2, SA2PI, T,
C       TOC2PI, AND OUTPUT NONE, IF IRES=3
C CODED BY W. CHIANG, 14JUL83
C 10OCT86, S. HSU TRANSLATED FROM (3037-CASCAD[HP9845]-RESULT) WHICH,
C       REVISED 07NOV85, WAS ADAPTED FROM (3725-CASCAD[HP9845]-RESULT),
C       05AUG85
C REVISED 17NOV86, CHIANG; TOUCHED 01DEC86

      PARAMETER ( NFRMX=15)

      COMMON/DGILR/     LISTO, NFR
      COMMON/ GILR/     CSA
      COMMON/ G LR/     G(NFRMX,NFRMX), H(NFRMX,NFRMX),
     +                  B(NFRMX), T(NFRMX),            AMDAD, CS
      COMMON/  ILR/     ALPHAM, CALPHA, CB, PI2
      COMMON/  I R/     IRES,          SA2PI, TOC2PI
      COMMON/   LR/     A(NFRMX),            CB0

C 100 CALC. CL **********
      CL        =( A(1)+A(2) )*PI2
      GOTO ( 3900, 2100, 3100), IRES
      PRINT *, 'IRES SHOULD BE 1, 2, OR 3, NOT', IRES
      STOP 'RESULT.110'

C2000 (3037-MELLOR-RESULT-CL1) CALC. BETA1 & CL1 **********
C     CODED BY W. CHIANG, 12AUG85; TRANSLATED BY S. HSU, 10OCT86
C     REVISED 16OCT86, CHIANG
 2100 BETAM     =ALPHAM+AMDAD
      SINBM=SIND(BETAM)
      TEM=CL*CS*.25+SINBM
      IF (TEM.EQ.0.) THEN
        BETA1   =0.
        CL1     =0.
        DTH     =-ATAND( TAND(BETAM)*2 )
        IF ( COSD(BETAM).LT.0 )     DTH=DTH-180.
        GOTO 3900
        END IF
      IF (BETAM.EQ.90.) THEN
        BETA1   =BETAM
        DTH     =0.
       ELSE
        COSBM   =COSD(BETAM)
        TANB1   =TEM/COSBM
        BETA1   =ATAND(TANB1)
        DTH     =BETA1 - ATAND( (SINBM+SINBM)/COSBM-TANB1 )
        IF (COSBM.LT.0.) THEN
          BETA1 =BETA1+180.
          DTH   =DTH-180.
          END IF
       END IF

      TEM       =SIND(BETA1)/TEM
      CL1=TEM*TEM*CL
CCC   WRITE(6,7200) ALPHAM, CL, DTH, BETAM, BETA1, BETA1-AMDAD, CL1
      GOTO 3900

C3000 (3037-MELLOR-RESULT-SYMCAM) CALC. LIFT COEF. FOR A SYM. CAMBER,
C                                 USING A 3-TERM METHOD, SEE MELLOR (1959)
C     CODED BY W. CHIANG, 03AUG83
 3100 Q =-1./G(2,2)
      G33=G(3,3)
      P7=G(3,1)/G33
C     P0        =G(1,1) - G(1,3)*P7 - ( H(1,1)-H(1,3)*P7 )*CB
      P1=1-G(1,1)-(H(1,2)-(H(1,3)*CB-G(1,3))*H(3,1)/G33)*CB
      P3=B(3)*CB+T(3)
      P4=P3*CB*H(3,1)/G33
      P5=(H(2,3)*P7-H(2,1))*CB*Q
```

```
        Q9=Q*P1
        A0A=1/(P5*P1+G(1,1)-G(1,3)*P7-(H(1,1)-H(1,3)*P7)*CB)
        A0C=-A0A*CALPHA*P5
        T9=B(2)*CB+T(2)+P4
        A0T=(P3*P7-B(1)*CB-T(1)-T9*P5)*A0A
        A1A=Q9*A0A
        A1C=Q9*A0C+Q*CALPHA
        A1T=Q9*A0T+Q*T9
        AC=A0C+A1C
        AA=A0A+A1A
        AT=A0T+A1T

        GOTO ( 3900, 3200, 3300) LISTO
        PRINT *, 'ERROR... LISTO', LISTO
        STOP 'RESULT.3190'

 3200   IF (CS.EQ.CSA) THEN
          IF (CB.EQ.CBO) THEN
            WRITE(6,8200) AMDAD, CB, CS, A0A, A0C, A0T, A1A, A1C, A1T, AA,
     +                    AC, AT
           ELSE
            WRITE(6,8210) CB, CS, A0A, A0C, A0T, A1A, A1C, A1T, AA, AC, AT
           END IF
         ELSE
          WRITE(6,8220)       CS, A0A, A0C, A0T, A1A, A1C, A1T, AA, AC, AT
         END IF
        GOTO 3900

 3300   IF (CS.EQ.CSA) THEN
          IF (CB.EQ.CBO) THEN
            WRITE(6,8300) AMDAD, CB, CS, CL, AC*CB + AA*SA2PI + AT*TOC2PI,
     +                    ( A(N), N=1,MIN(NFR,11))
           ELSE
            WRITE(6,8310)       CB, CS, CL, AC*CB + AA*SA2PI + AT*TOC2PI,
     +                    ( A(N), N=1,MIN(NFR,11))
           END IF
         ELSE
          WRITE(6,8320)             CS, CL, AC*CB + AA*SA2PI + AT*TOC2PI,
     +                    ( A(N), N=1,MIN(NFR,11))
         END IF

 3900   RETURN

C7200   FORMAT(2(F9.0,2(F9.3))F9.0)
 8200   FORMAT(/F4.0,F7.2,F6.1,1X9(F7.3))
 8210   FORMAT(/    F11.2,F6.1,1X9(F7.3))
 8220   FORMAT(           F17.1,1X9(F7.3))
 8300   FORMAT(/F4.0,F6.2,F7.1,1X11(F9.3))
 8310   FORMAT(/    F10.2,F7.1,1X11(F9.3))
 8320   FORMAT(           F17.1,1X11(F9.3))
        END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes								
0	$CODE	1481	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
1	$PDATA	171	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT	LONG
2	$LOCAL	40	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	LONG
3	DGILR	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
4	GILR	4	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
5	GLR	1928	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
6	ILR	16	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
7	IR	12	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
8	LR	64	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
	Total Space Allocated	3724									

ENTRY POINTS

Address	Type	Name	References
0-00000000		RESULT	1#

VARIABLES

Address	Type	Name	Attributes	References							
**	R*4	A0A		79=	80	82	83	87	96	99	102
**	R*4	A0C		80=	84	86	96	99	102		
**	R*4	A0T		82=	85	88	96	99	102		
**	R*4	A1A		83=	87	96	99	102			
**	R*4	A1C		84=	86	96	99	102			
**	R*4	A1T		85=	88	96	99	102			
2-00000000	R*4	AA		87=	96	99	102	108	111	115	
**	R*4	AC		86=	96	99	102	108	111	115	
6-00000000	R*4	ALPHAM	COMM	25	38						
5-00000780	R*4	AMDAD	COMM	23	38	96	108				
2-00000004	R*4	AT		88=	96	99	102	108	111	115	
AP-00000004@	R*4	BETA1		1	42=	49=	54=	55	57(2)=	62	
**	R*4	BETAM		38=	39	44	45	48	49	52	
6-00000004	R*4	CALPHA	COMM	25	80	84					
6-00000008	R*4	CB	COMM	25	74(2)	75	76	77	79	81	82
				95	96	99	107	108(2)	111(2)	115	
8-0000003C	R*4	CBO	COMM	27	95	107					
AP-00000008@	R*4	CL		1	30=	40	63	108	111	115	
AP-0000000C@	R*4	CL1		1	43=	63=					
**	R*4	COSBM		52=	53	55	56				
5-00000784	R*4	CS	COMM	23	40	94	96	99	102	106	108
				111	115						
4-00000000	R*4	CSA	COMM	22	94	106					
AP-00000010@	R*4	DTH		1	44=	45(2)=	50=	55=	58(2)=		
**	R*4	G33		71=	72	74	76				
7-00000000	I*4	IRES	COMM	26	31	32					
3-00000000	I*4	LISTO	COMM	21	90	91					
**	I*4	N		108(2)=	111(2)=	115(2)=					
3-00000004	I*4	NFR	COMM	21	108	111	115				
**	R*4	P1		74=	78	79					
**	R*4	P3		75=	76	82					
**	R*4	P4		76=	81						
**	R*4	P5		77=	79	80	82				
**	R*4	P7		72=	77	79(2)	82				
6-0000000C	R*4	PI2	COMM	25	30						
**	R*4	Q		70=	77	78	84	85			
**	R*4	Q9		78=	83	84	85				
7-00000004	R*4	SA2PI	COMM	26	108	111	115				
**	R*4	SINBM		39=	40	55(2)					
**	R*4	T9		81=	82	85					
**	R*4	TANB1		53=	54	55					
**	R*4	TEM		40=	41	53	62(2)=	63(2)			
7-00000008	R*4	TOC2PI	COMM	26	108	111	115				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
8-00000000	R*4	A	COMM	60	(15)	27	30(2)	108	111	115
5-00000708	R*4	B	COMM	60	(15)	23	75	81	82	
5-00000000	R*4	G	COMM	900	(15, 15)	23	70	71	72	74(2)
						79(2)				
5-00000384	R*4	H	COMM	900	(15, 15)	23	74(3)	76	77(2)	79(2)
5-00000744	R*4	T	COMM	60	(15)	23	75	81	82	

PARAMETER CONSTANTS

Type	Name	References		
I*4	NFRMX	19#	23(6)	27

LABELS

Address	Label	References	
0-0000006C	2100	31	38#
0-0000012C	3100	31	70#

0-0000025C	3200	90	94#				
0-000003FC	3300	90	106#				
0-000005C8	3900	31	46	65	90	104	119#
1-00000041	8200'	96	122#				
1-00000056	8210'	99	123#				
1-00000068	8220'	102	124#				
1-00000076	8300'	108	125#				
1-00000088	8310'	111	126#				
1-0000009D	8320'	115	127#				

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References		
R*4	MTH$ATAND	44	54	55
R*4	MTH$COSD	45	52	
R*4	MTH$SIND	39	62	
R*4	MTH$TAND	44		

```
+------------------------------------------------+
|            KEY TO REFERENCE FLAGS              |
|    =   - Value Modified                        |
|    #   - Defining Reference                    |
|    A   - Actual Argument, possibly modified    |
|    D   - Data Initialization                   |
|   (n)  - Number of occurrences on line         |
+------------------------------------------------+
```

B.4 PROGRAM DSN3

Provided in this section is a listing of the computer program DSN3 and the following subroutines:

DSN2
FILONC
INP
INP2
MAP32
MELLOR2
MELLOR3
PFM3
SIMUL1
and SPLIN1.

```
      PROGRAM DSN3
C (3037-DSN3m) (M) Blade design for 3D flow
C v. 1.0
C Includes SUBs (D) DSN2, (F) FILONC, (I) INP, (J) INP2, (L) MAP32,
C                (N) MELLOR2, (O) MELLOR3, (P) PFM3, ( ) SIMUL1, AND
C                (S) SPLIN1
C References DSN2, INP, INP2, MAP32, AND PFM3
C Coded 23MAR87; TOUCHED 21JAN88; INCLUDED DSNCOM 15JUN88

COM (3037-DSN3-DSNCOM) COMMON STATEMENTS... 09APR87; REVISED 01JUN88
COM      PARAMETER      ( MXN=200, NAMDAM=90, NFRMX=15, NSECRM=15,
COM     *                 NSTAMX=11, NSTLMX=9, NYSMX=101)
COMC    MXN:/S/; NAMDAM:/NO/; NFRMX:/NO/; NSECRM:/IJ/;
COMC    NSTAMX:/DIJNOP/; NSTLMX:INP2; NYSMX:/NO/
COM
COM      EQUIVALENCE    ( A, VEC), ( CL1, CL1D), ( F(1), VY(1))
COM
COM      COMMON/DI   / CONFB, CONFSO, EPSS, EPSSO, PI2N
COM      COMMON/DIJ P/ NSTA,    USTAR
COM      COMMON/DI L / OMEGA
COM      COMMON/DI  P/ EPSA,    IDBUG, MMAX
COM      COMMON/D J  / R(NSTAMX), XM(NSTAMX)
COM      COMMON/D JL / CT1, R1
COM      COMMON/D JLP/ R1N, R2N
COM      COMMON/D J P/ SIG
COM      COMMON/D  L / BETAO, CL1CLM, WXO
COM      COMMON/D  LP/ BETA1, BETA2, TANB1, TANB2
COM      COMMON/D   P/ AO, A1, BO, CB, CHI, CLM, F, PHI, SIGH, STAG, TANBO
COM      COMMON/ IJ  / DEN,     INTERP,        RSTAR
COM      COMMON/ I  P/ CONFC, CONFSA
COM      COMMON/  JL / CM1, CM2, CT2, R2
COM      COMMON/  J P/ B(NSTAMX), BK(NSTAMX), RHO(NSTAMX), RN(NSTAMX),
COM     +                WM1N, WM2N
COM
COM      COMMON/NO/ GJ(NAMDAM,NFRMX,NFRMX), HJ(NAMDAM,NFRMX,NFRMX),
COM     +                BJ(NAMDAM,NFRMX), TJ(NAMDAM,NFRMX),
COM     +                AMDAJ(NAMDAM), FCP(NYSMX), V(NFRMX),
COM     +                XAXIS(NSTAMX), XCC(NYSMX),
COM     +                AMDA, AMDI, AMDAMN, AMDAMX,
COM     +                NAMDA, NFR, NFRSQ, NFRSQP, NYS, NYSM,
COM     +                PI, PI2, TOC

      CHARACTER TDATE*9, VS*3

      DATA      VS/'1.0'/

      OPEN (  5, FILE='DSN3I',  STATUS='OLD', READONLY)
      OPEN (  6, FILE='DSN3O',  STATUS='NEW')
C     OPEN ( 11, FILE='MELO10', STATUS='OLD', READONLY) ... in sub MELLOR2
C     OPEN ( 21, FILE='DSN3ZI', STATUS='OLD', READONLY) ... in sub INP2

      CALL    DATE (TDATE)

      WRITE(6,101) VS, TDATE
  101 FORMAT('1OUTPUT FROM "DSN3"...VERSION 'A3,T61,A9//)

      CALL    INP(NSECR)

      DO      K=1,NSECR

        CALL    INP2(K)

        CALL    MAP32

        CALL    DSN2

        CALL    PFM3

        END DO

      WRITE (6,*) 'Done...'
      STOP 'Done...'
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	166	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	60	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	136	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	362	

ENTRY POINTS

Address	Type	Name	References
0-00000000		DSN3	1#

VARIABLES

Address	Type	Name	Attributes	References		
2-00000010	I*4	K		59=	61A	
2-0000000C	I*4	NSECR		57A	59	
2-00000000	CHAR	TDATE		43	52A	54
2-00000009	CHAR	VS		43	45D	54

LABELS

Address	Label	References	
1-00000013	101'	54	55#

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References	
	DSN2	65	
	FOR$DATE_T_DS	52	
	FOR$OPEN	47	48
	INP	57	
	INP2	61	
	MAP32	63	
	PFM3	67	

```
+------------------------------------------+
|          KEY TO REFERENCE FLAGS          |
|   =   - Value Modified                   |
|   #   - Defining Reference               |
|   A   - Actual Argument, possibly modified|
|   D   - Data Initialization              |
|  (n)  - Number of occurrences on line    |
+------------------------------------------+
```

COMMAND QUALIFIERS

F/LIS/CHE/CRO DSN3M

```
/CHECK=(BOUNDS,OVERFLOW,UNDERFLOW)
/DEBUG=(NOSYMBOLS,TRACEBACK)
/STANDARD=(NOSYNTAX,NOSOURCE_FORM)
/SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
/WARNINGS=(GENERAL,NODECLARATIONS)
/CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
/NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE
```

```
      SUBROUTINE DSN2
C (3037-DSN3-DSN2) (D) DESIGN IN 2D FLOW
C Refernced by MAIN (DSN3)
C References MELLOR3 & SPLIN1 & MTH$ATAND (thrice), MTH$COSD, & MTH$TAND
C To be compiled by VAX-11 FORTRAN, v.4.0
C Coded 03APR87; REVISED 01JUN88

      PARAMETER ( NSTAMX=11)

      CHARACTER TEMS*80

      COMMON/DI   / CONFB, CONFSO, EPSS, EPSSO, PI2N
      COMMON/DIJ P/ NSTA,        USTAR
      COMMON/DI L / OMEGA
      COMMON/DI  P/ EPSA,        IDBUG, MMAX
      COMMON/D J  / R(NSTAMX), XM(NSTAMX)
      COMMON/D JL / CT1, R1
      COMMON/D JLP/ R1N, R2N
      COMMON/D J P/ SIG
      COMMON/D  L / BETA0, CL1CLM, WX0
      COMMON/D  LP/ BETA1, BETA2, TANB1, TANB2
      COMMON/D   P/ A0, A1, B0, CB, CHI, CLM, F, PHI, SIGH, STAG, TANB0

      DIMENSION DUMMY(1), RR(NSTAMX)

C1000 DETERMINE SOLIDITY BY ITERATION *****
      SIGIN     =SIG

      WRITE(6,1010) SIG
 1010 FORMAT(//' LOOP TO DETERMINE SOLIDITY, IN SUB DSN2:'//
     + '                                       DESIGN   STAGGER'/
     + '    M     CL(m)     CL(1)     Cb       ALPHA1    ANGLE   SOLI',
     + 'DITY'//
     + '    0'F60.5)

      DATA      B1MN, B1MX, CL1MN, CL1MX, SIGMN, SIGMX
     +          / 30., 70.,    0.,   1.2,    .5,  1.5/

      IF (BETA1.LT.B1MN .OR. BETA1.GT.B1MX) THEN
        TEMS    ='WARNING(DSN2.1020):  BETA1 out or range...'
        PRINT *,   TEMS, BETA1
        WRITE(6,*) TEMS, BETA1
        END IF

      M =0
      FCOSB2    =(TANB1-TANB2) * COSD(BETA0) * 2.
 1100 M =M+1
      CLM       =FCOSB2/SIG
      CL1       =CL1CLM*CLM
      IF (SIG.LT.SIGMN .OR. SIG.GT.SIGMX .OR. CL1.LT.CL1MN .OR.
     +          CL1.GT.CL1MX) THEN
        TEMS    ='WARNING(DSN2.1110):  SIG or CL1 out or range...'
        PRINT *,   TEMS, SIG, CL1, M
        WRITE(6,*) TEMS, SIG, CL1, M
        END IF

C1200 CAMBER (Cb) & DESIGN ALPHA_1 (FROM MULTIPLE REGRESSION ANALYSIS)
      CB        =( ( ( -2.74752*CL1+20.91537 )*CL1-45.84094 )*CL1
     ,           +36.29869 )*CL1 - 5.23047
     ,           + ( ( ( -1.66604*SIG-CL1*4.42872+8.26201 )*SIG
     ,           + ( -10.87314*CL1+26.19466 )*CL1-14.15108 )*SIG
     ,           + ( ( -9.83274*CL1+41.23668 )*CL1-47.03403 )*CL1
     ,                 +13.59117 )*SIG +
     ,  ( ( ( BETA1*.38594E-6-SIG*.50739E-4-CL1*.35308E-4-.20018E-4 )*
     ,  BETA1 + ( .0013187*SIG+.0040173 )*SIG +
     ,  ( -.0016033*CL1+.0059786 )*CL1 - .00080978 )*BETA1 +
     ,  ( ( -.0014369*SIG-.11946 )*SIG-.010265 )*SIG +
     ,    ( ( -.093633*CL1+.33569 )*CL1-.39504 )*CL1+.015917 )*BETA1
     ALFA1D     =( ( ( -1.78681*SIG+7.68063 )*SIG-12.95580 )*SIG
     ,           +12.87888 )*SIG-2.22656 +
     ,  ( ( ( CB*.01716-SIG*.06408+.01298 )*CB +
     ,  ( SIG*.02078+.30280 )*SIG-.30496 )*CB +
     ,  ( ( SIG*.51975-2.50213 )*SIG+6.54778 )*SIG-2.78086 )*CB

C1300 STAGGER ANGLE
      STAG      =BETA1-ALFA1D
```

```
C1400 SOLIDITY ON X-Y PLANE
      DO       I=1,NSTA
        RR(I)   =1./R(I)
        END DO

      CALL      SPLIN1( NSTA, XM, RR, 0, DUMMY, DUMMY, DUMMY, SIGMA, 1.,
     *                  1.E-6)

      SIGMA     =SIGMA/( COSD(STAG) * PI2N )
      IF ( ABS(SIG-SIGMA) .GE. EPSS ) THEN
        IF (M.GT.MMAX) THEN
          PRINT *, 'ITERATION # EXCEEDED ', MMAX
          STOP 'DSN2.1411'
          END IF
        IF (IDBUG.GT.0)
     +           WRITE(6,6420) M, CLM, CL1, CB, ALFA1D, STAG, SIGMA
        SIG     =(SIGMA-SIG)*CONFSO+SIG
        GOTO 1100
       END IF

      WRITE(6,6420)            M, CLM, CL1, CB, ALFA1D, STAG, SIGMA
      IF (CB.GT.4. .OR. CB.LT.-1. .OR. ALFA1D.GT.36. .OR. ALFA1D.LT.0.)
     + THEN
        PRINT *, 'CHECK CB OR ALFA1D...', CB, ALFA1D
        STOP 'DSN2.1425'
        END IF

      IF ( ABS(SIGIN-SIGMA) .LT. EPSSO) THEN
        SIG     =SIGIN
       ELSE
        PRINT 1455, SIGIN, SIGMA
 1455   FORMAT(' SOLIDITY USED IN "MELLOR" TO ESTABLISH "MELO10.DAT"'/
     +          ' HAS TO BE CORRECTED FROM'F8.5' TO'F8.5)
        STOP 'DSN2.1456'
       END IF

C2000 CONSTANTS *****
      PHI       =WXO/USTAR
      SIGH      =SIG*.5

      CHI       =(R2N*R2N-R1N*R1N)/PHI
C     PSI       =(R2N*R2N-R1N*R1N+PHI*DWYN)*2.

C3000 TURNING ANGLE WITHOUT 3D EFFECTS *****
      M         =0
      BETA0     =ATAND( (TANB1+TANB2)*.5 )
      WRITE(6,3010) TANB1-TANB2, BETA0
 3010 FORMAT(//'    M      A0        A1       CL(m)       F      Beta 0 ',
     + '   Beta 2'//
     + '      0'F40.5,F10.5)

C3100 DETERMINE B0 (OR BETA1 OR TURNING ANGLE) BY ITERATION

 3110 CALL    MELLOR3( BETA0-STAG, CB, STAG, NSTA, VNX, VNY, 2,
     +                 A0, A1, CLM)

      M         =M+1
      F         =CLM * SIGH / COSD(BETA0)
      B0        =ATAND(TANB1-F*.5)

      IF ( ABS(B0-BETA0) .GT. EPSA ) THEN
        IF (M.GT.MMAX) THEN
          PRINT *, 'ITERATION # EXCEEDED ', MMAX
          STOP 'DSN2.2111'
          END IF
        IF (IDBUG.GT.0)      WRITE(6,6420) M, A0, A1, CLM, F, B0
        BETA0   =(B0-BETA0)*CONFB+BETA0
        GOTO 3110
        END IF

      TANB0     =TAND(B0)
      TANB2     =TANB1-F
      BETA2     =ATAND(TANB2)
      WRITE(6,6420) M, A0, A1, CLM, F, B0, BETA2

 6420 FORMAT(I5,6F10.5)

      RETURN
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes								
0	$CODE	1932	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
1	$PDATA	558	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT	LONG
2	$LOCAL	312	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	LONG
3	DI	20	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
4	DIJP	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
5	DIL	4	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
6	DIP	12	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
7	DJ	88	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
8	DJL	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
9	DJLP	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
10	DJP	4	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
11	DL	12	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
12	DLP	16	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
13	DP	44	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
	Total Space Allocated	3026									

ENTRY POINTS

Address	Type	Name	References
0-00000000		DSN2	1#

VARIABLES

Address	Type	Name	Attributes	References							
13-00000000	R*4	A0	COMM	22	131A	143	151				
13-00000004	R*4	A1	COMM	22	131A	143	151				
**	R*4	ALFA1D		69=	76	92	98	99(2)	101		
13-00000008	R*4	B0	COMM	22	136=	138	143	144	148	151	
2-00000084	R*4	B1MN		360	39						
2-00000088	R*4	B1MX		360	39						
11-00000000	R*4	BETA0	COMM	20	46	123=	124	131	135	138	144(3)=
12-00000000	R*4	BETA1	COMM	21	39(2)	41	42	58(4)	76		
12-00000004	R*4	BETA2	COMM	21	150=	151					
13-0000000C	R*4	CB	COMM	22	58=	69(4)	92	98	99(2)	101	131A
13-00000010	R*4	CHI	COMM	22	118=						
**	R*4	CL1		49=	50(2)	53	54	58(16)	92	98	
11-00000004	R*4	CL1CLM	COMM	20	49						
2-0000008C	R*4	CL1MN		360	50						
2-00000090	R*4	CL1MX		360	50						
13-00000014	R*4	CLM	COMM	22	48=	49	92	98	131A	135	143
				151							
3-00000000	R*4	CONFB	COMM	12	144						
3-00000004	R*4	CONFSO	COMM	12	94						
8-00000000	R*4	CT1	COMM	17							
6-00000000	R*4	EPSA	COMM	15	138						
3-00000008	R*4	EPSS	COMM	12	87						
3-0000000C	R*4	EPSSO	COMM	12	105						
13-00000018	R*4	F	COMM	22	135=	136	143	149	151		
**	R*4	FCOSB2		46=	48						
**	I*4	I		79=	80(2)						
6-00000004	I*4	IDBUG	COMM	15	92	143					
**	I*4	M		45=	47(2)=	53	54	88	92	98	122=
				134(2)=	139	143	151				
6-00000008	I*4	MMAX	COMM	15	88	89	139	140			
4-00000000	I*4	NSTA	COMM	13	79	83A	131A				
5-00000000	R*4	OMEGA	COMM	14							
13-0000001C	R*4	PHI	COMM	22	115=	118					
3-00000010	R*4	PI2N	COMM	12	86						
8-00000004	R*4	R1	COMM	17							
9-00000000	R*4	R1N	COMM	18	118(2)						
9-00000004	R*4	R2N	COMM	18	118(2)						
10-00000000	R*4	SIG	COMM	19	27	29	48	50(2)	53	54	58(10)
				69(10)	87	94(3)=	106=	116			
13-00000020	R*4	SIGH	COMM	22	116=	135					

2-00000080	R*4	SIGIN		27=	105	106	108				
2-0000009C	R*4	SIGMA		83A	86(2)=	87	92	94	98	105	108
2-00000094	R*4	SIGMN		36D	50						
2-00000098	R*4	SIGMX		36D	50						
13-00000024	R*4	STAG	COMM	22	76=	86	92	98	131(2)A		
13-00000028	R*4	TANBO	COMM	22	148=						
12-00000008	R*4	TANB1	COMM	21	46	123	124	136	149		
12-0000000C	R*4	TANB2	COMM	21	46	123	124	149=	150		
2-00000030	CHAR	TEMS		10	40=	41	42	52=	53	54	
4-00000004	R*4	USTAR	COMM	13	115						
2-000000A0	R*4	VNX		131A							
2-000000A4	R*4	VNY		131A							
11-00000008	R*4	WXO	COMM	20	115						

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References		
2-00000000	R*4	DUMMY		4	(1)	24	83(3)A	
7-00000000	R*4	R	COMM	44	(11)	16	80	
2-00000004	R*4	RR		44	(11)	24	80=	83A
7-0000002C	R*4	XM	COMM	44	(11)	16	83A	

PARAMETER CONSTANTS

Type	Name	References		
I*4	NSTAMX	8#	16(2)	24

LABELS

Address	Label	References				
1-0000005E	1010'	29	30#			
0-00000008	1100	47#	95			
1-00000119	1455'	108	109#			
1-00000177	3010'	124	125#			
0-00000508	3110	131#	145			
1-000001CD	6420'	92	98	143	151	153#

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References		
	MELLOR3	131		
R*4	MTH$ATAND	123	136	150
R*4	MTH$COSD	46	86	135
R*4	MTH$TAND	148		
	SPLIN1	83		

```
+----------------------------------------------+
|           KEY TO REFERENCE FLAGS             |
|   =   - Value Modified                       |
|   #   - Defining Reference                   |
|   A   - Actual Argument, possibly modified   |
|   D   - Data Initialization                  |
|  (n)  - Number of occurrences on line        |
+----------------------------------------------+
```

```
      SUBROUTINE FILONC(N2,F,NK,SUM)

C (3037-MELLOR-FICONC) INTEGRATE USING FILON'S FORMULA TO FIND FOURIER
C                     (COSINE) COEFS.
C REFERENCED BY INPUT; REFERENCES MTH$COSD (4 TIMES) & MTH$SIND (TWICE)
C INPUT N2, NK, & F
C OUTPUT SUM(J) = INT( F(X) * COS[(J-1)*X] DX ), FOR J = 1 TO NK
C F IS SUPPOSED TO BE CORRESOPNDING TO X(1)=0 TO X(N2+1)=PI WHILE
C       ELEMENTS OF X ARE OF EQUAL INCREMENT OF H=PI/N2
C N2 SHOULD BE AN EVEN NUMBER
C CODED BY W. CHIANG, 28JUL83
C 10OCT86, S. HSU TRANSLATED FROM (3037-CASCAD[HP9845]-FILONC) WHICH WAS
C       COPIED FROM (3725-CASCAD-FILONC), 06AUG85, WHICH WAS ADAPTED
C       FROM (3725-FRRFIT)
C REVISED 30FEB87, CHIANG

      DIMENSION F(N2+1), SUM(NK)

      DATA      PI/3.14159265359/

C 100
      IF ( MOD(N2,2).GT.0 ) THEN
        PRINT *, 'N2 should be even...', N2
        STOP 'FILONC.101'
        END IF

C 200
      H=PI/N2
      N2M=N2-1
      F1H=F(1)*.5
      FN2=F(N2)
      SFN2PH=F(N2+1)*.5
      C2NM      =0.
      C2N=F1H-SFN2PH
      DO        I=2,N2,2
        C2NM    =F(I)+C2NM
        C2N     =F(I+1)+C2N
        END DO
      SUM(1)    =(C2N+C2NM+C2NM)*(H+H)/3.
      DO        J=2,NK
        SFN2PH=-SFN2PH
        TH=H*(J-1)
        RTH=1./TH
        RTHSQ2  =RTH*RTH*2.
        CO=COS(TH)
        C2NM    =0.
        C2N=F1H+SFN2PH
        DO      I=2,N2M,2
          C2NM  =COS( (I-1)*TH ) * F(I) + C2NM
          C2N   =COS(I*TH) * F(I+1) + C2N
          END DO
        C2NM=COS(N2M*TH)*FN2+C2NM
        SUM(J)  =( ( CO * CO + 1. - SIN(TH+TH) * RTH ) * RTHSQ2 * C2N +
     +              ( SIN(TH)*RTH - CO ) * ( RTHSQ2+RTHSQ2 ) * C2NM ) * H
        END DO

      RETURN
      END
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 $CODE	561	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 $PDATA	30	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 $LOCAL	144	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
Total Space Allocated	735	

ENTRY POINTS

Address Type Name References

```
0-00000000       FILONC                  1#

VARIABLES

   Address    Type  Name          Attributes  References

      **       R*4  C2N                          34=       37(2)=    39        47=       50(2)=    53
      **       R*4  C2NM                         33=       36(2)=    39(2)     46=       49(2)=    52(2)=    53
 2-00000028    R*4  CO                           45=       53(3)
 2-0000000C    R*4  F1H                          30=       34        47
 2-00000010    R*4  FN2                          31=       52

 2-00000004    R*4  H                            28=       39(2)     42        53
      **       I*4  I                            35=       36        37        48=       49(2)     50(2)
 2-00000018    I*4  J                            40=       42        53
AP-00000004@   I*4  N2                            1        17        22        23        28        29        31        32
                                                 35
 2-00000008    I*4  N2M                          29=       48        52

AP-0000000C@   I*4  NK                            1        17        40
 2-00000000    R*4  PI                           19D       28
 2-00000020    R*4  RTH                          43=       44(2)     53(2)
 2-00000024    R*4  RTHSQ2                       44=       53(3)
 2-00000014    R*4  SFN2PH                       32=       34        41(2)=    47

      **       R*4  TH                           42=       43        45        49        50        52        53(3)

ARRAYS

   Address    Type  Name          Attributes         Bytes  Dimensions        References

AP-00000008@   R*4  F                                  **   (*)                 1         17        30        31        32
                                                                               36         37        49        50
AP-00000010@   R*4  SUM                                **   (*)                 1         17        39=       53=

FUNCTIONS AND SUBROUTINES REFERENCED

  Type  Name                                References

  R*4   MTH$COS                               45        49        50        52
  R*4   MTH$SIN                               53(2)

+-------------------------------------------+
|            KEY TO REFERENCE FLAGS         |
|   =    - Value Modified                   |
|   #    - Defining Reference               |
|   A    - Actual Argument, possibly modified|
|   D    - Data Initialization              |
|  (n)   - Number of occurrences on line    |
+-------------------------------------------+
```

```
      SUBROUTINE INP(NSECR)
C (3037-DSN3-INP) (I) INPUT DATA
C Refernced by MAIN (DSN3); references none.
C Output NSECR
C To be compiled by VAX-11 FORTRAN, v.4.0
C Coded 09APR87; REVISED 20MAY87; SLIGHTLY REVISED 16OCT87; TOUCHED 16JUN88

C
C READ1-TITLE FOR THIS RUN
C
C READ2-DATA FOR THE SYSTEM
C DEN   FLUID DENSITY; ENTER A NUMBER, IN PROPER UNIT, HERE IF IT IS
C       CONSTANT OTHERWISE ENTER 0 AND ENTER THE DATA IN SUB "INP2"
C RPM   ROTATIONAL SPEED, IN rpm
C RSTAR REFERENCE RADIUS, IN USER'S UNIT
C CONVR CONVERSION FACTOR TO BE MULTIPLY TO RSTAR SUCH THAT RSTAR IS IN
C       METERS OR FEET
C
C READ3-CONVERGENCE CRITERIA
C CONFB CONVERGENCE FACTOR FOR ADJUSTING BETA0, e.g., .5
C CONFC CONVERGENCE FACTOR FOR ADJUSTING CB, e.g., 1.
C CONFSACONVERGENCE FACTOR FOR ADJUSTING STAGGER ANGLE, STAG, e.g., 0.
C CONFSOCONVERGENCE FACTOR FOR ADJUSTING SOLIDITY, SIG, e.g., 1.
C EPSA  CONVERGENCE CRITERIA FOR BOTH del EQ. PHI AND del BETA1, e.g., .001
C EPSS  CONVERGENCE CRITERIA FOR SOLIDITY ITERATION, e.g., .001
C EPSSO ACCEPTABLE CRITERIA FOR INPUT SOLIDITY, e.g., .01
C MMAX  MAXIMUM ITERATION NUMBER FOR A LOOP (del EQ. PHI & del BETA1) IN
C       SUB "PFM3"
C IDBUG > 0 TO HAVE EXTRA OUTPUT FOR DEBUGGING PURPOSE;
C       = 0 FOR NORMAL OUTPUT
C
C READ4-SOME NUMBERS FOR ROTOR & ITS BLADES
C NBLADE# OF BLADES
C NSECR # OF CROSS-SECTIONS ALONG THE RADIUS DIRECTION (USE NSECR=1 AT
C       THIS MOMENT BECAUSE SOLIDITY VARIES FROM SECTION TO SECTION)
C NSTA  # OF STATIONS ALONG A CHORD, INCLUDING LEADING & TRAILING EDGES
C INTERP=  0 TO READ THE SECTION DATA DIRECTLY FROM INPUT;
C       <> 0 TO OBTAIN THE SECTION DATA THROUGH INTERPOLATION BASED ON
C            DATA READ FROM UNIT 21 (IN SUB INP2)
C
C READ2XX.... IN SUB "INP2"
C
C READ11XX... IN SUB "MELLOR2"
C

      PARAMETER ( NSECRM=15, NSTAMX=11)

      COMMON/DI   / CONFB, CONFSO, EPSS, EPSSO, PI2N
      COMMON/DIJ P/ NSTA,       USTAR
      COMMON/DI L / OMEGA
      COMMON/DI  P/ EPSA,       IDBUG, MMAX
      COMMON/ IJ  / DEN,        INTERP,        RSTAR
      COMMON/ I  P/ CONFC, CONFSA

      DATA     PI/3.14159265359/

C 100 READ DATA & ECHO
      READ (5,*)                                                     READ1
      READ (5,*) DEN, RPM, RSTAR, CONVR                              READ2
      WRITE(6,130)    RPM, RSTAR, CONVR
  130 FORMAT(' ROTATIONAL SPEED (rpm) ='T60,F10.5/
     +  ' REFERENCE RADIUS ( in user''s unit) ='T60,F10.5/
     +  ' CONVERSION FACTOR TO BE MULTIPLY TO "RSTAR" SUCH THAT THE'/
     +  '    RADIUS IS IN feet OR meters ='T60,F10.5)

      IF (DEN.GT.0.) THEN
        WRITE(6,*) 'FLUID DENSITY IS CONSTANT...'
      ELSE
        PRINT *, 'IF DEN=0, CHECK STATEMENTS WITH CRHO IN PFM3...'
        STOP 'INP.135'
      END IF

      IF (CONVR.GT.8.3 .AND. CONVR.LT.8.4) THEN
        RSTAR    =RSTAR/12.
      ELSE
        RSTAR    =RSTAR*CONVR
      END IF
```

```

      READ (5,*)  CONFB, CONFC, CONFSA, CONFSO, EPSA, EPSS, EPSSO,
     +            MMAX, IDBUG                                        READ3
      WRITE(6,140) CONFB, CONFC, CONFSA, CONFSO, EPSA, EPSS, EPSSO,
     +            MMAX, IDBUG
  140 FORMAT(/' CONVERGENCE FACTOR FOR ADJUSTING BETAO ='T60,F7.2/
     +  ' CONVERGENCE FACTOR FOR ADJUSTING Cb ='T60,F7.2/
     +  ' CONVERGENCE FACTOR FOR ADJUSTING STAGGER ANGLE ='T60,F7.2/
     +  ' CONVERGENCE FACTOR FOR ADJUSTING SOLIDITY ='T60,F7.2/
     +  ' CONVERGENCE CRITERIA FOR del BETAO & del EQIVA. PHI ='T59,
     +  1PE11.1/
     +  ' CONVERGENCE CRITERIA TO DETERMINE SOLIDITY ='T59,1PE11.1/
     +  ' ACCEPTABLE ERROR IN SOLIDITY ='T59,1PE11.1/
     +  ' MAXIMUM ITERATION NUMBER ='T60,I4/' IDBUG ='T60,I4)

      OMEGA     =PI*RPM/30.
      USTAR     =RSTAR*OMEGA
      WRITE(6,210) OMEGA, RSTAR, USTAR
  210 FORMAT(/' ANGULAR VELOCITY (rad/sec) =',T60,F10.5/
     +  ' REFERENCE RADIUS AFTER CONVERTED TO BE IN ft or meters ='T60,
     +  F10.5/
     +  ' REFERENCE BLADE SPEED IN fps or mps ='T60,F10.5)

      READ (5,*)  NBLADE, NSECR, NSTA, INTERP                        READ4
      WRITE(6,310) NBLADE, NSECR, NSTA
  310 FORMAT(/' NUMBER OF BLADES ='T60,I4/
     +  ' NUMBER OF CROSS-SECTIONS ALONG THE RADIUS DIRECTION ='T60,I4/
     +  ' NUMBER OF STATIONS ALONG A CHORD ='T60,I4/)

      IF (INTERP.EQ.0) THEN
        WRITE(6,*)
     +          'THE FOLLOWING SECTION DATA READ DIRECTLY FROM INPUT:'
      ELSE
        WRITE(6,*) 'SECTION DATA OBTAINED THROUGH INTERPOLATION BASED ',
     +          'ON DATA READ FROM UNIT 21'
      END IF

      IF (NSECR.GT.NSECRM .OR. NSTA.LT.3 .OR. NSTA.GT.NSTAMX) THEN
        PRINT *, 'NSECR.GT.NSECRM .OR. NSTA.GT.NSTAMX...', NSECR,
     +                    NSECRM, NSTA, NSTAMX
        STOP 'INP.153'
      END IF

C 500 DATA PREPARATIONS
      PI2N      =(PI+PI)/NBLADE

      RETURN
      END
```

PROGRAM SECTIONS

Name	Bytes	Attributes								
0 $CODE	901	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
1 $PDATA	1136	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT	LONG
2 $LOCAL	80	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	LONG
3 DI	20	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
4 DIJP	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
5 DIL	4	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
6 DIP	12	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
7 IJ	12	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
8 IP	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
Total Space Allocated	2181									

ENTRY POINTS

Address	Type	Name	References
0-00000000		INP	1#

VARIABLES

Address	Type	Name	Attributes	References

```
 3-00000000  R*4  CONFB       COMM         48      79=      81
 8-00000000  R*4  CONFC       COMM         53      79=      81
 8-00000004  R*4  CONFSA      COMM         53      79=      81
 3-00000004  R*4  CONFSO      COMM         48      79=      81
 2-00000008  R*4  CONVR                    59=     60       73(2)    76

 7-00000000  R*4  DEN         COMM         52      59=      66
 6-00000000  R*4  EPSA        COMM         51      79=      81
 3-00000008  R*4  EPSS        COMM         48      79=      81
 3-0000000C  R*4  EPSSO       COMM         48      79=      81
 6-00000004  I*4  IDBUG       COMM         51      79=      81

 7-00000004  I*4  INTERP      COMM         52      101=     107
 6-00000008  I*4  MMAX        COMM         51      79=      81
 2-0000000C  I*4  NBLADE                   101=    102      122
AP-00000004@ I*4  NSECR                    1       101=     102      115      116
 4-00000000  I*4  NSTA        COMM         49      101=     102      115(2)   116

 5-00000000  R*4  OMEGA       COMM         50      93=      94       95
 2-00000000  R*4  PI                       55D     93       122(2)
 3-00000010  R*4  PI2N        COMM         48      122=
 2-00000004  R*4  RPM                      59=     60       93
 7-00000008  R*4  RSTAR       COMM         52      59=      60       74(2)=   76(2)=   94     95

 4-00000004  R*4  USTAR       COMM         49      94=      95
```

```
PARAMETER CONSTANTS

 Type  Name                                References

  I*4  NSECRM                              46#     115      116
  I*4  NSTAMX                              46#     115      116
```

```
LABELS

   Address    Label                        References

 1-00000106   130'                         60      61#
 1-000001BA   140'                         81      83#
 1-00000358   210'                         95      96#
 1-000003ED   310'                         102     103#
```

```
+--------------------------------------------+
|           KEY TO REFERENCE FLAGS           |
|   =    - Value Modified                    |
|   #    - Defining Reference                |
|   A    - Actual Argument, possibly modified|
|   D    - Data Initialization               |
|  (n)   - Number of occurrences on line     |
+--------------------------------------------+
```

```
COMMAND QUALIFIERS

 F/LIS/CHE/CRO INP.FOR

 /CHECK=(BOUNDS,OVERFLOW,UNDERFLOW)
 /DEBUG=(NOSYMBOLS,TRACEBACK)
 /STANDARD=(NOSYNTAX,NOSOURCE_FORM)
 /SHOW=(NOPREPROCESSOR,NOINCLUDE,MAP,NODICTIONARY,SINGLE)
 /WARNINGS=(GENERAL,NODECLARATIONS)
 /CONTINUATIONS=19  /CROSS_REFERENCE  /NOD_LINES  /NOEXTEND_SOURCE  /F77
 /NOG_FLOATING  /I4  /NOMACHINE_CODE  /OPTIMIZE
```

```
      SUBROUTINE INP2(K)
C (3037-DSN3-INP2) (J) 2ND INPUT ROUTINE
C Referenced by MAIN (DSN3); references MELLOR2 & SPLIN1
C Coded 03APR87; REVISED 01JUN88

C
C ...AFTER READ7 IN "INP"...
C
C READ201       TITLE
C READ202       SIG     INITIALLY GUESSED SOLIDITY

C ***** READ203 TO READ210 REQUIRED iff INTERP=0 *****
C ***** READ210 REQUIRED iff DEN=0 *****
C READ203       CT1     PERIPHERAL VEL. AT THE LEADING EDGE (L/T)
C               CT2     PERIPHERAL VEL. AT THE TRAILING EDGE (L/T)
C READ204       XAX     X COORDINATES ALONG AXIX OF ROTATION, FOR
C                       EACH OF NSTA STATIONS ALONG THE CHORD (L).
C                       XAX WILL BE NORMALIZED TO CHORD LENGTH, IN SUB
C                       "MELLOR2", SUCH THAT XAX(1)=0 & XAX(NSTA)=1 (.)
C READ205       R       RADIUS FROM AXIS OF ROTATION (L)
C READ206       XM      DISTANCE ALONG STREAMLINE (M-COORDINATE) (L)
C READ207       CM      MERIDIONAL VEL. (L/T)
C READ208       B       DISTANCE BETWEEN ADJACENT STREMLINES (L)
C                       (EITHER BETWEEN I-1 & I+1 OR HALF OF THAT)
C READ209       BK      BLOCKAGE FACTOR OWING TO BLADE THICKNESS INSIDE
C                       THE IMPELLER (.)
C ***** READ210 REQUIRED iff DEN=0 *****
C READ210       RHO     DENSITY OF FLUID (M/L^3)

C ***** READ211 REQUIRED iff INTERP<>0 *****
C READ211       RSL     RADIUS DISTANCE OF THE STREAMLINE FROM THE AXIS
C                       OF ROTATION, AT SECTION IRSL (SEE BELOW) (L)
C                       (DATA INPUT FROM FILE #21 ARE USED TO ESTABLISH
C                       THE PERAMETERS BASED ON THIS RADIUS & ITS
C                       NEARBY INPUT RADII BY SIMPLE INTERPOLATION)
C               IRSL    STATION # (FROM 1 TO NSTA) WHERE RSL IS DEFINED

C ***** READ2101 TO READ210 REQUIRED iff INTERP<>0 *****
C ***** READ2111 REQUIRED iff DEN=0 *****
C READ2101      (REM)   TITLE LINE
C READ2102      NI      # OF STATIONS (q-LINES) FROM LEADING EDGE TO
C                       TRAILING EDGE
C               NJ      # OF STREAMLINES
C READ2103      CT1Z    PERIPHERAL VEL. AT THE LEADING EDGE (L/T)
C READ2104      CT2Z    PERIPHERAL VEL. AT THE TRAILING EDGE (L/T)
C READ2105      XAXZ    X COORDINATES ALONG AXIX OF ROTATION, FOR EACH
C                       OF NSTA STATIONS ALONG THE CHORD (L).
C                       XAX WILL BE NORMALIZED TO CHORD LENGTH, IN SUB
C                       "MELLOR2", SUCH THAT XAX(1)=0 & XAX(NSTA)=1
C READ2106      RZ      RADIUS FROM AXIS OF ROTATION (L)
C READ2107      XMZ     DISTANCE ALONG STREAMLINE (M-COORDINATE) (L)
C READ2108      CMZ     MERIDIONAL VEL. (L/T)
C READ2109      BZ      DISTANCE BETWEEN ADJACENT STREMLINES (L)
C                       (EITHER BETWEEN I-1 & I+1 OR HALF OF THAT)
C READ2110      BKZ     BLOCKAGE FACTOR OWING TO BLADE THICKNESS INSIDE
C                       THE IMPELLER (.)
C ***** READ2111 REQUIRED iff DEN=0 *****
C READ2111      RHOZ    DENSITY OF FLUID (M/L^3)
C

      CHARACTER*80      REM

      PARAMETER ( NSTAMX=11, NSTLMX=9)
C       NSTLMX=MAX. # OF STREAM LINES W/ DATA PROVIDED BY UNIT 21

      COMMON/DIJ P/ NSTA,       USTAR
      COMMON/D J  / R(NSTAMX), XM(NSTAMX)
      COMMON/D JL / CT1, R1
      COMMON/D JLP/ R1N, R2N
      COMMON/D J P/ SIG
      COMMON/ IJ  / DEN,        INTERP,         RSTAR
      COMMON/  JL / CM1, CM2, CT2, R2
      COMMON/  J P/ B(NSTAMX), BK(NSTAMX), RHO(NSTAMX), RN(NSTAMX),
     +                 WM1N, WM2N

      DIMENSION BKZ(NSTAMX,NSTLMX), CMZ(NSTAMX,NSTLMX),
     + QZ(NSTAMX,NSTLMX), RHOZ(NSTAMX,NSTLMX), RZ(NSTAMX,NSTLMX),
```

```
     + XAXZ(NSTAMX,NSTLMX), XMZ(NSTAMX,NSTLMX),
     + CM(NSTAMX), XAX(NSTAMX), WMN(NSTAMX),
     + CT1Z(NSTLMX), CT2Z(NSTLMX)

C 100 INPUT & ECHO FOR EACH SECTION
      READ (5,5110) REM                                                 READ201
      READ (5,*) SIG                                                    READ202
      WRITE(6,105) K, REM, SIG
  105 FORMAT('1***********************'/' CROSS-SECTION NUMBER'I3/
     + ' ***********************'//1XA80///
     + ' INITIALLY GUESSED SOLIDITY ='T54,F6.2)
      IF (INTERP.EQ.0) THEN
C 110   INPUT W/ KNOWN VALUES
        READ (5,*) CT1, CT2                                             READ203
        READ (5,*) ( XAX(I), I=1,NSTA)                                  READ204
        READ (5,*) (   R(I), I=1,NSTA)                                  READ205
        READ (5,*) (  XM(I), I=1,NSTA)                                  READ206
        READ (5,*) (  CM(I), I=1,NSTA)                                  READ207
        READ (5,*) (   B(I), I=1,NSTA)                                  READ208
        READ (5,*) (  BK(I), I=1,NSTA)                                  READ209
        IF (DEN.EQ.0.) THEN
          READ (5,*) ( RHO(I), I=1,NSTA)                                READ210
        END IF
      ELSE
C 130   OBTAIN PARAMETERS THROUGH INTERPOLATION
        READ (5,*)   RSL, IRSL                                          READ211
        WRITE(6,132) RSL, IRSL
  132   FORMAT(' RADIUS OF STREAMLINE AT THE FOLLOWING SEC. # IRSL ='
     +  T54,F10.6/
     +  ' IRSL, SECTION NUMBER TO LOCATE THE ABOVE RADIUS ='T54,I3)
        OPEN( 21, FILE='DSN3ZI', STATUS='OLD', READONLY)                OPEN21
        READ (21,*)                                                     READ2101
        READ (21,*) NI, NJ                                              READ2102
        IF (NI.NE.NSTA) THEN
          PRINT *, 'NI.NE.NSTA... CHECK CODING...'
          STOP 'INP2.135'
        END IF
        READ (21,*) ( CT1Z(J), J=1,NJ)                                  READ2103
        READ (21,*) ( CT2Z(J), J=1,NJ)                                  READ2104
        READ (21,*) ( ( XAXZ(I,J), I=1,NI), J=1,NJ)                     READ2105
        READ (21,*) ( (   RZ(I,J), I=1,NI), J=1,NJ)                     READ2106
        READ (21,*) ( (  XMZ(I,J), I=1,NI), J=1,NJ)                     READ2107
        READ (21,*) ( (  CMZ(I,J), I=1,NI), J=1,NJ)                     READ2108
        READ (21,*) ( (   QZ(I,J), I=1,NI), J=1,NJ)                     READ2109
        READ (21,*) ( (  BKZ(I,J), I=1,NI), J=1,NJ)                     READ2110
        IF (DEN.EQ.0.) THEN
          READ (21,*) ( (  RHOZ(I,J), J=1,NJ), I=1,NI)                  READ2111
        END IF
        CLOSE(21)                                                       CLOSE21

        DO     J=1,NJ
          IF ( RZ(IRSL,J).GT.RSL ) GOTO 140
        END DO
        J      =NJ
        GOTO 150

  140   IF (J.EQ.1) GOTO 150
        JM      =J-1
        RR1     =( RSL-RZ(IRSL,JM) ) / ( RZ(IRSL,J)-RZ(IRSL,JM) )
        RR2     =1.-RR1
        CT1     =CT1Z(J)*RR1+CT1Z(JM)*RR2
        CT2     =CT2Z(J)*RR1+CT2Z(JM)*RR2
        DO  JJ=1,NJ
          CT1Z(JJ)      =RZ(IRSL,JJ)
        END DO
        DO     I=1,NSTA
          IF (J.EQ.2) THEN
            BJM =( QZ(I,2)-QZ(I,1) )*2.
          ELSE
            BJM =QZ(I,J)-QZ(I,J-2)
          END IF
          IF (J.EQ.NJ) THEN
            BJ  =( QZ(I,NJ)-QZ(I,NJM) )*2.
          ELSE
            BJ  =QZ(I,J+1)-QZ(I,J-1)
          END IF

          B(I)  =        BJ*RR1 +        BJM*RR2
          BK(I) = BKZ(I,J)*RR1 +  BKZ(I,JM)*RR2
          R(I)  =  RZ(I,J)*RR1 +   RZ(I,JM)*RR2
```

```
            RHO(I)=RHOZ(I,J)*RR1 + RHOZ(I,JM)*RR2
            XAX(I)=XAXZ(I,J)*RR1 + XAXZ(I,JM)*RR2
            XM(I) = XMZ(I,J)*RR1 +  XMZ(I,JM)*RR2
            DO  JJ=1,NJ
              CT2Z(JJ)    =CMZ(I,JJ)
            END DO

            CALL  SPLIN1( NJ, CT1Z, CT2Z, 1, RSL, CMI, DUM, DUM, 0., 1E-4)

            CM(I) =CMI
          END DO
          GOTO 160

  150     REM     =' *** WARNING *** RSL BEYOND RANGE OF STREAMLINE...'
          WRITE(6,*) REM, RSL, IRSL, J, RZ(IRSL,J)
          PRINT *,   REM
          CT1     =CT1Z(J)
          CT2     =CT2Z(J)
          DO     I=1,NSTA
            IF (J.EQ.1) THEN
              B(I)=( QZ(I,2) -QZ(I,1)   )*2.
            ELSE
              B(I)=( QZ(I,NJ)-QZ(I,NJM) )*2.
            END IF
            BK(I) = BKZ(I,J)
            CM(I) = CMZ(I,J)
            R(I)  =  RZ(I,J)
            RHO(I)=RHOZ(I,J)
            XAX(I)=XAXZ(I,J)
            XM(I) = XMZ(I,J)
          END DO
  160 END IF

        IF (DEN.GT.0.) THEN
          DO     I=1,NSTA
            RHO(I)=DEN
          END DO
        END IF

        WRITE(6,170) CT1, CT2,
     +  ( I, R(I), XAX(I), XM(I), CM(I), B(I), BK(I), RHO(I), I=1,NSTA)
  170 FORMAT(' PERIPHERAL VEL. AT LEADING EDGE = 'T54,F9.5/
     +  ' PERIPHERAL VEL. AT TRAILING EDGE ='T54,F9.5///
     +  4X'I      R         X        XM       CM        B       BK',
     +  '           RHO'//
     +  (I5,7F10.5))

C 200 PREPARE FOR MELLOR'S METHOD

        CALL      MELLOR2( NSTA, SIG, XAX)

C1000 PREPARATION
        CM1       =CM(1)
        CM2       =CM(NSTA)
        R1        =R(1)
        R2        =R(NSTA)

        DO        I=1,NSTA
          RN(I)   =R(I)/RSTAR
          WMN(I)  =CM(I)/USTAR
        END DO

        R1N       =RN(1)
        R2N       =RN(NSTA)
        WM1N      =WMN(1)
        WM2N      =WMN(NSTA)

C5000 FORMAT
 5110 FORMAT(A80)

        RETURN
        END
```

PROGRAM SECTIONS

Name   Bytes   Attributes

```
 0 $CODE                          3129   PIC CON REL LCL   SHR   EXE    RD NOWRT LONG
 1 $PDATA                          527   PIC CON REL LCL   SHR NOEXE    RD NOWRT LONG
 2 $LOCAL                         3224   PIC CON REL LCL NOSHR NOEXE    RD   WRT LONG
 3 DIJP                              8   PIC OVR REL GBL   SHR NOEXE    RD   WRT LONG
 4 DJ                               88   PIC OVR REL GBL   SHR NOEXE    RD   WRT LONG
 5 DJL                               8   PIC OVR REL GBL   SHR NOEXE    RD   WRT LONG
 6 DJLP                              8   PIC OVR REL GBL   SHR NOEXE    RD   WRT LONG
 7 DJP                               4   PIC OVR REL GBL   SHR NOEXE    RD   WRT LONG
 8 IJ                               12   PIC OVR REL GBL   SHR NOEXE    RD   WRT LONG
 9 JL                               16   PIC OVR REL GBL   SHR NOEXE    RD   WRT LONG
10 JP                              184   PIC OVR REL GBL   SHR NOEXE    RD   WRT LONG

   Total Space Allocated          7208


ENTRY POINTS

   Address   Type  Name                 References

  0-00000000       INP2                    1#


VARIABLES

   Address   Type  Name      Attributes References

      **     R*4   BJ                     150=     152=     155
      **     R*4   BJM                    145=     147=     155
  9-00000000 R*4   CM1       COMM          72      210=
  9-00000004 R*4   CM2       COMM          72      211=
  2-00000C10 R*4   CMI                    165A     167

  5-00000000 R*4   CT1       COMM          68       91=     138=     174=     197
  9-00000008 R*4   CT2       COMM          72       91=     139=     175=     197
  8-00000000 R*4   DEN       COMM          71       98      123      191      193
  2-00000C14 R*4   DUM                    165(2)A
      **     I*4   I                       92(2)=   93(2)=   94(2)=   95(2)=   96(2)=   97(2)=   99(2)=  117(2)=
                                          118(2)=  119(2)=  120(2)=  121(2)=  122(2)=  124(2)=  143=     145(2)
                                          147(2)   150(2)   152(2)   155      156(3)   157(3)   158(3)   159(3)
                                          160(3)   162      167      176=     178(3)   180(3)   182(2)   183(2)
                                          184(2)   185(2)   186(2)   187(2)   192=     193      197(9)=  215=
                                          216(2)   217(2)

  8-00000004 I*4   INTERP    COMM          71       89
  2-00000BF4 I*4   IRSL                   103=     104      129      136(3)   141      172(2)
  2-00000C00 I*4   J                      115(2)=  116(2)=  117(2)=  118(2)=  119(2)=  120(2)=  121(2)=  122(2)=
                                          124(2)=  128=     129      131=     134      135      136      138
                                          139      144      147(2)   149      152(2)   156      157      158
                                          159      160      172(2)   174      175      177      182      183
                                          184      185      186      187
      **     I*4   JJ                     140=     141(2)   161=     162(2)
      **     I*4   JM                     135=     136(2)   138      139      156      157      158      159
                                          160

 AP-00000004@ I*4  K                        1       85
  2-00000BF8 I*4   NI                     110=     111      117      118      119      120      121      122
                                          124
  2-00000BFC I*4   NJ                     110=     115      116      117      118      119      120      121
                                          122      124      128      131      140      149      150      161
                                          165A     180
  2-00000C0C I*4   NJM                    150      180
  3-00000000 I*4   NSTA      COMM          66       92       93       94       95       96       97       99
                                          111      143      176      192      197      207A     211      213
                                          215      221      223

  5-00000004 R*4   R1        COMM          68      212=
  6-00000000 R*4   R1N       COMM          69      220=
  9-0000000C R*4   R2        COMM          72      213=
  6-00000004 R*4   R2N       COMM          69      221=
  2-00000BA0 CHAR  REM                     61       83=      85      171=     172      173

      **     R*4   RR1                    136=     137      138      139      155      156      157      158
                                          159      160
      **     R*4   RR2                    137=     138      139      155      156      157      158      159
                                          160
  2-00000BF0 R*4   RSL                    103=     104      129      136      165A     172
  8-00000008 R*4   RSTAR     COMM          71      216
  7-00000000 R*4   SIG       COMM          70       84=      85      207A

  3-00000004 R*4   USTAR     COMM          66      217
```

10-000000B0	R*4	WM1N	COMM	73	222=
10-000000B4	R*4	WM2N	COMM	73	223=

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
10-00000000	R*4	B	COMM	44	(11)	73	96=	155=	178=	180=
						197				
10-0000002C	R*4	BK	COMM	44	(11)	73	97=	156=	182=	197
2-00000000	R*4	BKZ		396	(11, 9)	76	122=	156(2)	182	
2-00000AD4	R*4	CM		44	(11)	76	95=	167=	183=	197
						210	211	217		
2-0000018C	R*4	CMZ		396	(11, 9)	76	120=	162	183	
2-00000B58	R*4	CT1Z		36	(9)	76	115=	138(2)	141=	165A
						174				
2-00000B7C	R*4	CT2Z		36	(9)	76	116=	139(2)	162=	165A
						175				
2-00000318	R*4	QZ		396	(11, 9)	76	121=	145(2)	147(2)	150(2)
						152(2)	178(2)	180(2)		
4-00000000	R*4	R	COMM	44	(11)	67	93=	157=	184=	197
						212	213	216		
10-00000058	R*4	RHO	COMM	44	(11)	73	99=	158=	185=	193=
						197				
2-000004A4	R*4	RHOZ		396	(11, 9)	76	124=	158(2)	185	
10-00000084	R*4	RN	COMM	44	(11)	73	216=	220	221	
2-00000630	R*4	RZ		396	(11, 9)	76	118=	129	136(3)	141
						157(2)	172	184		
2-00000B2C	R*4	WMN		44	(11)	76	217=	222	223	
2-00000B00	R*4	XAX		44	(11)	76	92=	159=	186=	197
						207A				
2-000007BC	R*4	XAXZ		396	(11, 9)	76	117=	159(2)	186	
4-0000002C	R*4	XM	COMM	44	(11)	67	94=	160=	187=	197
2-00000948	R*4	XMZ		396	(11, 9)	76	119=	160(2)	187	

PARAMETER CONSTANTS

Type	Name	References			
I*4	NSTAMX	63#	67(2)	73(4)	76(10)
I*4	NSTLMX	63#	76(9)		

LABELS

Address	Label	References		
1-00000038	105'	85	86#	
1-00000085	132'	104	105#	
0-000005A8	140	129	134#	
0-000008B1	150	132	134	171#
0-00000A8E	160	169	189#	
1-0000012A	170'	197	199#	
1-000001DA	5110'	83	226#	

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	FOR$CLOSE	126
	FOR$OPEN	108
	MELLOR2	207
	SPLIN1	165

```
+-------------------------------------------------+
|              KEY TO REFERENCE FLAGS             |
|   =   - Value Modified                          |
|   #   - Defining Reference                      |
|   A   - Actual Argument, possibly modified      |
|   D   - Data Initialization                     |
|  (n)  - Number of occurrences on line           |
+-------------------------------------------------+
```

```
0001          SUBROUTINE MAP32
0002    C (3037-DSN3-MAP32) (L) MAPPING FROM 3D TO 2D
0003    C Refernced by MAIN (DSN3); references MTH$ATAND (twice)
0004    C To be compiled by VAX-11 FORTRAN, v.4.0
0005    C Coded 05MAR87; REVISED 01JUN88
0006
0007          COMMON/DI L / OMEGA
0008          COMMON/D JL / CT1, R1
0009          COMMON/D JLP/ R1N, R2N
0010          COMMON/D  L / BETA0, CL1CLM, WX0
0011          COMMON/D  LP/ BETA1, BETA2, TANB1, TANB2
0012          COMMON/  JL / CM1, CM2, CT2, R2
0013
0014          WX1       =R1N*CM1
0015          WX2       =R2N*CM2
0016          WX0       =(WX1+WX2)*.5
0017          WY1       =R1N*( R1*OMEGA-CT1 )
0018          TANB1     =WY1/WX0
0019          BETA1     =ATAND(TANB1)
0020          WY2       =R2N*( R2*OMEGA-CT2 )
0021          TANB2     =WY2/WX0
0022          BETA2     =ATAND(TANB2)
0023          WY0       =(WY1+WY2)*.5
0024          BETA0     =ATAND(WY0/WX0)
0025          WX0SQ     =WX0*WX0
0026          CL1CLM    =( WY0*WY0+WX0SQ ) / ( WY1*WY1+WX0SQ )
0027          WRITE(6,110) WX1, WX2, WX0, WY1, WY2, WY0, BETA1, BETA2, BETA0,
0028         +  (WX2-WX1)/WX1, CL1CLM
0029      110 FORMAT(//' AFTER MAPPED INTO X-Y PLANE,'/
0030         + '      WX1       WX2       WX0       WY1       WY2       WY0   ',
0031         + '     BETA1     BETA2     BETA0    ='/9F10.5/
0032         + ' VARIATION OF AXIAL VELOCITY RATIO, (WX2-WX1)/WX1, ='F10.5/
0033         + ' CL1/CLm RATIO ='F43.2)
0034
0035          RETURN
0036          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes								
0	$CODE	323	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
1	$PDATA	219	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT	LONG
3	DIL	4	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
4	DJL	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
5	DJLP	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
6	DL	12	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
7	DLP	16	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
8	JL	16	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
	Total Space Allocated	606									

ENTRY POINTS

Address	Type	Name	References
0-00000000		MAP32	1#

VARIABLES

Address	Type	Name	Attributes	References		
6-00000000	R*4	BETA0	COMM	10	24=	27
7-00000000	R*4	BETA1	COMM	11	19=	27
7-00000004	R*4	BETA2	COMM	11	22=	27
6-00000004	R*4	CL1CLM	COMM	10	26=	27
8-00000000	R*4	CM1	COMM	12	14	
8-00000004	R*4	CM2	COMM	12	15	
4-00000000	R*4	CT1	COMM	8	17	
8-00000008	R*4	CT2	COMM	12	20	
3-00000000	R*4	OMEGA	COMM	7	17	20
4-00000004	R*4	R1	COMM	8	17	
5-00000000	R*4	R1N	COMM	9	14	17

8-0000000C	R*4	R2	COMM	12	20					
5-00000004	R*4	R2N	COMM	9	15	20				
7-00000008	R*4	TANB1	COMM	11	18=	19				
7-0000000C	R*4	TANB2	COMM	11	21=	22				
6-00000008	R*4	WX0	COMM	10	16=	18	21	24	25(2)	27
**	R*4	WX0SQ		25=	26(2)					
**	R*4	WX1		14=	16	27(3)				
**	R*4	WX2		15=	16	27(2)				
**	R*4	WY0		23=	24	26(2)	27			
**	R*4	WY1		17=	18	23	26(2)	27		
**	R*4	WY2		20=	21	23	27			

LABELS

Address	Label	References	
1-00000000	110'	27	29#

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References		
R*4	MTH$ATAND	19	22	24

```
+----------------------------------------------+
|           KEY TO REFERENCE FLAGS             |
|    =   - Value Modified                      |
|    #   - Defining Reference                  |
|    A   - Actual Argument, possibly modified  |
|    D   - Data Initialization                 |
|   (n)  - Number of occurrences on line       |
+----------------------------------------------+
```

```
      SUBROUTINE MELLOR2( NSTA, SIG, XAX)

C (3037-DSN3-MELLOR2) (N) READ STORED TABLE OF G, H, B, & T FUNCTIONS
C REFERENCED BY INP2; REFERENCES MTH$ATAN & FOR$CLOSE & FOR$OPEN
C TO BE COMPILED BY VAX-11 FORTRAN, V. 4.0
C CODED BY W. CHIANG, 26MAR87; TOUCHED 19MAY87

C
C READ1101      TITLE
C
C READ1102      REMARK
C
C READ1103
C NFR   # OF FOURIER SERIES TERMS USING, e.g., NFR=3 TO HAVE A0, A1, & A2
C
C READ1104      REMARK
C
C READ1105
C CS    SOLIDITY
C AMDA  STARTING STAGGER ANGLE (FOR INTERPOLATION), in deg.
C AMDI  (CONSTANT) INTERVAL OF STAGGER ANGLE, in deg.
C AMDZ  ENDING STAGGER ANGLE, in deg. (AMDZ>AMDA)
C
C READ1106      REMARK
C
C READ1107
C NYS   # OF PTS. ALONG THE CHORD TO INPUT CAMBER FCP(XCC), FROM
C       (3037-MELLOR-INPUT)
C
C READ1108      REMARK
C
C READ1109
C XCC   X-COORDINATES OF CAMBER SLOPE FUNCTION FCPP, SIZE NYS
C
C READ1110
C FCP   CAMBER SLOPE (NORMALIZED BY CB, 'FCP' IN 'MELLOR,' AS A
C       FUNCTION OF XCC, SIZE NYS
C
C READ1111      REMARK
C
C READ1112
C TOC   THICKNESS DEVIDED BY CHORD LENGTH
C
C READ1113      REMARK (3 LINES)
C
C READ1114 (FOR AMDAD=AMDAA,AMDAZ,AMDAI)
C GJ    G FUNCTION
C HJ    H FUNCTION
C BJ    B FUNCTION
C TJ    T FUNCTION
C
C READ1115
C IDUM  SHOULD BE 999 TO CHECK THE END OF INPUT FILE
C

      PARAMETER ( NAMDAM=90, NFRMX=15, NYSMX=101, NSTAMX=11)

      COMMON/NO/ GJ(NAMDAM,NFRMX,NFRMX), HJ(NAMDAM,NFRMX,NFRMX),
     +                  BJ(NAMDAM,NFRMX), TJ(NAMDAM,NFRMX),
     +                  AMDAJ(NAMDAM), FCP(NYSMX), VEC(NFRMX),
     +                  XAXIS(NSTAMX), XCC(NYSMX),
     +                  AMDA, AMDI, AMDAMN, AMDAMX,
     +                  NAMDA, NFR, NFRSQ, NFRSQP, NYS, NYSM,
     +                  PI, PI2, TOC

      DIMENSION XAX(NSTA)

      OPEN( 11, FILE='MELO10', STATUS='OLD', READONLY)

C 100 INPUT DATA FROM UNIT 11
      READ (11,*)                                            READ1101
      READ (11,*)                                            READ1102
      READ (11,*) NFR                                        READ1103
      READ (11,*)                                            READ1104
      READ (11,*) CS, AMDA, AMDI, AMDZ                       READ1105
      IF (CS.NE.SIG) THEN
        PRINT *, 'CS.NE.SIG...', CS, SIG
```

```
            STOP 'MELLOR2.105'
            END IF
          READ (11,*)                                                       READ1106
          READ (11,*) NYS                                                   READ1107
          IF (NYS.GT.NYSMX) THEN
            PRINT *, 'NYS.GT.NYSMX...', NYS, NYSMX
            STOP 'MELLOR2.107'
            END IF
          READ (11,*)                                                       READ1108
          READ (11,*) ( XCC(N), N=1,NYS)                                    READ1109
          READ (11,*) ( FCP(N), N=1,NYS)                                    READ1110
          READ (11,*)                                                       READ1111
          READ (11,*) TOC                                                   READ1112
          IF (TOC.LT.0. .OR. TOC.GT..2) THEN
            PRINT *, 'TOC < 0 or > .2 ...', TOC
            STOP 'MELLOR2.109'
            END IF
          READ (11,5120)                                                    READ1113
          NAMDA     =0
          DO   AMDAD=AMDA,AMDZ,AMDI
            NAMDA   =NAMDA+1
            AMDAJ(NAMDA)    =AMDAD
            READ (11,*) ( ( GJ(NAMDA,N,K), K=1,NFR), N=1,NFR),
     +                  ( ( HJ(NAMDA,N,K), K=1,NFR), N=1,NFR),
     +                  ( BJ(NAMDA,N), N=1,NFR), ( TJ(NAMDA,N), N=1,NFR)    READ1114
            END DO
          READ (11,*) IDUM                                                  READ1115
          IF (IDUM.NE.999) THEN
            PRINT *, 'CHECK MELO10.DAT...', IDUM
            STOP 'MELLOR2.130'
            END IF
          CLOSE(11)

C 200 PREPARE CONSTANTS
          AMDAMN     =AMDA
          AMDAMX     =AMDAJ(NAMDA)
          NFRSQ      =NFR*NFR
          NFRSQP     =NFRSQ+1

C 300 PREPARE FCP
          NYSM       =NYS-1
          PI         =ATAN(1.) * 4.
          PI2        =PI+PI
          X0         =XAX(1)
          CHORD      =XAX(NSTA)-X0
          DO         I=1,NSTA
            XAXIS(I)=( XAX(I)-X0 )/CHORD
            END DO

          RETURN

C5000 FORMAT
 5120 FORMAT(//)
          END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1339	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	123	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	168	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3	NO	174124	PIC OVR REL GBL SHR NOEXE RD WRT LONG
	Total Space Allocated	175754	

ENTRY POINTS

Address	Type	Name	References
0-00000000		MELLOR2	1#

VARIABLES

Address	Type	Name	Attributes	References

3-0002A7F8	R*4	AMDA	COMM	58	75=	97	112			
**	R*4	AMDAD		97=	99					
3-0002A800	R*4	AMDAMN	COMM	58	112=					
3-0002A804	R*4	AMDAMX	COMM	58	113=					
3-0002A7FC	R*4	AMDI	COMM	58	75=	97				
2-00000004	R*4	AMDZ		75=	97					
**	R*4	CHORD		122=	124					
2-00000000	R*4	CS		75=	76	77				
**	I*4	I		123=	124(2)					
2-00000010	I*4	IDUM		104=	105	106				
**	I*4	K		100(4)=						
**	I*4	N		87(2)=	88(2)=	100(8)=				
3-0002A808	I*4	NAMDA	COMM	58	96=	98(2)=	99	100(4)	113	
3-0002A80C	I*4	NFR	COMM	58	73=	100(6)	114(2)			
3-0002A810	I*4	NFRSQ	COMM	58	114=	115				
3-0002A814	I*4	NFRSQP	COMM	58	115=					
AP-00000004@	I*4	NSTA		1	66	122	123			
3-0002A818	I*4	NYS	COMM	58	81=	82	83	87	88	118
3-0002A81C	I*4	NYSM	COMM	58	118=					
3-0002A820	R*4	PI	COMM	58	119=	120(2)				
3-0002A824	R*4	PI2	COMM	58	120=					
AP-00000008@	R*4	SIG		1	76	77				
3-0002A828	R*4	TOC	COMM	58	90=	91(2)	92			
**	R*4	XO		121=	122	124				

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
3-0002A300	R*4	AMDAJ	COMM	360	(90)	58	99=	113		
3-000278D0	R*4	BJ	COMM	5400	(90, 15)	58	100=			
3-0002A468	R*4	FCP	COMM	404	(101)	58	88=			
3-00000000	R*4	GJ	COMM	81000	(90, 15, 15)	58	100=			
3-00013C68	R*4	HJ	COMM	81000	(90, 15, 15)	58	100=			
3-00028DE8	R*4	TJ	COMM	5400	(90, 15)	58	100=			
3-0002A5FC	R*4	VEC	COMM	60	(15)	58				
AP-0000000C@	R*4	XAX		**	(*)	1	66	121	122	124
3-0002A638	R*4	XAXIS	COMM	44	(11)	58	124=			
3-0002A664	R*4	XCC	COMM	404	(101)	58	87=			

PARAMETER CONSTANTS

Type	Name	References			
I*4	NAMDAM	56#	58(5)		
I*4	NFRMX	56#	58(7)		
I*4	NSTAMX	56#	58		
I*4	NYSMX	56#	58(2)	82	83

LABELS

Address	Label	References	
1-00000078	5120'	95	130#

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	FOR$CLOSE	109
	FOR$OPEN	68
R*4	MTH$ATAN	119

```
      SUBROUTINE MELLOR3( ALPHAM, CB, STAG, NSTA, VNX, VNY, I23,
     +                           A0, A1, CLM)

C (3037-DSN3-MELLOR3) (O) CALC. CASCADE LIFT COEF. BASED ON ANGLE OF INCIDENT,
C        CAMBER, & VELOCITIES INDUCED BY VORTICES & SOURCES DUE TO 3D EFFECTS
C REFERENCED BY DSN2 & PFM3
C REFERENCES FILONC, SIMUL1, & SPLIN1, & MTH$COSD & MTH$SIND
C INPUT ALPHAM, CB, STAG, VNX, & VNY & ALL DATA IN COMMON STATEMENTS
C OUTPUT CLM
C CODED BY W. CHIANG, 27MAR87; REVISED 16APR87; TOUCHED 19MAY87

C ALPHAMMEAN ANGLE OF INCIDENT RELATIVE TO CHORD LINE
C CB    CAMBER
C NSTA  # OF STATIONS ALONG CHORD
C STAG  STAGGER ANGLE in deg.
C VNX   X-COMP. OF VELOCITIES (NORMALIZED TO USTAR) INDUCED BY VORTICES
C       & SOURCES DUE TO 3D EFFECTS; NOT USED IF I23=2
C VNY   Y-COMP. OF VELOCITIES (NORMALIZED TO USTAR) INDUCED BY VORTICES
C       & SOURCES DUE TO 3D EFFECTS; NOT USED IF I23=2
C I23   2 FOR 2D CASE (VNX=VNY=0); 3 FOR 3D CASE
C A0    CASCADE COEFS. A(1)
C A1    CASCADE COEFS. A(2)
C CLM   LIFT COEF. NORMALIZED TO MEAN RELATIVE VELOCITY

      PARAMETER ( NAMDAM=90, NFRMX=15, NYSMX=101, NSTAMX=11)

      EQUIVALENCE      ( F(1), VY(1))

      COMMON/NO/ GJ(NAMDAM,NFRMX,NFRMX), HJ(NAMDAM,NFRMX,NFRMX),
     +                  BJ(NAMDAM,NFRMX), TJ(NAMDAM,NFRMX),
     +                  AMDAJ(NAMDAM), FCP(NYSMX), V(NFRMX),
     +                  XAXIS(NSTAMX), XCC(NYSMX),
     +                  AMDA, AMDI, AMDAMN, AMDAMX,
     +                  NAMDA, NFR, NFRSQ, NFRSQP, NYS, NYSM,
     +                  PI, PI2, TOC

      DIMENSION A(NFRMX), DUMMY(NYSMX), F(NYSMX),
     +  GHV( NFRMX*(NFRMX+1) ),
     +  VEC(NFRMX), VNX(NSTA), VNY(NSTA), VX(NYSMX), VY(NYSMX)

C 100 FIND FUNCTION VALUES FOR PARTICULAR STAGGER ANGLE
      IF ( AMDAMN.GT.STAG .OR. AMDAMX.LT.STAG ) THEN
        PRINT *, 'CHECK STAG...', STAG, AMDAMN, AMDAMX
        STOP 'MELLOR3.101'
        END IF

      CO        =COSD(ALPHAM)
      L         =(STAG-AMDA)/AMDI + 1.00001
      LP        =L+1
      R2        =( AMDAJ(L)-STAG )/( AMDAJ(LP)-AMDAJ(LP) )
      R1        =1-R2

      IF (I23.EQ.2) THEN

        TEM     =-CB*CO
        DO     I=1,NYS
          F(I)  =FCP(I)*TEM
          END DO

       ELSE

        CALL     SPLIN1( NSTA, XAXIS, VNY, NYS, XCC, VY, DUMMY, DUM, 1.,
     +                  1.E-9)

        CALL     SPLIN1( NSTA, XAXIS, VNX, NYS, XCC, VX, DUMMY, DUM, 1.,
     +                  1.E-9)

        DO     I=1,NYS
          F(I)  =VY(I) - ( VX(I)+CO ) * FCP(I) * CB
          END DO

       END IF

      CALL      FILONC( NYSM, F, NFR, VEC)

      KN        =0
      DO        N=1,NFR
```

```
0078            GHV(N+NFRSQ)     =VEC(N)/PI - ( ( BJ(L,N)*R1+BJ(LP,N)*R2 )*CB +
0079         +                                 ( TJ(L,N)*R1+TJ(LP,N)*R2 )  )*TOC
0080            DO      K=1,NFR
0081              GHV(KN+K)      =( GJ(L,N,K)*R1+GJ(LP,N,K)*R2 ) -
0082         +                    ( HJ(L,N,K)*R1+HJ(LP,N,K)*R2 )*CB
0083              END DO
0084            KN=KN+NFR
0085            END DO
0086          GHV(NFRSQP)        =GHV(NFRSQP) + SIND(ALPHAM)
0087
0088    C 600           CALC. FOURIER COEFS. **********
0089
0090          CALL       SIMUL1( 3, NFR, GHV, A, 1.E-9, DETER)
0091
0092    C 700 CALC. LIFT COEF. CLM **********
0093          A0         =A(1)
0094          A1         =A(2)
0095          CLM        =(A0+A1)*PI2
0096
0097          RETURN
0098          END
```

PROGRAM SECTIONS

Name	Bytes	Attributes								
0 $CODE	928	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
1 $PDATA	36	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT	LONG
2 $LOCAL	2540	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	LONG
3 NO	174124	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
Total Space Allocated	177628									

ENTRY POINTS

Address	Type	Name	References
0-00000000	·	MELLOR3	1#

VARIABLES

Address	Type	Name	Attributes	References					
AP-00000020@	R*4	A0		1	93=	95			
AP-00000024@	R*4	A1		1	94=	95			
AP-00000004@	R*4	ALPHAM		1	47	86			
3-0002A7F8	R*4	AMDA	COMM	29	48				
3-0002A800	R*4	AMDAMN	COMM	29	42	43			
3-0002A804	R*4	AMDAMX	COMM	29	42	43			
3-0002A7FC	R*4	AMDI	COMM	29	48				
AP-00000008@	R*4	CB		1	55	69	78	81	
AP-00000028@	R*4	CLM		1	95=				
**	R*4	CO		47=	55	69			
2-00000900	R*4	DETER		90A					
2-000008F4	R*4	DUM		62A	65A				
**	I*4	I		56=	57(2)	68=	69(4)		
AP-0000001C@	I*4	I23		1	53				
**	I*4	K		80=	81(5)				
2-000008F8	I*4	KN		76=	81	84(2)=			
**	I*4	L		48=	49	50(2)	78(2)	81(2)	
**	I*4	LP		49=	50	78(2)	81(2)		
2-000008FC	I*4	N		77=	78(6)	81(4)			
3-0002A808	I*4	NAMDA	COMM	29					
3-0002A80C	I*4	NFR	COMM	29	74A	77	80	84	90A
3-0002A810	I*4	NFRSQ	COMM	29	78				
3-0002A814	I*4	NFRSQP	COMM	29	86(2)				
AP-00000010@	I*4	NSTA		1	37(2)	62A	65A		
3-0002A818	I*4	NYS	COMM	29	56	62A	65A	68	
3-0002A81C	I*4	NYSM	COMM	29	74A				
3-0002A820	R*4	PI	COMM	29	78				
3-0002A824	R*4	PI2	COMM	29	95				
**	R*4	R1		51=	78(2)	81(2)			
**	R*4	R2		50=	51	78(2)	81(2)		

AP-0000000C@	R*4	STAG		1	42(2)	43	48	50		
**	R*4	TEM		55=	57					
3-0002A828	R*4	TOC	COMM	29	78					

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
2-00000194	R*4	A		60	(15)	37	90A	93	94	
3-0002A300	R*4	AMDAJ	COMM	360	(90)	29	50(3)			
3-000278D0	R*4	BJ	COMM	5400	(90, 15)	29	78(2)			
2-000001D0	R*4	DUMMY		404	(101)	37	62A	65A		
2-00000000	R*4	F	EQUIV	404	(101)	27	37	57=	69=	74A
3-0002A468	R*4	FCP	COMM	404	(101)	29	57	69		
2-00000364	R*4	GHV		960	(240)	37	78=	81=	86(2)=	90A
3-00000000	R*4	GJ	COMM	81000	(90, 15, 15)	29	81(2)			
3-00013C68	R*4	HJ	COMM	81000	(90, 15, 15)	29	81(2)			
3-00028DE8	R*4	TJ	COMM	5400	(90, 15)	29	78(2)			
3-0002A5FC	R*4	V	COMM	60	(15)	29				
2-00000724	R*4	VEC		60	(15)	37	74A	78		
AP-00000014@	R*4	VNX		**	(*)	1	37	65A		
AP-00000018@	R*4	VNY		**	(*)	1	37	62A		
2-00000760	R*4	VX		404	(101)	37	65A	69		
2-00000000	R*4	VY	EQUIV	404	(101)	27	37	62A	69	
3-0002A638	R*4	XAXIS	COMM	44	(11)	29	62A	65A		
3-0002A664	R*4	XCC	COMM	404	(101)	29	62A	65A		

PARAMETER CONSTANTS

Type	Name	References		
I*4	NAMDAM	25#	29(5)	
I*4	NFRMX	25#	29(7)	37(4)
I*4	NSTAMX	25#	29	
I*4	NYSMX	25#	29(2)	37(4)

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References	
	FILONC	74	
R*4	MTH$COSD	47	
R*4	MTH$SIND	86	
	SIMUL1	90	
	SPLIN1	62	65

```
+---------------------------------------------+
|          KEY TO REFERENCE FLAGS             |
|   =    - Value Modified                     |
|   #    - Defining Reference                 |
|   A    - Actual Argument, possibly modified |
|   D    - Data Initialization                |
|  (n)   - Number of occurrences on line      |
+---------------------------------------------+
```

```
      SUBROUTINE PFM3
C (3037-DSN3-PFM3) (P) PERFORMANCE STUDY IN 3D FLOW
C Refernced by MAIN (DSN3)
C References MELLOR3 & MTH$ATAND (twice), MTH$COSD, & MTH$TAND
C Use statements with CRHO instead if density is not constant
C Use statements with CBO instead if B0 & B1 are allowed to vary during
C iteration
C To be compiled by VAX-11 FORTRAN, v.4.0
C Coded 09APR87; REVISED 16APR87; SLIGHTLY REVISED 15JUN87; TOUCHED 01JUN88

      PARAMETER ( NSTAMX=11)

      COMMON/DIJ P/ NSTA,       USTAR
      COMMON/DI  P/ EPSA,       IDBUG, MMAX
      COMMON/D JLP/ R1N, R2N
      COMMON/D J P/ SIG
      COMMON/D  LP/ BETA1, BETA2, TANB1, TANB2
      COMMON/D   P/ A0, A1, B0, CB, CHI, CLM, F, PHI, SIGH, STAG, TANB0
      COMMON/ I  P/ CONFC, CONFSO
      COMMON/  J P/ B(NSTAMX), BK(NSTAMX), RHO(NSTAMX), RN(NSTAMX),
     +                  WM1N, WM2N

      DIMENSION VNX(NSTAMX), VNY(NSTAMX), VNMUXK(NSTAMX), VNZEYK(NSTAMX)

C 100 CONSTANTS
      ALPHA0    =B0-STAG
      BRHO1     =B(1)
CRHO  BRHO1     =B(1)*RHO(1)
CBO   CON1      =SIG * .5 / COSD(B0)
      RSQMN     =(R1N*R1N+R2N*R2N)*.5
CBO
      SIGH      =SIG*.5
      XI        =(R2N*WM2N-R1N*WM1N)/PHI
      XIH       =XI*.5
      XIH1M     =1.-XIH

C 300 INITIAL VALUES
      M         =0
      BETA10    =BETA1
      CB0       =CB
      EPSB0     =BETA1-BETA2
      STAG0     =STAG
      PSIH      =(F+CHI)*PHI
      IF (IDBUG.GT.0)     WRITE(6,310) TANB1, TANB2

  310 FORMAT(///' * SUB PFM3 *'//' TANB1 & TANB2 ='2X2F9.5)
      WRITE(6,320) PHI, PSIH+PSIH, CHI, XI, CHI/F, PSIH*USTAR*USTAR
  320 FORMAT(//
     +  ' LOCAL FLOW COEFFICIENT, PHI, ='F20.5/
     +  ' LOCAL ENTHALPY RISE COEFFICIENT, PSI, =',F11.5/
     +  ' STREAM INCLINATION PARAMETER, CHI, ='F14.5/
     +  ' AXIAL VELOCITY VARIATION PARAMETER, XI, ='F9.5/
     +  ' Chi/F RATIO (MAGNITUDE SHOULD BE NO HIGHER THAN .2 TO .3) ='
     +  F9.5/
     +  ' RISE OF TOTAL ENTHALPY, del H, ='F18.5///
     +  30X'Lift   Cir.para. Turnning'23X
     +  'Stagger    delta'/
     +  '     M      A0        A1      coeff.      F       Angle     ',
     +  'Beta 1       Cb       Angle   T. Angle'/)
      IF (IDBUG.GT.0)
     +  WRITE(6,7110) 0, A0, A1, CLM, F, EPSB0, BETA10, CB, STAG
      IF (MMAX.GT.100) THEN
        PRINT *, 'MMAX ALLOWED TO BE > 100?  CHECK...', MMAX
        STOP 'PFM3.290'
        END IF

      DO        I=1,NSTA
        VNZEYK(I)       =( RN(I)*RN(I)-RSQMN )/PHI
        VNMUXK(I)       =( BRHO1/  B(I)          -1. )*XIH1M/BK(I) - XIH
CRHO    VNMUXK(I)       =( BRHO1/( B(I)*RHO(I) )-1. )*XIH1M/BK(I) - XIH
        END DO

C 400 START AN ITERATION PROCESS WITH NEW CB & STAG
  410 TANS      =TAND(STAG)
      Q         =1.+TANB0*TANS
      DO        I=1,NSTA
        VNZEY   =VNZEYK(I)/Q
```

```
           VNMUX   =VNMUXK(I)/Q
           VNX(I)  =VNZEY*TANS + VNMUX
           VNY(I)  =VNZEY - VNMUX*TANS
           END DO

C1000 FIND A0, A1, & CIRculation

      CALL    MELLOR3( B0-STAG, CB, STAG, NSTA, VNX, VNY, 3, A0, A1,
CB0   CALL    MELLOR3(  ALPHA0, CB, STAG, NSTA, VNX, VNY, 3, A0, A1,
     +                  CLM)

C2000 ADJUST IF NECESSARY
      M       =M+1
CB0   F       =CLM * CON1           - CHI
      F       =CLM * SIGH / COSD(B0) - CHI
      TANB1   =TANB2+F
      BETA1   =ATAND(TANB1)
      EPSBM   =BETA1-BETA2
      DEPS    =EPSB0-EPSBM
CCC   DBETA1  =BETA1-BETA10
CCC   IF (ABS(DEPS) .GT. EPSA  .OR.  ABS(DBETA1) .GT. EPSA) THEN
      IF (ABS(DEPS) .GT. EPSA) THEN

        IF (M.GT.MMAX) THEN
          PRINT *, 'ITERATION # EXCEEDED ', MMAX
          STOP 'PFM3.2199'
          END IF

        CB      =( DEPS*CONFC/EPSBM +1. ) * CB
CCC     STAG    =STAG0+(DBETA1-DEPS*.5)*CONFSO
        STAG    =STAG0-DEPS*CONFSO
CB0     TANB0   =TANB1-F*.5
CB0     B0      =ATAND(TANB0)
        IF (IDBUG.GT.0)     WRITE(6,7110) M, A0, A1, CLM, F, EPSBM,
     +                                   BETA1, CB, STAG, DEPS
        GOTO 410
       END IF

      WRITE(6,7110) M, A0, A1, CLM, F, EPSBM, BETA1, CB, STAG, DEPS
      Q =CB-CB0
      WRITE(6,2020) Q, Q*100./CB0
 2020 FORMAT(//' Cb IS INCREASED BY'F7.3', OR'F6.1'%'//)

      RETURN

 7110 FORMAT(I5,9F10.5)
      END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes								
0	$CODE	1252	PIC	CON	REL	LCL	SHR	EXE	RD	NOWRT	LONG
1	$PDATA	610	PIC	CON	REL	LCL	SHR	NOEXE	RD	NOWRT	LONG
2	$LOCAL	260	PIC	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	LONG
3	DIJP	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
4	DIP	12	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
5	DJLP	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
6	DJP	4	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
7	DLP	16	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
8	DP	44	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
9	IP	8	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
10	JP	184	PIC	OVR	REL	GBL	SHR	NOEXE	RD	WRT	LONG
	Total Space Allocated	2406									

ENTRY POINTS

Address	Type	Name	References
0-00000000		PFM3	1#

VARIABLES

Address	Type	Name	Attributes	References

8-00000000	R*4	A0	COMM	18	60	85A	111	116			
8-00000004	R*4	A1	COMM	18	60	85A	111	116			
**	R*4	ALPHA0		26=							
8-00000008	R*4	B0	COMM	18	26	85	92				
7-00000000	R*4	BETA1	COMM	17	39	41	94=	95	111	116	
**	R*4	BETA10		39=	60						
7-00000004	R*4	BETA2	COMM	17	41	95					
**	R*4	BRHO1		27=	69						
8-0000000C	R*4	CB	COMM	18	40	60	85A	106(2)=	111	116	117
2-000000B4	R*4	CB0		40=	117	118					
8-00000010	R*4	CHI	COMM	18	43	47(2)	92				

8-00000014	R*4	CLM	COMM	18	60	85A	92	111	116		
9-00000000	R*4	CONFC	COMM	19	106						
9-00000004	R*4	CONFSO	COMM	19	108						
**	R*4	DEPS		96=	99	106	108	111	116		
4-00000000	R*4	EPSA	COMM	14	99						
**	R*4	EPSB0		41=	60	96					
**	R*4	EPSBM		95=	96	106	111	116			
8-00000018	R*4	F	COMM	18	43	47	60	92=	93	111	116
**	I*4	I		67=	68(3)	69(3)	76=	77	78	79	80
4-00000004	I*4	IDBUG	COMM	14	44	60	111				
2-000000B0	I*4	M		38=	90(2)=	101	111	116			
4-00000008	I*4	MMAX	COMM	14	62	63	101	102			
3-00000000	I*4	NSTA	COMM	13	67	76	85A				
8-0000001C	R*4	PHI	COMM	18	33	43	47	68			
**	R*4	PSIH		43=	47(3)						
**	R*4	Q		75=	77	78	117=	118(2)			
5-00000000	R*4	R1N	COMM	15	30(2)	33					
5-00000004	R*4	R2N	COMM	15	30(2)	33					
**	R*4	RSQMN		30=	68						
6-00000000	R*4	SIG	COMM	16	32						
8-00000020	R*4	SIGH	COMM	18	32=	92					
8-00000024	R*4	STAG	COMM	18	26	42	60	74	85(2)A	108=	111
				116							
**	R*4	STAG0		42=	108						
8-00000028	R*4	TANB0	COMM	18	75						
7-00000008	R*4	TANB1	COMM	17	44	93=	94				
7-0000000C	R*4	TANB2	COMM	17	44	93					
**	R*4	TANS		74=	75	79	80				
3-00000004	R*4	USTAR	COMM	13	47(2)						
**	R*4	VNMUX		78=	79	80					
**	R*4	VNZEY		77=	79	80					
10-000000B0	R*4	WM1N	COMM	20	33						
10-000000B4	R*4	WM2N	COMM	20	33						
**	R*4	XI		33=	34	47					
**	R*4	XIH		34=	35	69					
**	R*4	XIH1M		35=	69						

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References		
10-00000000	R*4	B	COMM	44	(11)	20	27	69
10-0000002C	R*4	BK	COMM	44	(11)	20	69	

10-00000058	R*4	RHO	COMM	44	(11)	20		
10-00000084	R*4	RN	COMM	44	(11)	20	68(2)	
2-00000058	R*4	VNMUXK		44	(11)	23	69=	78
2-00000000	R*4	VNX		44	(11)	23	79=	85A
2-0000002C	R*4	VNY		44	(11)	23	80=	85A

2-00000084	R*4	VNZEYK	44	(11)	23	68=	77

PARAMETER CONSTANTS

Type	Name	References		
I*4	NSTAMX	11#	20(4)	23(4)

LABELS

Address	Label	References			
1-00000051	310'	44	46#		
1-0000007F	320'	47	48#		
0-00000282	410	74#	113		
1-00000231	2020'	118	119#		
1-0000025A	7110'	60	111	116	123#

FUNCTIONS AND SUBROUTINES REFERENCED

Type	Name	References
	MELLOR3	85
R*4	MTH$ATAND	94
R*4	MTH$COSD	92
R*4	MTH$TAND	74

```
+--------------------------------------------+
|          KEY TO REFERENCE FLAGS            |
|   =    - Value Modified                    |
|   #      Defining Reference                |
|   A    - Actual Argument, possibly modified|
|   D    - Data Initialization               |
|  (n)   - Number of occurrences on line     |
+--------------------------------------------+
```

```
        SUBROUTINE SIMUL1( INDIC, N, A, X, EPS, DETER)

C (3037-MELLOR-SIMUL1) SIMULTANEOUS LINEAR EQS. SOLVER WHICH USES GAUSS-
C                     JORDAN ELIMINATION METHOD WITH THE MAX. PIVOT STRATEGY
C VS.1: MATRIX IN 1-D FORM
C REFERENCED BY LOOP (TWICE); REFERENCES MTH$MOD
C OR, REFERENCED BY SIMUL1T
C INPUT A, EPS, INDIC, & N; OUTPUT A, DETER & X
C 04NOV86 ADAPTED FROM (3037-MELLOR-SIMUL); WHICH WAS ADAPTED, ON 15OCT86,
C FROM S. MAEKAWA WHO ADOPTED FROM CARNAHAN, et al., 1969, 290-1
C REVISED 01DEC86

C A     AUGMENTED MATRIX OF COEFS.; W/ COEFF. MATRIX IN THE 1ST N COLS.
C       & VECTOR OF RIGHT HAND SIDE IN THE (N+1)TH COL.; IN THE FORM OF
C       ( N x N+1 ) ARRAY
C DETER DETERMINANT OF THE ORIGINAL COEF. MATRIX
C EPS   MIN. ALLOWABLE MAGNITUDE FOR A PIVOT ELEMENT
C INDIC COMPUTATIONAL SWITCH:
C       <0, TO COMPUTE THE INVERSE OF THE MATRIX IN PLACE
C       =0, TO COMPUTE N SOLS. CORRESPONDING TO THE SET OF LINEAR EQS.
C           W/ AUGMENTED MATRIX OF COEFS. IN THE N BY N+1 ARRAY A AND IN
C           ADDITION TO COMPUTE AS WHEN INDIC<0
C       >0, TO SOLVE THE SET OF EQS. BUT THE INVERSE IS NOT COMPUTED IN
C           PLACE
C N     # OF ROWS IN A MATRIX
C X     VECTOR OF SOLUTIONS, SIZE N

      PARAMETER (NMAX=11)

      DIMENSION A( N*(N+1) ), IROW(NMAX), JCOL(NMAX), JORD(NMAX), X(N),
     +          Y(NMAX)

      IF (N.GT.NMAX) THEN
        PRINT *, 'N > NMAX...', N, NMAX
        STOP 'SIMUL.10'
        END IF

      IF (INDIC.LT.0) THEN
        MAX     =N
       ELSE
        MAX     =N+1
       END IF

      DETER     =1
      DO  200  K=1,N
        PIVOT   =0.
        IF (K.EQ.1) THEN
          DO 60I=1,N
            IJ  =I
            DO   60   J =1,N
              IF ( ABS( A(IJ) ) .GT. ABS(PIVOT) ) THEN
                PIVOT   =A(IJ)
                IROW(1) =I
                JCOL(1) =J
                END IF
 60           IJ=IJ+N
C 60        CONTINUE
         ELSE
          KM1   =K-1
          DO170I=1,N
            IJ  =I
            DO      ISCAN=1,KM1
              IF ( I .EQ. IROW(ISCAN) )  GO TO 170
              END DO
            DO   160   J=1,N
              DO   JSCAN=1,KM1
                IF ( J .EQ. JCOL(JSCAN) )  GO TO 160
                END DO
              IF ( ABS( A(IJ) ) .GT. ABS(PIVOT) ) THEN
                PIVOT   =A(IJ)
                IROW(K) =I
                JCOL(K) =J
                END IF
160           IJ=IJ+N
170         CONTINUE
          END IF
```

```
    IF ( ABS(PIVOT) .LT. EPS ) THEN
     PRINT *, '...PIVOT < EPS...', PIVOT, EPS
     STOP 'SIMUL.55'
     END IF

    IROWK   =IROW(K)
    JCOLK   =JCOL(K)
    JN      =(JCOLK-1)*N
    DETER   =DETER*PIVOT
    IJ      =IROWK
    DO   J =1,MAX
     A(IJ) =A(IJ)/PIVOT
     IJ    =IJ+N
     END DO
    A(IROWK+JN)     =1./PIVOT
    IJC     =JN
    DO     I=1,N
     IJC   =IJC+1
     IF (I.NE.IROWK) THEN
      AIJCK       =A(IJC)
      A(IJC)      =-AIJCK/PIVOT
      IJ  =I
      IRMI=IROWK-I
      DO J=1,MAX
       IF (J.NE.JCOLK)      A(IJ)=A(IJ)-AIJCK*A(IJ+IRMI)
       IJ          =IJ+N
       END DO
      END IF
     END DO
 200    CONTINUE

   JN          =(MAX-1)*N
   DO       I =1,N
    IROWI   =IROW(I)
    JCOLI   =JCOL(I)
    JORD(IROWI)     =JCOLI
CCC IF (INDIC.GE.0)      X(JCOLI)=A(IROWI+JN)
    X(JCOLI)=A(IROWI+JN)
    END DO
   INTCH      =0
   DO        I=1,N-1
    IP1      =I+1
    DO     J =IP1,N
     IF ( JORD(J).LT.JORD(I) ) THEN
      JTEMP        =JORD(J)
      JORD(J)      =JORD(I)
      JORD(I)      =JTEMP
      INTCH        =INTCH+1
      END IF
     END DO
    END DO

   IF ( MOD(INTCH,2) .GT. 0 )      DETER=-DETER
   IF (INDIC.GT.0) GOTO 180

   JN          =0
   DO       J =1,N
    DO     I =1,N
     Y( JCOL(I) )  =A( IROW(I)+JN )
     END DO
    DO     I =1,N
     A(I+JN)       =Y(I)
     END DO
    JN       =JN+N
    END DO
   DO        I =1,N
    DO     J =1,N
     Y( IROW(J) )  =A( I + ( JCOL(J)-1 )*N )
     END DO
    IJ       =I
    DO     J =1,N
     A(IJ) =Y(J)
     IJ    =IJ+N
     END DO
    END DO

 180 RETURN
   END
```

PROGRAM SECTIONS

	Name	Bytes	Attributes
0	$CODE	1275	PIC CON REL LCL SHR EXE RD NOWRT LONG
1	$PDATA	48	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2	$LOCAL	336	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
	Total Space Allocated	1659	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SIMUL1	1#

VARIABLES

Address	Type	Name	Attributes	References							
**	R*4	AIJCK		97=	98	102					
AP-00000018@	R*4	DETER		1	44=	86(2)=	130(2)=				
AP-00000014@	R*4	EPS		1	78	79					
**	I*4	I		48=	49	53	60=	61	63	71	94=
				96	99	100	110=	111	112	118=	119
				121	123	124	135=	136(2)	138=	139(2)	143=
				145	147						
**	I*4	IJ		49=	51	52	56(2)=	61=	69	70	74(2)=
				87=	89(2)	90(2)=	99=	102(3)	103(2)=	147=	149
				150(2)=							
**	I*4	IJC		93=	95(2)=	97	98				
AP-00000004@	I*4	INDIC		1	38	131					
**	I*4	INTCH		117=	125(2)=	130					
**	I*4	IP1		119=	120						
**	I*4	IRMI		100=	102						
**	I*4	IROWI		111=	113	115					
**	I*4	IROWK		83=	87	92	96	100			
**	I*4	ISCAN		62=	63						
**	I*4	J		50=	54	65=	67	72	88=	101=	102
				120=	121	122	123	134=	144=	145(2)	148=
				149							
**	I*4	JCOLI		112=	113	115					
2-000000C4	I*4	JCOLK		84=	85	102					
**	I*4	JN		85=	92	93	109=	115	133=	136	139
				141(2)=							
**	I*4	JSCAN		66=	67						
**	I*4	JTEMP		122=	124						
2-00000084	I*4	K		45=	47	59	71	72	83	84	
**	I*4	KM1		59=	62	66					
2-00000080	I*4	MAX		39=	41=	88	101	109			
AP-00000008@	I*4	N		1	30(3)	33	34	39	41	45	48
				50	56	60	65	74	85	90	94
				103	109	110	118	120	134	135	138
				141	143	144	145	148	150		
**	R*4	PIVOT		46=	51	52=	69	70=	78	79	86
				89	92	98					

ARRAYS

Address	Type	Name	Attributes	Bytes	Dimensions	References				
AP-0000000C@	R*4	A		**	(*)	1	30	51	52	69
						70	89(2)=	92=	97	98=
						102(3)=	115	136	139=	145
						149=				
2-00000000	I*4	IROW		44	(11)	30	53=	63	71=	83
						111	136	145		
2-0000002C	I*4	JCOL		44	(11)	30	54=	67	72=	84
						112	136	145		
2-00000058	I*4	JORD		44	(11)	30	113=	121(2)	122	123(2)=
						124=				
AP-00000010@	R*4	X		**	(*)	1	30	115=		
2-00000084	R*4	Y		44	(11)	30	136=	139	145=	149

PARAMETER CONSTANTS

Type	Name	References			
I*4	NMAX	28#	30(4)	33	34

LABELS

Address	Label	References		
**	60	48	50	56#
0-000001CB	160	65	67	74#
0-000001D3	170	60	63	75#
0-000004FA	180	131	154#	
**	200	45	107#	

```
+---------------------------------------------+
|          KEY TO REFERENCE FLAGS             |
|   =   - Value Modified                      |
|   #   - Defining Reference                  |
|   A   - Actual Argument, possibly modified  |
|   D   - Data Initialization                 |
|  (n)  - Number of occurrences on line       |
+---------------------------------------------+
```

```
      SUBROUTINE SPLIN1( N, X, Y, NARG, DOMAIN, FUNC, DERIV, SUM, SEND,
     *                          EPS)

C (3037-DSN3-SPLIN1) (S) CUBIC SPLINE INTERPOLATION (AND INTEGRATION)
C (3037-MELLOR-SPLIN1)   CUBIC SPLINE INTERPOLATION (AND INTEGRATION)
C Version 1 - BASIC SPLINE SUBROUTINE
C SEE ALSO Version 2 - INCLUDING SINGLE ARGUMENT INTERPOLATION &
C       EXTRAPOLATION
C SEE ALSO Version 3 - INTEGRATION TO EVERY PT.
C REFERENCED BY (3037-DSN3-DSN2) & (3037-DSN3-MELLOR3) (TWICE)
C REFERENCED BY (3037-MELLOR-INPUT) (4 TIMES) & (3037-MELLOR-LOOP) (4 TIMES)
C REFERENCES NONE
C INPUT EPS, DOMAIN(*), N, NARG, X(*), AND Y(*)
C OUTPUT DERIV(*), FUNC(*), AND SUM...
C ADAPTED FROM (8001-SPLIN1), 24OCT86
C SLIGHTLY REVISED 20MAY87; INCREASED DIM 24MAY88

C DERIV OUTPUT VECTOR (SIZE NARG) CONTAINING DERIVATIVE VALUES FOR
C       DOMAIN(*)
C DOMAININPUT VECTOR (SIZE NARG) CONTAINING DOMAIN VALUES FOR WHICH THE
C       DERIVATIVE OR FUCTIONAL VALUE IS TO BE COMPUTED
C EPS   ERROR TOLERANCE IN ITERATIVE SOL. OF SIMUL. EQS.
C FUNC  OUTPUT VECTOR (SIZE NARG) CONTAINING INTERPOLATED FUNCTION
C       VALUES FOR DOMAIN(*)
C MXN   MAX. VALUE OF N
C N     # OF DATA PTS.; 2<N<MXN
C NARG  # OF ARGUMENTS FOR WHICH FUNC(*) &/OR DERIV(*) ARE REQ'D.;
C       POSITIVE IF THE INTEGRAL IS NEEDED; NEG. IF THE INTEGRAL IS NOT
C       NEEDED; 0 IF ONLY THE INTEGRAL IS NEEDED (NEITHER FUNC NOR DERIV
C       IS NEEDED); ABS(NARG).LE.NARGMX
C NARGMXMAX. VALUE OF ABS(NARG)
C OMEGA RELAXATION FACTOR FOR SUCCESSIVE OVER-RELAXATION
C SEND  A FACTOR TO BE APPLIED TO S(2) & S(N-1) TO OBTAIN S(1) & S(N),
C       RESPECTIVELY, S BEING CURVATURE; NORMALLY 0, .5, OR 1
C SUM   INTEGRAL
C X     VECTOR (SIZE N) CONTAINING DOMAIN VALUES OF THE DATA PTS. [X(i)
C       VALUES SHOULD BE IN ACCENDING ORDER]
C Y     VECTOR (SIZE N) CONTAINING RANGE VALUES OF THE DATA PTS.

      PARAMETER ( MXN=100, NARGMX=201)

      DIMENSION DERIV(NARGMX), DOMAIN(NARGMX), DX(MXN), DYDX(MXN),
     *          FUNC(NARGMX), G(MXN), S(MXN), WORK(MXN), X(N), Y(N)

      DATA      MNITER/3/, MXITER/50/, OMEGA/1.071796768/
C                                     OMEGA=8.-4.*SQRT(3.)

C 100 CHECK
      IF (N.GT.MXN .OR. N.LT.3 .OR. ABS(NARG).GT.NARGMX) THEN
        PRINT *, 'CHECK N OR NARG...', N, NARG, MXN, NARGMX
        STOP 'SPLINE.111'
        END IF

C 200 DYDX
      NM1       =N-1
      DO        I=1,NM1
        DX(I)   =X(I+1)-X(I)
        DYDX(I) =( Y(I+1)-Y(I) )/DX(I)
        END DO

C 300 S & G
      DO        I=2,NM1
        DX2     =X(I+1)-X(I-1)
        WORK(I) =DX(I-1)/DX2
        DYDX2H  =( DYDX(I)-DYDX(I-1) )/DX2
        S(I)    =DYDX2H+DYDX2H
        G(I)    =S(I)+DYDX2H
        END DO
      DO        I=2,NM1
        WORK(I) =WORK(I)*.5
        END DO
      S(1)      =S(2)*SEND
      S(N)      =S(N-1)*SEND

      ITER      =0
  330 ETA       =0.
      ITER      =ITER+1
```

```
        DO      I=2,NM1
          TEM     =( G(I) - WORK(I)*S(I-1) - ( .5-WORK(I) )*S(I+1)
     *                  - S(I) ) * OMEGA
          ETA     =AMAX1( ABS(TEM), ETA)
          S(I)    =S(I)+TEM
          END DO
          IF (ITER.LT.MNITER) GOTO 330
          IF (ETA.GT.EPS) THEN
            IF (ITER.LT.MXITER) GOTO 330
            DO   I=1,N
              WRITE(6,*) I, X(I), Y(I), G(I), S(I), WORK(I)
              END DO
            PRINT *, 'NOT CONVERGED... ', MXITER, ETA, EPS
            STOP 'SPLINE.344'
            END IF
          IF (NARG.EQ.0) GOTO 520
        DO      I=1,NM1
          G(I)    =( S(I+1)-S(I) )/DX(I)
          END DO
C 400 CALC. FUNCTION VALUES AND DERIVATIVES
        DO      J=1,ABS(NARG)
          I       =1
          DOM     =DOMAIN(J)
          IF ( X(1).GT.DOM .OR. X(N).LT.DOM) THEN
            PRINT *, 'ARGUMENT OUT OF RANGE... ', J, N, DOM, X(1), X(N)
            STOP 'SPLINE.411'
            END IF
    430   I       =I+1
            IF ( X(I).LT.DOM ) GOTO 430
          I       =I-1
          H       =DOM-X(I)
          T       =DOM-X(I+1)
          HT      =H*T
          DSQS    =( G(I)*H+S(I)+S(I)+S(I+1) )/6.
          FUNC(J) =DYDX(I)*H + HT*DSQS +Y(I)
          DERIV(J)=(H+T)*DSQS + G(I)*HT/6 + DYDX(I)
          END DO
C 500 INTEGRATE FROM X(1) TO X(N)
        IF (NARG.LT.0) GOTO 990
    520 SUM     =0.
        DO      J=1,NM1
          H       =DX(J)
          SUM     =( (Y(J)+Y(J+1))*.5 - ( S(J) + S(J+1) )*H*H/24. )*H + SUM
          END DO
C 900
    990 RETURN
        END
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 $CODE	1648	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 $PDATA	98	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 $LOCAL	2232	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
Total Space Allocated	3978	

ENTRY POINTS

Address	Type	Name	References
0-00000000		SPLIN1	1#

VARIABLES

Address	Type	Name	Attributes	References

```
      **        R*4  DOM                     104=     105(2)   106      110      112      113
      **        R*4  DSQS                    115=     116      117
      **        R*4  DX2                      63=      64       65
      **        R*4  DYDX2H                   65=      66(2)    67
  AP-00000028@  R*4  EPS                       1       86       91

      **        R*4  ETA                      76=      81(2)=   86       91
      **        R*4  H                       112=     114      115      116      117      125=     126(3)
      **        R*4  HT                      114=     116      117
      **        I*4  I                        56=      57(3)    58(4)    62=      63(2)    64(2)    65(2)    66
                                              67(2)    69=      70(2)    78=      79(6)    82(2)    88=      89(6)
                                              97=      98(4)   103=     109(2)=  110      111(2)=  112      113
                                             115(4)   116(2)   117(2)
      **        I*4  ITER                     75=      77(2)=   85       87

      **        I*4  J                       102=     104      106      116      117      124=     125      126(4)
   2-000007D0   I*4  MNITER                   45D      85
   2-000007D4   I*4  MXITER                   45D      87       91
  AP-00000004@  I*4  N                         1       42(2)    49(2)    50       55       73(2)    88      105
                                             106(2)
  AP-00000010@  I*4  NARG                      1       49       50       95      102      121

   2-000007DC   I*4  NM1                      55=      56       62       69       78       97      124
   2-000007D8   R*4  OMEGA                    45D      79
  AP-00000024@  R*4  SEND                      1       72       73
  AP-00000020@  R*4  SUM                       1      123=     126(2)=
      **        R*4  T                       113=     114      117

      **        R*4  TEM                      79=      81       82
```

ARRAYS

```
    Address   Type  Name            Attributes          Bytes  Dimensions          References

  AP-0000001C@  R*4  DERIV                                804  (201)                 1        42       117=
  AP-00000014@  R*4  DOMAIN                               804  (201)                 1        42       104
   2-00000000   R*4  DX                                   400  (100)                42        57=       58       64       98
                                                                                   125
   2-00000190   R*4  DYDX                                 400  (100)                42        58=       65(2)   116      117
  AP-00000018@  R*4  FUNC                                 804  (201)                 1        42       116=

   2-00000320   R*4  G                                    400  (100)                42        67=       79       89       98=
                                                                                   115       117
   2-000004B0   R*4  S                                    400  (100)                42        66=       67       72(2)=   73(2)=
                                                                                    79(3)     82(2)=    89       98(2)   115(3)
                                                                                   126(2)
   2-00000640   R*4  WORK                                 400  (100)                42        64=       70(2)=   79(2)    89
  AP-00000008@  R*4  X                                     **  (*)                   1        42        57(2)    63(2)    89
                                                                                   105(2)    106(2)    110      112      113
  AP-0000000C@  R*4  Y                                     **  (*)                   1        42        58(2)    89      116
                                                                                   126(2)
```

PARAMETER CONSTANTS

```
  Type  Name                                   References

   I*4  MXN                                      40#      42(5)    49       50
   I*4  NARGMX                                   40#      42(3)    49       50
```

LABELS

```
    Address    Label                           References

  0-00000258   330                               76#      85       87
  0-000004F8   430                              109#     110
  0-000005F3   520                               95      123#
  0-0000066F   990                              121      130#
```

APPENDIX C

BRIEF INSTRUCTIONS ON SAMPLE INPUT DATA

C.1 PROGRAM SCM

The following is a nomenclature of the input data used in Program SCM.

```
IBFLOW=-2 IF G, CM(*), R(*) & Q(*) ARE INPUT DATA;
      =-1 FOR NONUNIFORM INFLOWS;
      = 0 FOR UNIFORM INFLOW;
      = 1 IF DATA ARE AVAILABLE FROM BLADE TO BLADE CALCULATIONS;
      =10 TO USE SUB BBFLOW TO PREPARE INPUT DATA.
IDSN3 = 0 FOR NO ACTION;
      = 1 TO STORE A PART OF RESULTS TO DSN3ZI.DAT TO BE USED BY
          (3037-DSN3)
IN    LOGICAL UNIT # FOR INPUT FILE
LG    = 0 FOR SEA WATER (DEN=1025 KG/CU.M.);
      = 1 FOR PURE WATER (DEN=1000);
      = 2 FOR GAS.

READ1  (A DUMMY LINE TO STORE TITLE, NOTE...)

READ2  NM      # OF MERIDIONAL STREAM LINES, >=5
       NQ      # OF Q-LINES, >=5
       NQI     LINE # OF Q-LINE AT THE LINE PRIOR TO THE
               LEADING EDGE OF BLADE
       NQB     # OF Q-LINES ON THE BLADE
       ICM     = 1 TO SOLVE D(CM)/DQ (INOUE);
               = 2 TO SOLVE D(CM*CM)/DQ

READ3  OMG     ANGULAR VEL. in rad/sec
       RAS     REFERENCE RADIUS in m
       DMPCM   DAMPING FACTOR USED IN CM ITERATION, INOUE 1.
       DMPG    DAMPING FACTOR USED IN G ITERATION, INOUE .5
       DMPL    DAMPING FACTOR USED IN Q ITERATION, INOUE .1
       EPSCM   CONVERGING CRITERION FOR CM ITERATION, INOUE 1.E-6
       EPSG    CONVERGING CRITERION FOR G ITERATION, INOUE 1.E-4
       EPSL    CONVERGING CRITERION FOR Q ITERATION, INOUE 1.E-4

READ4  RUMRQ   ANGLE BETWEEN Q-LINE (STRAIGHT) & RADIUS DIR.,
               POSITIVE C.W., in rad.

READ5  Z       Z-COORINATE in m

READ6  PHID1   ANGLE BETWEEN HUB & Z-ZXIS in deg
       PHIDN   ANGLE BETWEEN CASING & Z-ZXIS in deg
```

```
READ7   SM       M-COORDINATE in m

READ8   CTH      PERIPHERAL VEL. in m/sec

READ9   BLO      BLOCKAGE FACTOR, Kb=(theta2-theta1)/(2*pi/N), 0 TO
                 1.,
                 = 1 IF NO BLOCKAGE

***** READ10 & READ11 ARE REQUIRED ONLY IF LG=2 *****
READ10 DEN       DENSITY in kg/cu.m.

READ11 ENT       CHANGE OF ENTROPY in J/kg

***** READ12 TO READ15 ARE REQUIRED ONLY IF IBFLOW=-2 *****
READ12 GS        CUMULATIVE MASS FLOW RATE IN FLOW-TUBES, GS(1) TO
                 DESIGNATES THAT BN. J=1 & J=2, GS(NM-1)=G, in
kg/sec
       G         MASS FLOW RATE in kg/sec

READ13 CM        MERIDIONAL VEL. in m/sec

READ14 R         RADIAL DISTANCE FROM AXIS OF RATATION in m

READ15 H         ENTHALPY in J/kg or (m/sec)**2

***** READ16 TO READ21 ARE REQUIRED ONLY IF IBFLOW=-1 *****
READ16 R         RADIAL DISTANCE FROM AXIS OF RATATION in m

READ17 NCMNU     # OF INPUT DATA FOR INFLOW VEL. PROFILE

READ18 RNUC1     1ST CONVERSION FACTOR TO BE MULTIPLIED TO THE
INPUT                 RNU
       RNUC2     2ND CONVERSION FACTOR TO BE MULTIPLIED TO THE
INPUT                 RNU
       CMNUC1    1ST CONVERSION FACTOR TO BE MULTIPLIED TO THE
INPUT                 MNU
       CMNUC2    2ND CONVERSION FACTOR TO BE MULTIPLIED TO THE
INPUT                 MNU
       RERR      ALLOWABLE ERROR IN RNU DATA IN m
       EPS       CONVERGENCE CRITERION TO BE USED IN SPLINE ROUTINE
       V0        FLOW VEL. (OR VEHICLE VEL.) AT FREE STREAM, in m/s
                 AFTER BEEN MULTIPLIED BY V0C
       V0C       CONVERSION FAC. TO BE MULTI. TO V0
       PH0       STATIC PRESSURE HEAD AT FREE STREAM in m

READ19 RNU       RADII OF NCMNU PTS., in m AFTER MULTIPLYED W/
                 RNUC1*RNUC2

READ20 CMNU      MERIDIONAL VEL. AS FUNCTION OF RNU, in m/sec AFTER
                 BEEN MULTIPLIED BY CMNUC1*CMNUC2

READ21 CPTNU     PRESSURE COEFFICIENT AS FUNCTION OF RNU

***** READ22 & READ23 ARE REQUIRED ONLY IF IBFLOW=0 *****
```

READ22 G MASS FLOW RATE in kg/sec
CPTC CONSTAT PRESSURE COEFFICIENT
V0 FLOW VEL. (OR VEHICLE VEL.) AT FREE STREAM, in m/s AFTER BEEN MULTIPLIED BY V0C
V0C CONVERSION FAC. TO BE MULTI. TO V0
PH0 STATIC PRESSURE HEAD AT FREE STREAM in m

READ23 R RADIAL DISTANCE FROM AXIS OF RATATION in m

***** READ24 TO READ29 ARE REQUIRED ONLY IF IBFLOW=1 *****
READ24 G MASS FLOW RATE in kg/sec

READ25 CM MERIDIONAL VEL. in m/sec

READ26 R RADIAL DISTANCE FROM AXIS OF RATATION in m

READ27 H ENTHALPY in J/kg or (m/sec)**2

READ28 TNA TN OF EQ. 10

READ29 CTHD A KIND OF [DELTA(CM*CTH)]/CM, FROM BLADE TO BLADE PROGRAM

C.2 PROGRAM RIS

The following is the nomenclature of input data used in Program RIS:

AI	I function
AII	I function
AMDA	Lambda, in deg.
AMDAA	Starting Lambda, in deg.
AMDAI	Increment of Lambda, in deg.
AMDAK	AMDA(KA)
AMDAZ	Ending Lambda, in deg.
CO	cos(Lambda)
EPS	Converging criterion
EQUAL	Equal sign
K	A dummy counter
KA	Dummy counter, for AMDA
KAMAX	Max. allowable KA
KAZ	Final KA
KX	Dummy counter, for XS
KXMAX	Max. allowable KX
KXZ	Final KX
LENGTH	# of characters for the length of the horizontal lines in table
N	A dummy variable, 1 to NMAX
NMAX	Max. # of terms (either pos. or neg. side) in the series
NMAXMX	Max. N used
Q	A temporary variable
Q1	A temporary variable
Q2	A temporary variable
Q3	A temporary variable
Q9	A temporary variable
R	R function
RANDI	A string contains R & I
RPI	1/(PI)
RR	R function
SI	sin(Lambda)
UL	Underline sign
XS	(x0-x)/s = (x0-x)/c * c/s
XSA	Starting XS
XSI	Increment of XS
XSK	XS(KX)
XSZ	Ending XS

C.3 PROGRAM MELLOR

The following is the nomenclature of input data used in Program MELLOR:

```
READ1      TITLE (WILL NOT SHOW IN OUTPUT)

READ2      TITLE (TO BE SHOWN IN OUTPUT)

READ3
MH         = 1   TO CALCULATE THE CASE SHOWN IN MELLOR (1959);
           = 2   FOR THE CASE FOLLOWS HERRIG, et al. (1951)
           =11   TO PREPARE A TABLE OF G, H, B, & T FUNCTIONS (OF
                 STAGGER ANGLE) FOR A SPEIFIED SOLIDITY
INCAM      = 1   IF THE CAMBER IS A CIRCULAR ARC;
           = 2   IF THE BLADE IS NACA 65 W/ a=1.0;
           = 3*  IF THE CAMBER IS TO BE CALCULATED FROM A FORMULA
IFLAT      (NOT USED IF INCAM=1)
           = 0*  IF THE INPUT CAMBER IS NOT TO BE MODIFIED;
           = 1   IF THE SLOPES OF INPUT CAMBER  ARE TO BE KEPT
                 CONSTANT WITHIN 5% OF BOTH ENDS (SEE MELLOR, 1959)

NSEC       # OF SEGMENTS ALONG THE CHORD, e.g., 50;
           SET TO 10 IF MH=1 TO COMPARE W/ TABLES IN MELLOR (1959)

NFR        # OF FOURIER SERIES TERMS TO BE USED, e.g., NFR=3 TO
HAVE             A0, A1, & A2
LIST       = 0   IF CAMBER & THICKNESS DATA ARE NOT TO BE LISTED;
           = 1*  TO LIST

READ4
CBID       INPUT IDEAL CB (KNOWM) FOR THE INPUT CAMBER; IF THE
           CALCULATED, CB IS TO BE NEGLECTED; OTHERWISE, INPUT A
           NUMBER >=9 TO USE THE CALCULATED CB
SEND1      A FACTOR TO BE USED BY 4 SUB SPLIN1 IN THIS ROUTINE....
           IT IS TO BE APPLIED TO S(2) & S(N-1) TO OBTAIN S(1) &
           S(N), S BEING CURVATURE... NORMALLY 0, .5, OR 1...
           e.g., 1 FOR CAMBER & THICKNESS DISTRIBUTIONS

***** SKIP READ5 & READ6, IF MH.NE.1 *****
READ5 (IIF MH=1)
LISTO      = 1   TO MAKE DG00-DT1 & G00-T1 TABLES (MELLOR, 1959);
           = 2   TO MAKE A01-AT TABLE;
           = 3   TO MAKE CL-AT TABLE

READ6 (IIF MH=1)
ALPHAM     MEAN ANGLE OF INCIDENCE, IN deg. (NOT USED IF LISTO=1)

***** SKIP READ7 & READ8, IF MH.NE.2 *****
READ7 (IIF MH=2)
IRES       = 1   TO FIND LIFT COEF. BY CL=2*PI*(A1+A2);
           = 2   TO CALC. CL AS ABOVE PLUS BETA1 & CL1
```

```
ISCL      = 0  TO DO NOTHING
          = 1  TO STORE CALCULATED SET OF LIFT COEFFICIENTS TO BE
          COMPARED W/ HERRIG, et al., 1951

READ8 (IIF MH=2)
CC        FACTOR FOR EFFECTIVE CB, <=1, 1 FOR THEO. VALUES,
          MELLOR HAS .725
CA        FACTOR FOR EFFECTIVE ALPHA(M), <=1, 1 FOR THEORETICAL
          VALUES
CSA       STARTING C/S (SOLIDITY) FOR CASES TO BE CALCULATED,
          e.g., 0.
CSI       INCREMENT OF C/S TO BE CALCULATED, e.g., 2.0
CSZ       ENDING C/S, e.g., .5
SEND5     A FACTOR TO BE USED BY 4 SUB SPLIN1 IN SUB LOOP... IT
          IS TO BE APPLIED TO S(2) & S(N-1) TO OBTAIN S(1) &
          S(N), S BEING CURVATURE... NORMALLY 0, .5, OR 1...
          e.g., .5 FOR SOME DISTRIBUTIONS

***** SKIP READ9 IF MH.NE.11 *****
READ9 (IIF MH=11)
CSA       C/S (SOLIDITY) FOR THE GENERATED TABLE OF G, H, B, & T
          FUNCTIONS

READ10
IDUM      INTEGER 999, TO CHECK THE END OF INPUT FROM UNIT 5

***** SKIP READ701 THRU READ706, IF MH.NE.2 *****
READ701 (IIF MH=2)
NB        # OF ELEMENTS IN ARRAY BETA1
NS        # OF ELEMENTS IN ARRAY SIG
NC        # OF ELEMENTS IN ARRAY CB1
NA        # OF ELEMENTS IN ARRAY ALPHA1

READ702 (IIF MH=2)
BETA1     BETA AT INLET, IN DEG, SIZE NB

READ703 (IIF MH=2)
SIG       SOLIDITY, SIZE NS

READ704 (IIF MH=2)
CB1       Cb, SIZE NC

READ705 (IIF MH=2)
ALPHA1    INCIDENT ANGLE AT INLET, IN DEG, SIZE NA

READ706 (IIF MH=2)
ICL       INTEGER VALUES OF 10000 TIMES OF LIFT COEFFICIENT,
          EXPERIMENTAL VALUES WHEN INPUT, CALCULATED VALUES WHEN
          OUTPUT, SIZE (NA,NC,NS,NB)... THOSE NOT USED ARE
          ASSIGNED A VALUE HIGHER THAN 32600

READ901
TITL      TITLE FOR THIS R & I FUNCTION FILE
```

```
READ902
EPS         CLOSING CRITERION USED IN PROGRAM RI, NO USE HERE
AMDA        STARTING STAGGER ANGLE, IN DEG
AMDI        INTERVAL OF STAGGER ANGLE, IN DEG
AMDZ        ENDING STAGGER ANGLE, IN DEG
XSA         STARTING XS ( [X0-X]/S, SEE MELLOR, 1959)
XSI         INTERVAL OF XS
XSZ         ENDING XS

READ903
R           R FUNCTION, SEE MELLOR, 1959
AI          I FUNCTION, SEE MELLOR, 1959
JACK        A DUMMY NUMBER, SHOULD BE 999 IF DATA FILE IS CORRECT

***** SKIP READ1101 THRU READ1104 IF INCAM.NE.2 *****
READ1101 (IIF INCAM=2)
TITL        TITLE FOR THIS CAMBER DATA FILE

READ1102 (IIF INCAM=2)
NYC         # OF DATA SETS FOR CAMBER COORDINATES

READ1103 (IIF INCAM=2)
XC          X-COORDINATES FOR CAMBER YC, SIZE NYC

READ1104 (IIF INCAM=2)
YC          CAMBER AS FUNCTION OF XC, SIZE NYC

READ1105
TITL        TITLE FOR THIS BLADE THICKNESS FILE

READ1106
NYT         # OF DATA SETS FOR BLADE THICKNESS

READ1107
XT          X-COORDINATES FOR DATA OF THICKNESS YT, SIZE NYT

READ1108
YT          HALF-THICKNESS OF BLADE AT CORRESPONDING XT, SIZE BYT

READ1109
JACK        = 999 AS A CHECK
```

C.4 PROGRAM DSN3

The following is the nomenclature of the input data used in Program DSN3:

READ1-TITLE FOR THIS RUN

READ2-DATA FOR THE SYSTEM

DEN	FLUID DENSITY; ENTER A NUMBER, IN PROPER UNIT, HERE IF IT IS CONSTANT OTHERWISE ENTER 0 AND ENTER THE DATA IN SUB "INP2"
RPM	ROTATIONAL SPEED, IN rpm
RSTAR	REFERENCE RADIUS, IN USER'S UNIT
CONVR	CONVERSION FACTOR TO BE MULTIPLY TO RSTAR SUCH THAT RSTAR IS IN METERS OR FEET

READ3-CONVERGENCE CRITERIA

CONFB	CONVERGENCE FACTOR FOR ADJUSTING BETA0, e.g., .5
CONFC	CONVERGENCE FACTOR FOR ADJUSTING CB, e.g., 1.
CONFSA	CONVERGENCE FACTOR FOR ADJUSTING STAGGER ANGLE, STAG, e.g., 0.
CONFSO	CONVERGENCE FACTOR FOR ADJUSTING SOLIDITY, SIG, e.g., 1.
EPSA	CONVERGENCE CRITERIA FOR BOTH del EQ. PHI AND del BETA1, e.g., .001
EPSS	CONVERGENCE CRITERIA FOR SOLIDITY ITERATION, e.g., .001
EPSSO	ACCEPTABLE CRITERIA FOR INPUT SOLIDITY, e.g., .01
MMAX	MAXIMUM ITERATION NUMBER FOR A LOOP (del EQ. PHI & del BETA1) IN SUB "PFM3"
IDBUG	> 0 TO HAVE EXTRA OUTPUT FOR DEBUGGING PURPOSE; = 0 FOR NORMAL OUTPUT

READ4-SOME NUMBERS FOR ROTOR & ITS BLADES

NBLADE	# OF BLADES
NSECR	# OF CROSS-SECTIONS ALONG THE RADIUS DIRECTION (USE NSECR=1 AT THIS MOMENT BECAUSE SOLIDITY VARIES FROM SECTION TO SECTION)
NSTA	# OF STATIONS ALONG A CHORD, INCLUDING LEADING & TRAILING EDGES
INTERP	= 0 TO READ THE SECTION DATA DIRECTLY FROM INPUT; <> 0 TO OBTAIN THE SECTION DATA THROUGH INTERPOLATION BASED ON DATA READ FROM UNIT 21 (IN SUB INP2)

READ2XX.... IN SUB "INP2"

READ11XX... IN SUB "MELLOR2"

...AFTER READ7 IN "INP"...

READ201 TITLE

```
READ202        SIG      INITIALLY GUESSED SOLIDITY

***** READ203 TO READ210 REQUIRED iff INTERP=0 *****
***** READ210 REQUIRED iff DEN=0 *****
READ203        CT1      PERIPHERAL VEL. AT THE LEADING EDGE (L/T)
               CT2      PERIPHERAL VEL. AT THE TRAILING EDGE (L/T)

READ204        XAX      X COORDINATES ALONG AXIX OF ROTATION, FOR
                        EACH OF NSTA STATIONS ALONG THE CHORD (L).
                        XAX WILL BE NORMALIZED TO CHORD LENGTH, IN
                        SUB "MELLOR2", SUCH THAT XAX(1)=0 &
                        XAX(NSTA)=1 (.)
READ205        R        RADIUS FROM AXIS OF ROTATION (L)
READ206        XM       DISTANCE ALONG STREAMLINE (M-COORDINATE)
                        (L)
READ207        CM       MERIDIONAL VEL. (L/T)
READ208        B        DISTANCE BETWEEN ADJACENT STREMLINES (L)
                        (EITHER BETWEEN I-1 & I+1 OR HALF OF THAT)
READ209        BK       BLOCKAGE FACTOR OWING TO BLADE THICKNESS
                        INSIDE THE IMPELLER (.)
***** READ210 REQUIRED iff DEN=0 *****
READ210        RHO      DENSITY OF FLUID (M/L^3)

***** READ211 REQUIRED iff INTERP<>0 *****
READ211        RSL      RADIUS DISTANCE OF THE STREAMLINE FROM THE
                        AXIS OF ROTATION, AT SECTION IRSL (SEE
                        BELOW) (L)
                        (DATA INPUT FROM FILE #21 ARE USED TO
                        ESTABLISH THE PERAMETERS BASED ON THIS
                        RADIUS & ITS NEARBY INPUT RADII BY SIMPLE
                        INTERPOLATION)
               IRSL     STATION # (FROM 1 TO NSTA) WHERE RSL IS
                        DEFINED

***** READ2101 TO READ210 REQUIRED iff INTERP<>0 *****
***** READ2111 REQUIRED iff DEN=0 *****
READ2101       (REM)    TITLE LINE

READ2102       NI       # OF STATIONS (q-LINES) FROM LEADING
EDGE                         TO TRAILING EDGE
               NJ       # OF STREAMLINES

READ2103       CT1Z     PERIPHERAL VEL. AT THE LEADING EDGE (L/T)

READ2104       CT2Z     PERIPHERAL VEL. AT THE TRAILING EDGE (L/T)

READ2105       XAXZ     X COORDINATES ALONG AXIX OF ROTATION,
FOR                     EACH OF NSTA STATIONS ALONG THE CHORD (L).
                        XAX WILL BE NORMALIZED TO CHORD LENGTH, IN
                        SUB "MELLOR2", SUCH THAT XAX(1)=0
&                       XAX(NSTA)=1

READ2106       RZ       RADIUS FROM AXIS OF ROTATION (L)
```

READ2107	XMZ	DISTANCE ALONG STREAMLINE (M-COORDINATE) (L)
READ2108	CMZ	MERIDIONAL VEL. (L/T)
READ2109	BZ	DISTANCE BETWEEN ADJACENT STREMLINES (L) (EITHER BETWEEN I-1 & I+1 OR HALF OF THAT)
READ2110 THICKNESS	BKZ	BLOCKAGE FACTOR OWING TO BLADE INSIDE THE IMPELLER (.)

***** READ2111 REQUIRED iff DEN=0 *****

READ2111	RHOZ	DENSITY OF FLUID (M/L^3)