NASA Contractor Report 3831

Development of a Turbomachinery Design Optimization Procedure Using a Multiple-Parameter Nonlinear Perturbation Method

Stephen S. Stahara

CONTRACT NAS3-20836
SEPTEMBER 1984

NASA

NASA Contractor Report 3831

Development of a Turbomachinery Design Optimization Procedure Using a Multiple-Parameter Nonlinear Perturbation Method

Stephen S. Stahara
Nielsen Engineering & Research, Inc.
Mountain View, California

Prepared for
Lewis Research Center
under Contract NAS3-20836

LIMITED DISTRIBUTION

This document will remain under distribution limitation until September 1985.

NASA

National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1984

TABLE OF CONTENTS

Table of Contents (Cont.)

SUMMARY

An investigation was carried out to complete the preliminary development of a combined perturbation/optimization procedure and associated computational code for designing optimized blade-to-blade profiles of turbomachinery blades. The overall purpose of the procedures developed in this study is to provide demonstration of a rapid nonlinear perturbation method for minimizing the computational requirements associated with parametric design studies of turbomachinery flows. The method reported here combines the multiple-parameter nonlinear perturbation method, successfully developed in previous phases of this study, with the NASA TSONIC blade-to-blade turbomachinery flow solver, and the COPES-CONMIN optimization procedure into a user's code for designing optimized blade-to-blade surface profiles of turbomachinery blades. Results of several design applications and a documented version of the code together with a user's manual are provided.

1. INTRODUCTION

The remarkable success of advanced computational methods for determining complex fluid dynamic phenomena has created a continuing demand both for increasing the accuracy and generality of these methods, as well as the desire to incorporate these methods into a routine use mode needed for design and parametric investigations. However, a major impediment to such routine use of many of these current and emerging computational codes is the high computational cost required in their direct application to situations requiring repetitive high-frequency use. Consequently, a real need exists for determining the means to reduce these costs while retaining the required accuracy in such nonlinear applications. While this need exists in virtually all engineering applications when relatively sophisticated numerical codes are employed, for turbomachinery applications it is particularly severe since both the underlying aerodynamic computation of the flow field is costly and also the number of flow and geometry parameters needed to be varied in design studies is large.

The ultimate objective of this investigation is to develop and demonstrate methods that would provide the means to reduce substantially the overall computational requirements necessary for turbomachinery design studies. It is conceived that these methods would be coupled with high run-time general turbomachinery computational flow field solvers and would be used in conjunction with them in applications where large numbers of related nonlinear turbomachinery solutions are required.

That such methods can be realized has been successfully demonstrated in the previous phases (Refs. 1-3) of this study. In the first of those investigations (Ref. 1), several candidate methods were studied and the most promising method was identified. Extensive development and testing of that method was then carried out in the subsequent phase (Ref. 2). This testing was performed for a wide variety of both flow and geometry parameters for turbomachinery flows past isolated blades and compressor cascades at both subcritical and supercritical conditions. Emphasis was placed in particular on strongly supercritical flows which exhibited large surface shock movements over the parametric range studied. Comparisons of the perturbation predictions with the corresponding exact nonlinear solutions indicated a remarkable accuracy and range of validity of the perturbation method. In the most recently completed phase (Ref. 3), the perturbation method was extended to treat simultaneous multiple-parameter perturbations. Extensive testing of the method has demonstrated remarkable accuracy and range of validity of the multiple-parameter perturbation procedure in direct correspondence with the previous results obtained for single-parameter perturbations. Additionally, initial applications of the multiple-parameter perturbation method combined with an optimization procedure were made (Ref. 3) to several turbomachinery blade design problems. The results demonstrated the potential of the perturbation method for reducing the computational work in such applications by an order of magnitude with no degradation in accuracy.

The work reported here describes the continued development of the combined multiple-parameter perturbation method/ optimization procedure. The primary objective of this phase is on the development of that combined procedure into an operational method for designing optimized blade-to-blade profiles of turbomachinery compressor blades.

2. ANALYSIS

2.1 Perturbation Concept

The obvious method of carrying out a perturbation analysis, that is by establishing and solving a series of linear perturbation equations in the manner of Van Dyke (Ref. 4), appears to be an obvious choice for the current application. The initial phase of this study (Ref. 1) established that for sensitive flows, such as occur in turbomachinery, the basic linear variation assumption fundamental to the technique is sufficiently restrictive that the allowable range of parameter variation is so small as to be of little practical use. A novel alternative to the linear perturbation equation approach was then subsequently developed and successfully tested (Refs. 1-3) in which a correction method is used that employs two or more nonlinear solutions obtained from the basic nonlinear flow solver, rather than just one as in the linear perturbation equation approach. For this alternative method, the basic perturbation solution is determined simply by differencing two nonlinear flow solutions removed from one another by some nominal change of a particular flow or geometrical quantity. A unit perturbation solution is then obtained by dividing that result by the change in the perturbed quantity. Related solutions are determined by multiplying the unit perturbation by the desired parameter change and adding that result to the base flow solution. This simple procedure, however, only works directly for continuous flows for which the perturbation change does not alter the solution domain. For those perturbations which change the flow domain, coordinate stretching is necessary to ensure proper definition of the unit perturbation solution. For discontinuous flows, special coordinate straining is necessary to account for movement of discontinuities due to the perturbation.

At this point, the perturbation concept based on these ideas has both been implemented and thoroughly tested in a wide range of applications (Refs. 1-3, 5-11). These applications have purposely involved a number of different nonlinear flow field solvers to provide the base solutions necessary for the perturbation calculation. The most extensive and systematic of these studies are reported in References 2, 3, 9, and 11, where results are provided for case studies involving a variety of different flow and geometry parameter perturbations of nonlinear subsonic and transonic flows past isolated blade and compressor cascade geometries. For those applications, emphasis was placed in particular on strongly supercritical transonic flows which exhibit large surface shock movements over the parameter range studied. By extensive comparisons with the exact nonlinear solutions, these studies have established the accuracy, range of validity, and versatility of the perturbation method.

The underlying reason of the remarkable accuracy of the perturbation method developed in this study lies in the use of coordinate straining to define the unit perturbation. As shown in Figure 1, where the perturbation between two nonlinear solution states is displayed graphically as the shaded area between the base and the strained and unstrained calibration solution, coordinate straining provides the ability to account accurately for the displacement of a multiple number of discontinuities and maxima of high-gradient regions due to a parameter change. This enables the perturbation method to maintain very high accuracy in regions of high gradients where most perturbation methods commonly fail, and to maintain that accuracy over large parametric ranges.

In what follows, we provide a brief account of the theoretical essentials of the strained-coordinate perturbation concept as it configured and implemented in the present design application. This is to predict simultaneous multiple-parameter perturbation flow solutions for blade surface properties of turbomachinery blades for use in optimized blade design. The turbomachinery flow solutions thus considered can contain a total number N of discontinuities or high-gradient continuous regions. Complete details of the mathematical basis of the method, in particular, the application to flow field properties, may be found in Reference 3.

For the prediction of distributions of surface properties involving simultaneous multiple-parameter perturbations of aerodynamic flows where flow properties are required along some contour, the strained-coordinate, first-order, multiple-parameter perturbation approximation can be represented by

$$Q(x;\varepsilon) = Q_o(s) + \sum_{j=1}^{M} \varepsilon_j Q_{1_j}(s) + \ldots \qquad (1)$$

$$x = s + \sum_{j=1}^{M} \varepsilon_j x_1(s) + \ldots \qquad (2)$$

where x is the independent variable measuring distance along the surface contour or some convenient projection of that distance, s is the strained coordinate, and ε_j a small parameter representing the change in one of M flow or geometrical variables which we wish to vary simultaneously.

In order to determine the first-order corrections $Q_{1_j}(s)$, we require one base and M calibration solutions in which the calibration solutions are determined by varying each of the M arbitrary independent parameters q_j by some nominal amount from the base flow value while keeping the others fixed at their baseline values.

In this way, the first-order corrections $Q_{1_j}(s)$ can be determined as

$$Q_{1_j}(s) = \frac{Q_{c_j}(\bar{x}_j) - Q_o(s)}{\bar{\varepsilon}_j} \tag{3}$$

where Q_{c_j} is the calibration solution corresponding to changing the j*th* parameter to a new value q_{c_j}, $\bar{x}_j$ is the strained coordinate pertaining to the Q_{c_j} calibration solution, and $\bar{\varepsilon}_j = q_{c_j} - q_{o_j}$ represents the change in the q_j parameter from its base flow value. If we now desire to keep invariant during the perturbation process a total of N points corresponding to discontinuities or high-gradient maxima, we can represent the first-order solution by

$$Q(x,\varepsilon_j) = Q_o(s) + \sum_{j=1}^{M} \varepsilon_j Q_{1_j}(s) \tag{4}$$

where $Q_{1_j}(s)$ is given above and

$$\bar{x}_j = s + \sum_{i=1}^{N} \bar{\varepsilon}_j \delta x_i x_{1_i}(s) \tag{5}$$

$$x = s + \sum_{i=1}^{N} \varepsilon_j \delta x_i x_{1_i}(s) \tag{6}$$

$$\bar{\varepsilon}_j = q_{c_j} - q_{o_j} \tag{7}$$

$$\varepsilon_j = q_j - q_{o_j} \tag{8}$$

$$\bar{\varepsilon}_j \delta x_i = \left(x_i^c - x_i^o\right)_j \tag{9}$$

$$\epsilon_j \delta x_i = \frac{\epsilon_j}{\bar{\epsilon}_j}\left(x_i^c - x_i^o\right)_j \tag{10}$$

Here $\bar{\epsilon}_j \delta x_i$ given in Equations (5) and (9) represent the displacement of the *ith* invariant point in the *jth* calibration solution from its base flow location due to the selected change $\bar{\epsilon}_j$ in the q_j parameter given by Equation (7), $\epsilon_j \delta x_i$ given in Equations (6) and (10) represents the predicted displacement of the *ith* invariant point from its base flow location due to the desired change ϵ_j in the q_j parameter given by Equation (8), and $x_{1_i}(s)$ is a unit-order straining function having the property that

$$x_{1_i}\left(x_k^o\right) = \begin{cases} 1 & k = i \\ 0 & k \neq i \end{cases} \tag{11}$$

which assures alignment of the *ith* invariant point between the base and calibration solutions.

In References 2 and 3, detailed studies were made of the effect of different straining functions on the accuracy of the perturbation result. In Reference 3, identification was made of a superior class of straining functions for use in general nonlinear applications. This class turns out to be comprised of linear piecewise-continuous straining functions, and particular members of this class have proven effective in all case studies undertaken to date. It has been found that this type of straining function is able to maintain high accuracy of the perturbation predictive result in the vicinity of the invariant points, and furthermore, this class of straining function introduces no excessive straining in regions removed from those locations. Occurrence of the latter phenomenon has been found to be a common failure of certain other classes of straining functions (Refs. 2, 3).

The functional forms of the straining for linear piecewise-continuous straining functions can be compactly written. For example, the strained coordinate $\bar{x}_j$, in Equation (5) is given by

$$\bar{x}_j = s + \left\{\frac{x_{i+1}^o - s}{x_{i+1}^o - x_i^o} \cdot \left(x_i^c - x_i^o\right)_j + \frac{s - x_i^o}{x_{i+1}^o - x_i^o} \cdot \left(x_{i+1}^c - x_{i+1}^o\right)_j\right\} H\left(x_{i+1} - s\right) \cdot H\left(s - x_i^o\right) \tag{12}$$

where H denotes the Heaviside step function. In addition to the points corresponding to discontinuities or high-gradient maxima, it is usually also necessary in coordinate straining to hold invariant both of the end points along the contour. Consequently, for the application developed here, the array of invariant points in the base and calibration solutions are taken as

$$\begin{aligned} x_i^o &= \left\{0,\ x_1^o,\ x_2^o,\ \ldots,\ x_n^o,\ 1\right\} \\ x_{i_j}^c &= \left\{0,\ x_{1_j}^c,\ x_{2_j}^c,\ \ldots,\ x_{n_j}^c,\ 1\right\} \end{aligned} \tag{13}$$

where the contour length has been normalized to unity and where n is the number of invariant points along the blade contour exclusive of the end points.

2.2 Combination of Perturbation Method with Optimization Procedures for Blade Design

One of the major objectives of the previous phase of this investigation (Ref. 3) was the demonstration of the capability of the perturbation method to work effectively in an important nonlinear design environment related to turbomachinery. The particular application selected was optimized turbomachinery blade design. Toward the above objective, the perturbation method, configured to treat simultaneous multiple-parameter changes, was combined with proven optimization procedures (Refs. 12, 13). Next, performance and design constraints characteristic of certain turbomachinery blade design problems were constructed. Finally, applications of the combined procedure to several case studies involving blade profile optimization were made. The objectives of these initial applications were to demonstrate the workability of the perturbation concept in a design environment, provide a benchmark of the potential for computational savings of the combined perturbation/optimization procedure for some typical design problems, and determine the accuracy of the perturbation-predicted results for these cases.

Two different types of optimization problems were considered. The first set of case studies involved isolated blades and were more fundamental in nature, while the second set involved a practical turbomachinery compressor blade design. For the first set of studies, the particular isolated blade design optimization problems selected for study involved the alteration of a baseline profile shape by adding to the baseline profile a set of shape functions according to the relation

$$Z(x) = Z_o(x) + \sum_{i=1}^{M} A_i F_i(x) \qquad (14)$$

where Z_o are the ordinates of the baseline profiles, F_i are the shape functions, and the coefficients A_i are the design variables whose values are determined by the optimization process as a result of a search through design-variable solution space to achieve a desired design improvement. The general class of geometric shape functions employed, which have been found to be successful in previous applications involving optimization of supercritical airfoil sections (Ref. 14), consisted of exponential decay functions and sine functions. These are of the general form $(1 - x) \cdot x^p/e^{qx}$ and $\sin (\pi x^r)^n$, where the exponents p, q, r, and n are selected to provide a desired ordinate maximum at a particular chordwise location. The exponential functions are generally employed to provide adjustments near the leading edge, while the sine functions are used to provide maximum ordinate changes at particular chordwise stations. Illustrations of the chordwise variation of typical members of these classes of shape functions are provided in Figure 2, and it can be seen that these functions smoothly concentrate ordinate thickness at selected locations. Consequently, they can be used effectively to add a series of smoothly blended bumps or scallops at selected locations along a baseline blade profile in order to control locally the flow characteristics at particular sections on the blade.

A strategy that has proven convenient for performing optimization studies involving aerodynamic performance parameters (Ref. 14) has been to recontour the profile shape so as to tailor the surface pressure distribution to conform to a desired distribution. This type of objective provides local control over the basic aerodynamic surface flow property of importance, and provides a means of attempting to achieve aft pressure gradients sufficiently weak to avoid separation. An important corollary advantage of using such an objective is that viscous separation can be minimized. This allows use of an inviscid aerodynamic flow solver in the optimization process rather than a much more computationally-expensive viscous solver, and assures that the optimization result thus obtained at the inviscid level is representative of the actual flow.

In such studies, the characteristics that are primarily sought after in the optimization process are the minimization of both the peaky behavior near the leading edge and the compressive gradient on the aft portion of the suction surface that typically exists on the baseline profiles considered. This is illustrated schematically in Figure 3. For two of the three series of case studies undertaken for isolated blades, the objective function was taken as the minimization of the mean squared error between the predicted and desired surface pressure distribution, i.e.,

$$\mathrm{OBJ} = \sum_{k=1}^{K} \left[C_{P_{\mathrm{predicted}}}(x_k) - C_{P_{\mathrm{desired}}}(x_k) \right]^2 \qquad (15)$$

where K represents the number of chordwise locations x_k where desired and calculated surface pressures are compared. For the third series, the objective function was chosen to be the drag coefficient squared, a much more sensitive quantity. The optimization procedure employed was the now-standard CONMIN code (Ref. 12).

The detailed results presented in Reference 3 for the three case studies on isolated blades clearly established the ability of the perturbation method to work accurately in a highly nonlinear multiple-parameter design environment. This was found to be true for both subcritical as well as strongly supercritical flow situations. The supercritical case study employing drag coefficient as the objective function demonstrated the advantages and accuracy benefits of multiple invariant point clustering in high-gradient regions as well as an explicit straining concept for determining the perturbation result. Finally, the potential for computational savings with the perturbation method in such optimization problems was benchmarked at an order of magnitude, and the possibility was demonstrated of even obtaining in some cases improved results in terms of a more global minima of the objective function with employing the perturbation method as compared to not using it.

The final set of case studies involved the optimization of realistic compressor blades and was directed toward laying the foundations of a practical turbomachinery blade design/optimization procedure coupled with the simultaneous multiple-parameter perturbation method. The combined code consisted of the TSONIC blade-to-blade flow solver (Ref. 15) with generalized circular-arc blade geometry routines BLADE (Ref. 16) to describe the blade profiles, and the more generalized COPES-CONMIN optimization procedure (Ref. 13). The combined PERTURB/TSONIC/BLADE/COPES-CONMIN procedure, called BLDOPT, was tested on several NASA designed case studies to demonstrate the accuracy and capability of the combined procedure on problems typical of practical turbomachinery blade design. These case studies involved, as design variables, selected geometry parameters related to the NASA/Lewis circular-arc blade profiles. The optimization objective usually chosen was the minimization of the peak suction surface velocity diffusion. Although that choice of objective is a somewhat sensitive selection since it represents a point quantity in a high-gradient region, the combined procedure was able to demonstrate good results, with computational work savings comparable to those found in the previous case studies (Ref. 3).

In these case studies, however, because the design variables were selected as basic geometry parameters (blade curvature, maximum blade thickness, etc.) related to the circular-arc blade geometry, the optimization search problem itself becomes more sensitive. This occurs because these design variables by their very nature effect more global changes in the aerodynamic solution and therefore tend to interact more strongly with each other than would a corresponding optimization problem which employs, say, local shape functions as the design variables. For the latter case, the design variables generally effect only local changes in the aerodynamic solution and therefore tend to interact much more weakly with one another. Consequently, optimization problems posed with such design variables are usually much more stable and less sensitive to small changes in search direction.

However, the ability to employ basic blade geometry parameters as design variables is very attractive as it relates directly to the capability of performing the more general preliminary blade design problem where a wide universe of basic blade shapes is considered. This contrasts to the problem involving use of local shape functions as design variables which relates to a more specific refined-design problem where the basic blade profile has already been selected. The ability to treat both problems is important, with the former being the more general and more difficult to do.

3. RESULTS

Because the ultimate utility of the perturbation methods being developed under this investigation is in optimized turbomachinery design, the primary objective of the current study was to complete the development of the combined PERTURB/TSONIC/BLADE/COPES-CONMIN procedure (BLDOPT) for performing optimized turbomachinery blade-to-blade surface profile design, and to finalize the procedure into a user's code so as to make generally available such a procedure to facilitate future use and testing by the general turbomachinery community. Toward that end, we have completed the assembly and preliminary verification testing of the four component codes configured into a combined program and controlled under a user-friendly executive program.

A number of features have been incorporated into the current version of the BLDOPT program reported here, which were not available in the preliminary version reported in Reference 3. These features considerably enhance both the capability and generality of the method. An explicit straining procedure, which in essence specifies the points at which the final solution results are determined rather than allow these points to be determined implicitly from the straining of the base flow points as was done standardly in the past, has been implemented in the present BLDOPT code. The explicit procedure avoids a double interpolation of the perturbation result and has been found to yield significantly improved accuracy in high-gradient regions at only a very slight increase in computational work. In addition to the explicit straining procedure, the updated BLDOPT code has incorporated several new options available to the user when employing the perturbation method. These options, which are controlled by the parameter IOPT defined in Section A.4 of the user's manual, relate to the way the calibration solution matrix is defined. One of these options provides the user with a basically automatic hands-off procedure for using the perturbation method. Under this option, the user is not required to preselect and input the design variable values for the calibration solution matrix. Rather the matrix is determined completely by the program in the following way. For the first optimization cycle, the perturbation method is not used. Full nonlinear aerodynamic solutions are determined by the flow field code as required as input for the gradient and search optimization calculations. After the first search cycle is complete and a new design point determined, design variable values for the calibration solution matrix are then determined based on the direction that the first search cycle has taken. This results in an extremely good definition of the calibration solution matrix. The result is that the design variable solution space which is subsequently searched on the second and successive optimization searches usually requires only very reasonable interpolations/extrapolations within the design variable parameter range of the defined solution matrix.

This option (IOPT = 3) provides the automatic user-invisible procedure for defining the calibration solution matrix. An additional option (IOPT = 2) which requires user-input for defining the calibration solution matrix has also been incorporated into the code. Within this option, the calibration matrix definition can be accomplished by either individually specifying all the design variables (ICALB = 1) or alternatively (ICALB = 0) by employing a constant-value calibration stepsize which increments each design variable by this fractional change of its base flow value. The automatic IOPT = 3 option requires the additional cost of one optimization search cycle using full aerodynamic flow field solutions over that of the IOPT = 2 option which in contrast requires a user-input of the design variable values for the calibration solutions. Nevertheless, the IOPT = 3 option provides a highly accurate and basically hands-off means of employing the perturbation method and is recommended for use when no information is available on search direction from previous related calculations.

In terms of final design variable accuracy and potential computational time savings using this IOPT = 3 option, in Figures 4 and 5 we present comparisons of a severe test of this option for the optimization results for a 5 design variable supercritical pressure tailoring case study on an isolated blade using blade contour shape functions similar to those reported in Reference 3. Figure 4 provides a comparison of the perturbation-predicted final design variables and objective function (●) with results obtained with not using the perturbation method but employing full nonlinear full potential aerodynamic solutions throughout (⊙). These are the results after 5 optimization cycles. We note the essentially exact correspondence between the final design variable values. Corresponding comparison of the objective function, in fact, indicate a slightly better result obtained by the perturbation method (■). In Figure 5, the corresponding comparison of computational work in CPU seconds and objection function reduction per otimization search cycle is provided. Here we see that after the first cycle is complete, the perturbation procedure actually requires less time to define the calibration matrix solution and complete the second search than does the full nonlinear flow field method with the same reduction in objective. From that point on, the perturbation method requires essentially no time to complete searches 3 to 5, and then an additional increment to calculate the final design result using the nonlinear flow field solver. Time savings achieved with the perturbation method for this case is 58% of that required for the full nonlinear result. These and similar results obtained with the IOPT = 3 option confirm the utility of the automatic user option for defining the calibration solution matrix.

With regard to the particular optimization problem toward which the BLDOPT program has been configured, the following eight blade geometry parameters that are commonly used to characterize NASA circular-arc blade section profiles (Ref. 16) have been incorporated as design variables:

Blade Geometry Parameter	Program Name
Blade camber angle at inlet	KICR
Blade camber angle at outlet	KOCR
Transition location/chord	T
Maximum thickness location/chord	ZM
Inlet/outlet turning rate ratio	P
Blade maximum thickness/chord	TMX
Leading edge radius/chord	THLE
Trailing edge radius/chord	THTE

These geometry parameters are illustrated graphically in Figure 6. For more details about the geometry of these classes of blade shapes, we refer the reader to Reference 16. In the BLDOPT program, any arbitrary combination up to six of the above parameters may be used in the optimization analysis.

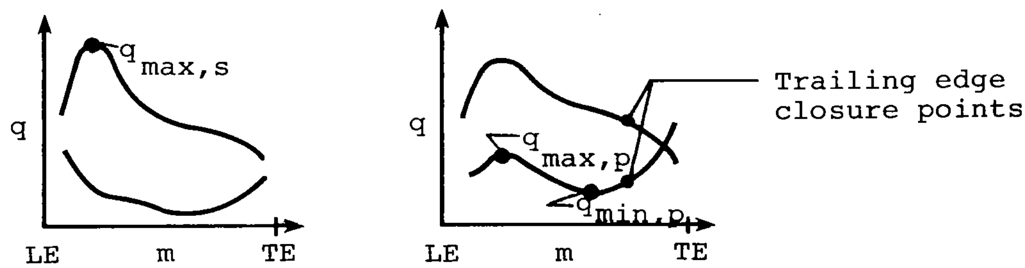
The sample optimization problem that has been examined to verify the combined procedure employs, as a design objective, the minimization of the velocity diffusion on the blade suction surface, i.e.,

$$OBJ = \frac{q_{max,suction}}{q_{avg,exit}} \tag{16}$$

where $q_{max,suction}$ is the maximum surface velocity on the blade suction surface and $q_{avg,exit}$ is the average exit velocity in the freestream. Six of the eight design variables described above are employed: blade outlet camber angle, KOCR; transition location between fore and aft circular arc sections, T; maximum thickness location, ZM; inlet to outlet turning rate ratio, P; maximum thickness, TMX; and radius of the leading edge circle, THLE. During the optimization process, each of the design variables was constrained to remain within certain prescribed bounds in order to prevent a physically-unrealistic blade design from occurring. Furthermore, several active side constraints were additionally imposed both to insure design of a physically-realistic blade and also to achieve certain desirable flow characteristics on the blade. The active side constraints employed were:

1. Maintenance of nonzero local blade thickness
2. Maintenance of low velocity diffusion on the blade pressure surface
3. Trailing edge closure via an effective Kutta condition

The basis of the first constraint is self-evident and is necessary since various combinations of the basic circular-arc blade geometry parameters can easily result in the upper and lower blade surface arcs crossing and thus lead to negative thickness. To understand the basis of the second and third constraints, it is helpful to examine the typical blade surface velocity plot as determined by the TSONIC solver. As sketched below in the plot on the left,



Baseline profile During optimization

the baseline profile has a large peak suction velocity which is desired to be reduced. It is also desirable to maintain on the pressure surface as uniform a velocity distribution as possible and, furthermore, to avoid a large mismatch in the upper and lower surface velocities at the trailing edge. During the optimization process, it is sometimes found that the lower surface velocity can develop peaks, such as shown in the plot on the right, and that the velocity distributions may cross ahead of the trailing edge. The constraints constructed to alleviate these occurrences are as follows. In order to keep large differences between the pressure surface velocity maxima and minima from occurring, we enforce the condition

$$0.0 < \frac{q_{max,press}}{q_{min,press}} < 1.3$$

where $q_{max,pres}$ is the maximum blade pressure surface velocity over the front half of the blade and $q_{min,press}$ is the minimum blade pressure surface velocity over the last two-thirds of the blade. Restricting the maximum velocity considered to the front half of the blade and the minimum velocity to the rear of the blade prevents the maximum from moving rearward and the minimum from moving forward and thereby defeating the constraint. To maintain an effective Kutta condition at the rear of the blade, we enforce the condition

$$-1 < \frac{q_{ITE-2,suction} - q_{ITE-2,press}}{25} < 1$$

Here, $q_{ITE-2,suction}$ and $q_{ITE-2,press}$ are the third last surface velocities on the flow field grid near the trailing edge on the suction and pressure surfaces, respectively. Additional constraints, or constraints different from the above, can easily be implemented into the optimization analysis with the BLDOPT code. Details for carrying this out are provided in the user's manual.

We have successfully completed a verification series of calculations of the new combined PERTURB/TSONIC/BLADE/COPES-CONMIN procedure in which the accuracy and sensitivity of the perturbation method was tested as a function of choice of the initial calibration solution matrix. The initial or base values of the design variables for the baseline blade profile, and the upper and lower bounds of the design variables that were specified for this test problem were:

Design Variable Number	Description	Lower Bound	Upper Bound	Initial Value
1	Outlet blade camber angle - KOCR	-15.0°	0.0°	-10.0°
2	Transition location/chord - T	0.20	0.60	0.25
4	Maximum thickness location/ chord - ZM	0.20	0.55	0.45
5	Inlet/outlet turning/chord - P	0.50	4.00	1.50
6	Maximum thickness/chord - TMX	0.03	0.10	0.05
7	Leading edge radius/chord - THLE	0.003	0.012	0.005

The results of these calculations are summarized in Table 1. There we have provided comparisons of the final design variables and objective function predicted when employing full nonlinear TSONIC solutions throughout the optimization process with corresponding results when using the perturbation method. For the perturbation results, different choices of the calibration solution matrix were made and are noted in the table. All the results represent converged solutions, with each calculation employing 10 optimization search cycles or less if no change in objective function should occur in three successive itermations.

The result indicated for the case when the perturbation method is not employed and TSONIC solutions are used throughout (IOPT = 1) provide the benchmark solution for comparison with the perturbation results. We note that similar full nonlinear benchmark results reported in Reference 3 for a related problem demonstrated the sensitivity of this class of optimization problems to the choice of the maximum velocity diffusion as an objective function. Identical benchmark results were obtained on the Ames Research Center CDC 7600 and the Lewis Research Center IBM 3033. The differences between those two results, which were of the same order as the differences between the various perturbation results, were due solely to the number of significant figures maintained in the respective calculations, i.e., eight for the IBM 3033 and 14 for the CDC 7600. This illustrates a common characteristic of many nonlinear multiple-parameter optimization problems, i.e., the existence of many local minimums. Furthermore, it also emphasizes the sensitivity of certain classes of optimization problems to both choice of objective function and design variables.

We observe from the perturbation results indicated in Table 1 that, with the exception of only one design variable (T) in certain instances of a deliberately made poor choice of calibration solution matrix, the final design variables predicted by the perturbation method for both IOPT=2 or 3 options trend in the same direction from the baseline value as the full nonlinear (IOPT=1) result and consistently improve the objective function. Under the IOPT=2 option, case 1 displays the results for a choice of calibration solution matrix which is very close to the final design result reached using the full nonlinear IOPT=1 option. The final perturbation-predicted design result for case 1 is slightly removed from the nonlinear result, but the objective function is quite close to the nonlinear result, indicating the presence of nearby alternative local optimization minimums. For case 2, the calibration matrix was determined by using the option ICALB=0 and selecting a constant-value calibration stepsize for each design variable of PSTEP=0.10. This implies that the value for each design variable for the calibration solution matrix is found by incrementing by 10 percent its base value. This manner of selecting the calibration matrix design variables is relatively crude since, on average, half of the perturbation flow solutions for the optimization searches will involve design variable interpolation and half will require extrapolation. Furthermore, a certain fixed percentage increment on some design variables may be far too much in range in that large solution interpolations may be required, while that same percentage increment may be far too small for other design variables which would then require large extrapolations. Nevertheless, in the face of no a priori information regarding the direction and range that the optimization search will proceed over, the use of the ICALB=0 option provides a convenient and inexpensive way of obtaining a preliminary optimization result. Additionally, as the results for case 2 indicate, the ICALB=0 results are often quite good. Analogous results using ICALB=0 and PSTEP=-0.10, which implies a decrement of 10 percent from the base design variable values for the calibration design variables, are given in case 3. As can be seen, these results are inferior to those of case 2, and illustrate the relative importance of chosing calibration matrix design variable values that result in modest interpolations/extrapolations, since for case 3 larger interpolations/extrapolations are required. The results of both cases 4 and 5 further illustrate this point. For case 4, the calibration design variables were chosen so as to result exclusively in modest perturbation solution extrapolations during the optimization searches. In case 5, similar choices were made so as to result exclusively in modest solution interpolations. Both results in terms of final design variable values and objective function are quite good. The final IOPT=2 result shown in case 6 illustrates the effect of an intentional bad selection for the calibration solution design variables in that large perturbation solution extrapolations/ interpolations are required during optimization. Although the final design result for the objective function is the least satisfactory of the six IOPT=2 cases, the majority of the design variables have trended in the appropriate direction. Consequently, even in for this situation involving deliberate poor choices of the calibration matrix, the perturbation method does not break down and yield spurious results, but provides instead a reasonable preliminary result. The final perturbation result shown in Table 1 is for the IOPT=3 option. For a slightly higher computational cost, that result provides a very good comparison between the final IOPT 1 nonlinear result.

The computational time needed to obtain the perturbation results in cases 1 to 6 under the IOPT=2 option were 76-78 secs. of CDC 7600 CPU time per case. The corresponding time for the IOPT=3 option was 97 secs. The benchmark IOPT=1 full nonlinear CDC 7600 result shown in Table 1 required 644 secs. Thus, the perturbation method provides a savings of (644 - 78)/ 644 = 88% of the computational time for the IOPT=2 option and (644 - 97)/644 = 85% for the IOPT=3 option for this example.

The significant conclusions to be drawn from this study are that the perturbation method can work accurately in multiple-parameter design environments even for very sensitive optimization problems and provide both meaningful final design results and large computational savings over not using the method. The choice of objective function such as was made for this case study, namely a point quantity located in a high-gradient region, requires careful user attention to the initial calibration matrix choice if IOPT=2 is employed in order to avoid large extrapolations, whereas if the IOPT=3 option is employed and the program allowed to automatically determine the calibration matrix, no exceptional difficulties are observed.

Additional optimization computations were made to try to determine whether an alternative choice of objective function would remove the sensitivity observed in the current problem. For example, computations were made employing objective functions defined as: (1) the sum of the maximum surface velocity on the blade suction surface plus the surface velocities on either side of that point, and (2) the sum of the first five surface velocities near the leading edge on the blade suction surface. The design variables and side constraints were kept the same as those of the original problem. The hope was that by spreading the objective function over a wider region on the blade surface, that the extreme sensitivity of the problem would be reduced. The results of these calculations, however, were disappointing in that they provided uniformily inferior results to the original choice of the objective function as a point quantity [Eq. (16)]. Our conclusion with regard to improving the posing of this particular optimization problem is that two issues must be addressed that were beyond the scope of this investigation. The first concerns the definition of the flow solution in the vicinity of the peak suction pressure. A detailed examination should be made of the accuracy and reliability of the flow solver (TSONIC) in that region. The second issue concerns the reliability of the basic CONMIN optimizer itself. In many of the optimization calculations undertaken in this study, a notable characteristic of the CONMIN optimizer was its penchant to move to a shallow objective function minimum in the near vicinity of the baseline configuration and to remain there. Alteration of stepsizes, design variable scalings, and certain tolerances did not change this characteristic. This behavior of CONMIN has been noted by others (Ref. 18) who have concluded that the conjugate gradient algorithm has serious convergence deficiencies when applied to the class of aerodynamic optimization problems being considered in this study. An attractive alternative method based on a quasi-Newton algorithm has demonstrated significantly superior performance characteristics over COPES/CONMIN for these same classes of problems (Ref. 18). The CONMIN procedure was originally developed for structural design problems with large numbers of variables and constraints, rather than for the relatively new aerodynamic optimization problem which involves a relatively limited number of design variables and constraints. Consequently, it is not surprising that superior procedures are now being discovered and developed. In future work, an investigation should be made of the desirability of replacing the COPES/CONMIN optimization procedure embodied in the present BLDOPT program developed here.

4. CONCLUSIONS AND RECOMMENDATIONS

An investigation was conducted to complete the preliminary development of a combined perturbation/optimization procedure and associated computational code for designing optimized blade-to-blade profiles of turbomachinery blades. The overall purpose of the procedures developed in this study is to provide demonstration of the utility of a rapid nonlinear perturbation method for minimizing the computational requirements associated with optimized design studies of turbomachinery flows. The nonlinear perturbation method employed has been successfully developed in previous phases of this study and employs coordinate straining concepts together with unit perturbations determined from a special calibration matrix of nonlinear base solutions to predict families of related nonlinear solutions without further need of the computational nonlinear flow field solver. The solutions predicted can be either continuous or discontinuous.

The results reported here relate to the combination of the perturbation method, configured to predict simultaneous multiple-parameter changes, with the NASA/Lewis Research Center TSONIC code for predicting blade-to-blade flow solutions, the NASA/Lewis Research Center BLADE code for generating NASA blade-to-blade double circular arc blade shapes, and the NASA COPES-CONMIN code for performing optimization searches in multiple-parameter design space. The combined PERTURB/TSONIC/BLADE/COPES-CONMIN code, called BLDOPT, has been configured to perform optimization studies employing one, all, or any combination of the following eight blade geometry parameters used to characterize NASA double circular arc blade profiles: inlet blade camber angle, outlet blade camber angle, transition location between the inlet and outlet circular arc sections, maximum thickness location, inlet to outlet turning rate ratio, blade maximum thickness, leading edge radius, and trailing edge radius. Redefinition of the objective function and active side constraints for other case studies have been made simple and straightforward by confining their definition to one subroutine. The sample objective function and constraint definitions included in the version of the code reported here employ the velocity diffusion on the blade suction surface as objective function and three active side constraints related to maintenance of nonzero local blade thickness, low velocity diffusion on the blade pressure surface, and an effective trailing edge Kutta condition. The combined BLDOPT code has been made user-friendly and has been documented in a user's manual included as part of this report. An option has been included to allow the user to bypass use of the perturbation method altogether and employ TSONIC solutions throughout the optimization process in order to establish selected benchmark calculations. Options are also available to allow the user to employ the perturbation method in an automatic hands-off fashion. This is accomplished at the modest computational expense of one additional nonlinear

TSONIC-solution only optimization cycle over the alternative option of employing the perturbation method with the user supplying information regarding the initial calibration solution matrix.

Results of a series of calculations of the combined BLDOPT code have verified the code, demonstrated the accuracy of the perturbation-predicted results, and established benchmark guidelines of the potential for computational savings of the method under the various options included in the code. In general, the perturbation method is capable of providing an order of magnitude reduction in computational work in these applications.

Based on these results, we conclude that perturbation methods formulated on these ideas are both accurate and extremely workable in design environments. They clearly can provide the means for substantially reducing the computational work required in such applications. We suggest the further testing of the perturbation method with the combined BLDOPT code in order to test the limits of the method in such important preliminary design applications. Forthermore, we recommend the combination of these same procedures with supercritical flow solvers so as to accomplish both subcritical and supercritical blade optimization design.

APPENDIX A

USER'S MANUAL FOR COMPUTER PROGRAM BLDOPT

A.1 INTRODUCTION

The purpose of this appendix is to describe the operation of the computer code that was developed in conjunction with the theoretical work presented in this report, and to provide sufficient detail to permit convenient use and change of the program. The program determines an optimized blade shape, with respect to certain blade surface geometry parameters (Ref. 16) and based on TSONIC blade-to-blade flow solutions (Ref. 15), by employing a modified version of the COPES-CONMIN optimization search program (Ref. 13). The typically large computational demands of such optimization procedures caused by the need for numerous blade-to-blade flow solutions during the optimum search process is substantially reduced by incorporation of a novel, recently-developed, rapid, nonlinear, strained-coordinate perturbation method (Refs. 2 and 3) as discussed in the main text.

A description of the general operating procedure of the combined program is given, together with complete description of both input and output. The program is written in FORTRAN IV and has been developed on the Ames Research Center CDC 7600 computer facility. Approximate program run times for an optimization problem involving six design variables and 10 optimization search cycles when not employing or employing the perturbation method under the various program options are:

CPU Run Time	Program Option, IOPT
800 secs.	1 (TSONIC solutions only)
100 secs.	2 (Perturbation method)
180 secs.	3 (1 TSONIC solutions only cycle followed by perturbation method)

The storage requirements are $141K_8$ for small core memory and $77K_8$ for large core memory.

A.2 PROGRAM DESCRIPTION

The combined blade optimization program BLDOPT consists of the following main elements:

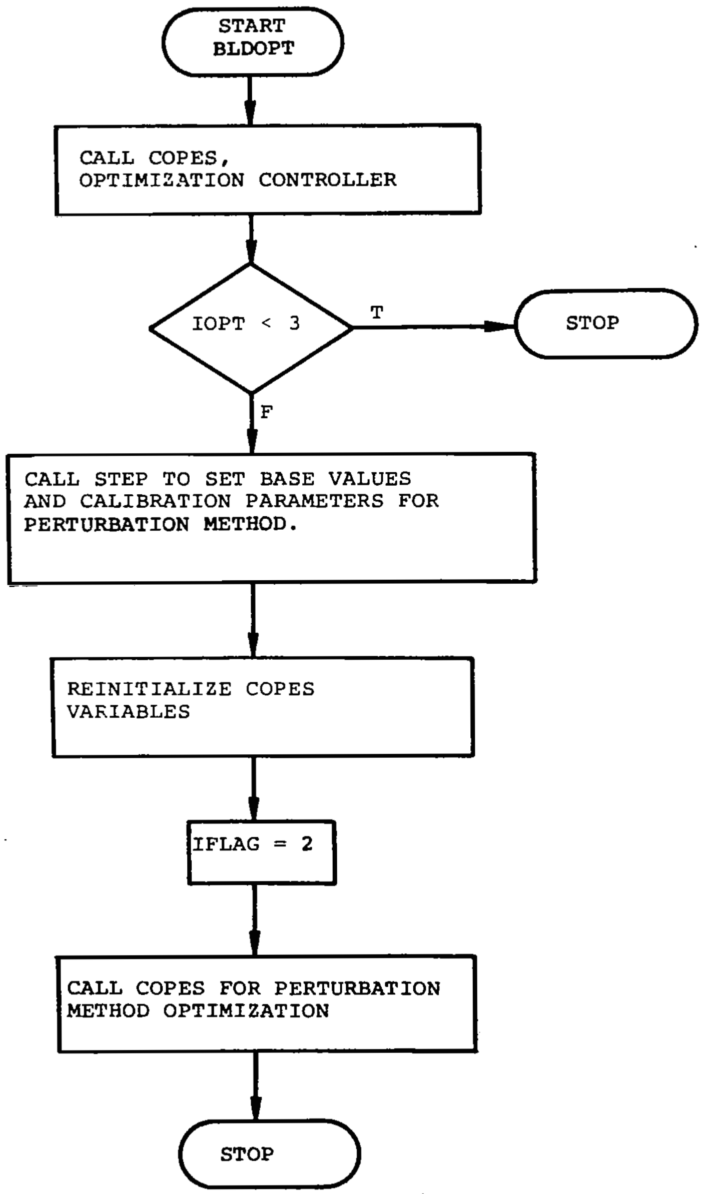
Code Element	Function
COPES-CONMIN[13]	Optimization procedure
TSONIC[15]	Turbomachinery blade-to-blade flow solver
BLADE[16]	Blade geometry description
PERTRB[3]	Nonlinear perturbation method

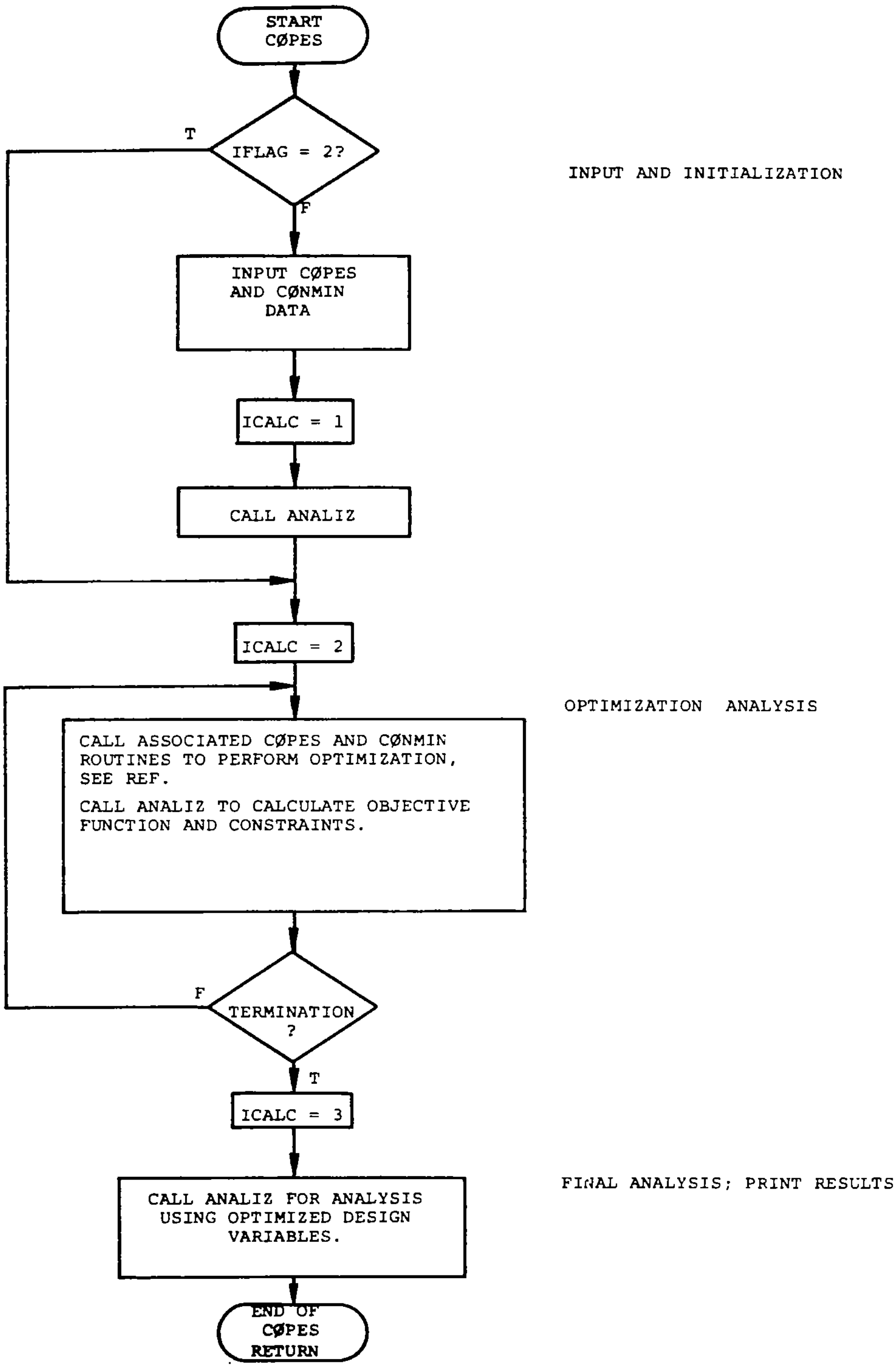
The program is configured to perform the optimization of a blade element as described geometrically in Reference 16. The optimization is based upon the following design variables which are geometric parameters describing the blade element:

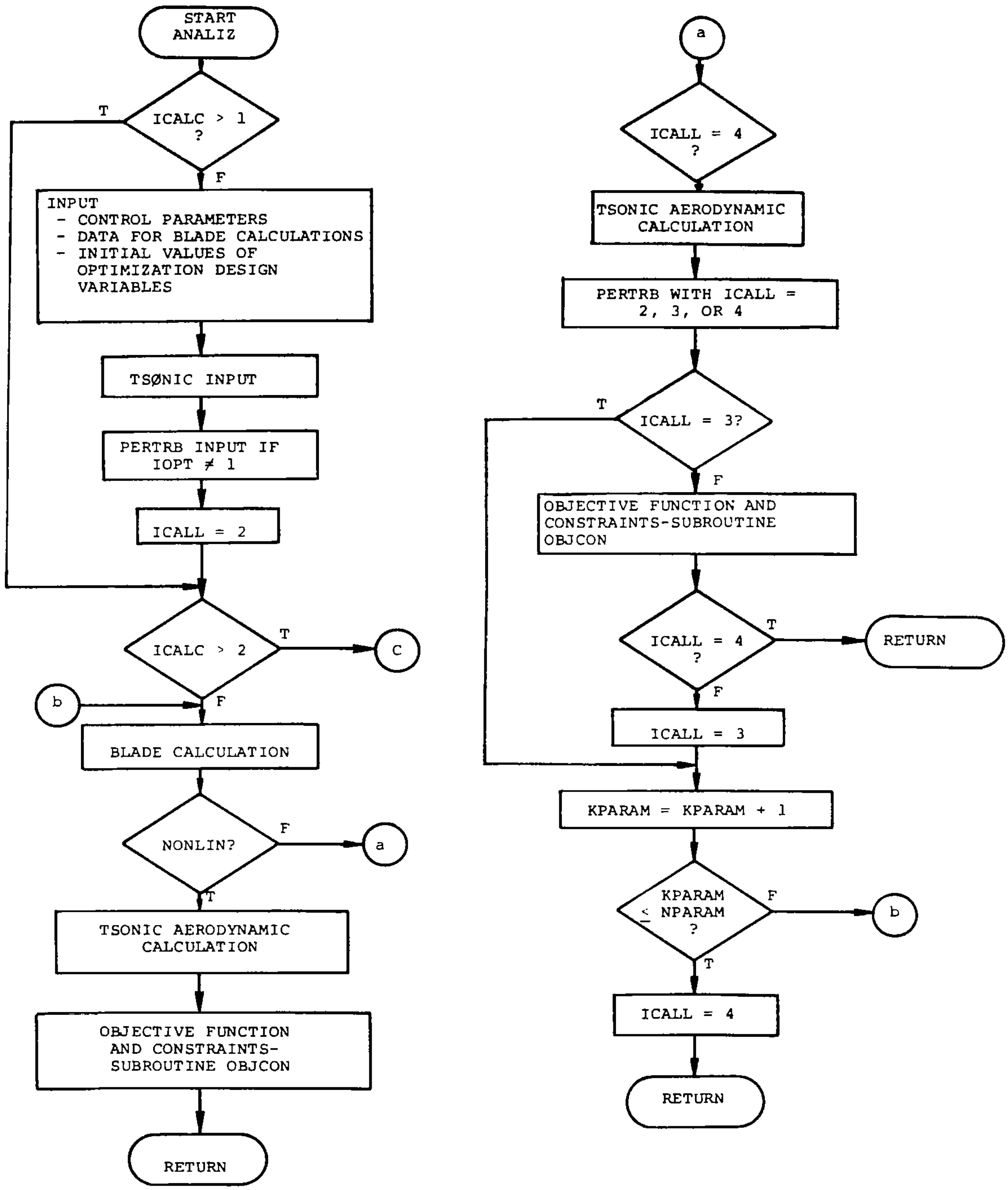
Design Variable Number	Geometric Parameter	Program Name
1	Blade camber angle at inlet	KICR
2	Blade camber angle at outlet	KOCR
3	Transition location/chord	T
4	Maximum thickness location/chord	ZM
5	Inlet/outlet turning rate ratio	P
6	Blade maximum thickness/chord	TMX
7	Leading edge radius/chord	THLE
8	Trailing edge radius/chord	THTE

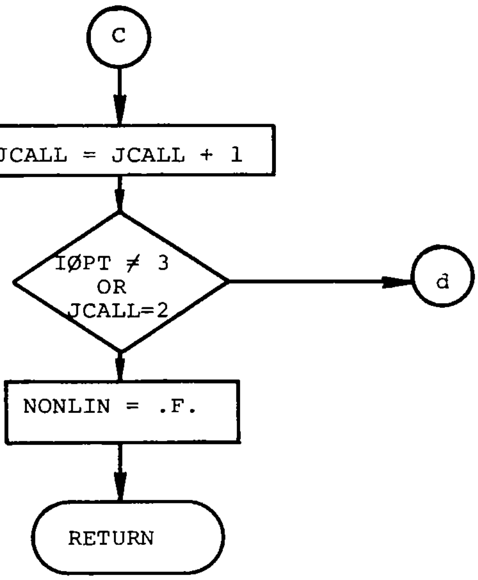
At the user's option, the optimization problem can be specified to employ any arbitrary combination up to six of the above design variables. Additionally, the user has the ability to construct readily particular objective functions and side constraints to be used in new optimization problems. Ease of definition and implementation of optimization problems with regard to objective function, side constraints, and design variable bounding were the primary reasons for selection of the COPES-CONMIN optimization driver. All information regarding definition of objective function and side constraints is contained in the subroutine OBJCON. In the program version listed here, the objective function is defined as the maximum velocity diffusion on the blade suction surface, and three side constraints are imposed to maintain (1) nonzero blade thickness, (2) low velocity diffusion on the blade pressure surface, and (3) trailing edge closure. Details of how these constraints are defined are provided in the main text, and their implementation in the code is straightforward and self-explanatory when viewing the program listing.

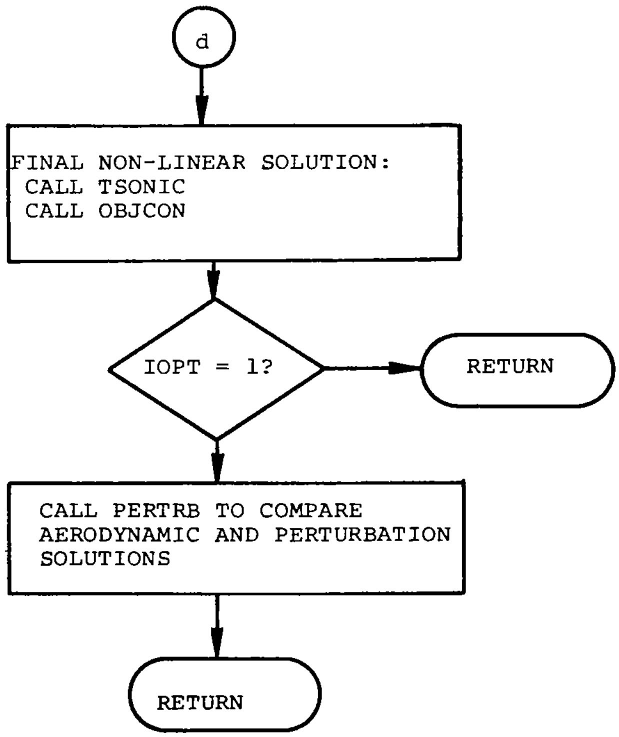
A.3 PROGRAM FLOW CHART




START CØPES
IFLAG = 2?
T
F
INPUT CØPES AND CØNMIN DATA
ICALC = 1
CALL ANALIZ
ICALC = 2
CALL ASSOCIATED CØPES AND CØNMIN ROUTINES TO PERFORM OPTIMIZATION, SEE REF.
CALL ANALIZ TO CALCULATE OBJECTIVE FUNCTION AND CONSTRAINTS.
TERMINATION ?
F
T
ICALC = 3
CALL ANALIZ FOR ANALYSIS USING OPTIMIZED DESIGN VARIABLES.
END OF CØPES RETURN
INPUT AND INITIALIZATION
OPTIMIZATION ANALYSIS
FINAL ANALYSIS; PRINT RESULTS


START ANALIZ
ICALC > 1 ?
T
F
INPUT
- CONTROL PARAMETERS
- DATA FOR BLADE CALCULATIONS
- INITIAL VALUES OF OPTIMIZATION DESIGN VARIABLES
TSØNIC INPUT
PERTRB INPUT IF IOPT ≠ 1
ICALL = 2
ICALC > 2
T
c
b
F
BLADE CALCULATION
NONLIN?
F
a
T
TSONIC AERODYNAMIC CALCULATION
OBJECTIVE FUNCTION AND CONSTRAINTS-SUBROUTINE OBJCON
RETURN
a
ICALL = 4 ?
TSONIC AERODYNAMIC CALCULATION
PERTRB WITH ICALL = 2, 3, OR 4
T
ICALL = 3?
F
OBJECTIVE FUNCTION AND CONSTRAINTS-SUBROUTINE OBJCON
ICALL = 4 ?
T
RETURN
F
ICALL = 3
KPARAM = KPARAM + 1
KPARAM ≤ NPARAM ?
F
b
T
ICALL = 4
RETURN


C
JCALL = JCALL + 1
IØPT ≠ 3
OR
JCALL=2
d
NONLIN = .F.
RETURN


d
FINAL NON-LINEAR SOLUTION:
CALL TSONIC
CALL OBJCON
IOPT = 1?
RETURN
CALL PERTRB TO COMPARE
AERODYNAMIC AND PERTURBATION
SOLUTIONS
RETURN

A.4 DICTIONARY OF INPUT VARIABLES

This section provides a dictionary of all input variables. The variables are divided into four sections corresponding to the four major parts of the program (see A.2).

A.4.1 Dictionary of Input Variables for Subroutine COPES (Ref. 13)

All COPES variables are defined in Section A.5.3.

A.4.2 Dictionary of Input Variables for Subroutine ANALIZ

ALP — Blade-element layout-cone half angle, degrees; see Reference 16.

DVCALB(I) — Array of dimension NDV, specifying the calibration parameters used in the perturbation method optimization

ICALB — Integer parameter which controls input of the array DVCALB:

ICALB = 0, do not input DVCALB
ICALB = 1, input DVCALB

IDV(I) — Array of integers, of dimension NDV, which is used to select a subset of variable blade parameters as design variables in the optimization. If IDV(I) = K, the *Ith* design variable is VV(K).

IOPT — Integer parameter which controls which method of analysis is to be used in optimization:

IOPT = 1, nonlinear aerodynamic solution
IOPT = 2, perturbation method
IOPT = 3, nonlinear optimization for one iteration to predict calibration stepsizes, followed by perturbation method.

ITMAX3 — Maximum number of optimization iterations allowed for the perturbation solution when IOPT = 3. Not used if IOPT = 1 or 2.

KICR — Centerline blade inlet angle on layout cone, degrees; see Reference 16.

KOCR — Centerline blade outlet angle on layout cone, degrees; see Reference 16.

NB	Number of blades
NCN	Number of constraint sets in the optimization problem; this value must be the same as that for NCONS in COPES input. NCN $\leq$ 5.
NDV	Number of independent design variables used in the optimization problem. This value must be the same as NDV in COPES input. NDV $\leq$ 8.
P	Ratio of inlet-segment turning ratio to outlet-segment turning rate for a blade element; see Reference 16.
PSTEP	Constant-value calibration stepsize, which may be used in the perturbation method optimization. If ICALB = 0, the value of the *Kth* calibration parameter DVCALB(K) = (1.+PSTEP)*QO(K), where QO(K) is the value of the *Kth* base solution parameter.
R	Inlet radius, length unit; see Reference 16.
SOLID	Blade tip solidity (chord/circumferential spacing); see Reference 16.
T	Blade centerline transition point location, made non-dimensional by the chord; see Reference 16.
THLE	Blade-element leading-edge circle radius, made non-dimensional by the chord; see Reference 16.
THTE	Blade-element trailing-edge circle radius, made non-dimensional by the chord; see Reference 16.
TMX	Blade-element maximum thickness, made non-dimensional by the chord; see Reference 16.
VNAME(I)	Array of dimension NDV, containing 10-character strings which identify, for printed output, the independent design variables used in optimization.
VV(I)	Array of dimension 8 which is equivalent to the list of input variable blade parameters: VV(1) = KICR VV(5) = P VV(2) = KOCR VV(6) = TMX VV(3) = T VV(7) = THLE VV(4) = ZM VV(8) = THTE
ZM	Blade centerline maximum thickness location, made non-dimensional by the chord; see Reference 16.

A.4.3 Dictionary of Input Variables for Subroutine TSONIC

This section presents definitions of the input variables required for the TSONIC flow analysis. For a complete description of the variables and their usage see Reference 15.

AR	Gas constant, J/(kg)(K)
BESP	Array of stream-channel normal thicknesses corresponding to the MR and RMSP arrays, meters, see Figure 12, Reference 15.
BETAI	Inlet flow angle β_{1e} along BG with respect to m-direction, deg, see Figure 11, Reference 15.
BETAO	Outlet flow angle β_{te} along CF with respect to m-direction, deg, see Figure 11, Reference 15.
DENTOL	Tolerance on density change per iteration for reduced weight flow (DENTOL may be left blank, and the value 0.01 will be used. If trouble is experienced in obtaining convergence (i.e., the maximum relative change in density (Item 14 of output, Ref. 15) does not get small enough, then a larger value of DENTOL may be used, or a smaller value of REDFAC may be used. (The value of 0.001 for DENTOL would be a tight tolerance, 0.01 is a medium tolerance, and 0.1 would be a loose tolerance.)
FSMI	m-coordinate corresponding to BETAI (it is assumed that FSMI $\leq$ 0); if not specified, FSMI = 0.
FSMO	m-coordinate corresponding to BETAO (it is assumed that FSMO $\geq$ chord); if not specified, FSMO = chord.
GAM	Specific heat ratio
LAMBDA	Upstream whirl, rV_θ, meters^2/sec.
LOPT	Integer variable which controls input of PLOSS array: LOPT = 0, PLOSS array is not given as input LOPT = 1, read in PLOSS array
LRVB	Integer variable which controls input: LRVB = 0, BETAI and BETAO are input LRVB = 1, LAMBDA and RVTHO are input
MBI	Number of vertical mesh lines from AH to BG inclusive, see Figure 13, Reference 15.

MBO	Number of vertical mesh lines from AH to CF inclusive, see Figure 13, Reference 15.
MM	Total number of vertical mesh lines in m-direction from AH to DE, maximum of 100, see Figure 13, Reference 15.
MOPT	Integer control variable: MOPT = 0, use only when REDFAC = 1, no correction is made to the BESP array for the reduced mass flow solution. MOPT = 1, read in the WOWCR array to use for calculating the reduced mass flow BESP array. MOPT = 2, the reduced BESP array will be calculated by the program using average blade angles and using SSM1 and SSM2 values to determine whether flow is subsonic or supersonic.
MR	Array of m-coordinates of spline points for stream-channel radii and stream-channel thickness, meters, see Figure 12, Reference 15 (MR is measured from the leading edge of the blade. These coordinates should cover the entire distance from AH to DE, and may extend beyond these bounds. The total number of points is NRSP.).
NBBI	Number of mesh spaces in θ-direction between AB and GH, maximum of 50, see Figure 13, Reference 15.
NBL	Number of blades.
MRSP	Number of spline points for stream-channel radius (RMSP) and thickness (BESP) coordinates, maximum of 50, see Figure 12, Reference 15.
OMEGA	Rotational speed, ω, rad/sec (note that ω is negative if rotation is in the opposite direction of that shown in Fig. 12, Ref. 15).
ORF	Value of overrelaxation factor to be used in the solution of the inner iteration simultaneous equations (if ORF = 0, the program calculates an estimated value for the overrelaxation factor. See page 25, Ref. 15 for discussion.).
PLOSS	array of fractional total pressure loss, $1 - \frac{P'}{P'_{ideal}}$, corresponding to the MR, RMSP, and BESP arrays.

REDFAC — Factor by which weight flow (WTFL) must be reduced in order to assure subsonic flow throughout passage (REDFAC is usually between 0.5 and 0.9).

RHOIP — Inlet stagnation density, kg/meter^3

RMSP — Array of r-coordinates of spline points for the stream-channel radii, corresponding to the MR array, meters, see Figure 12, Reference 15.

RVTHO — Downstream whirl, $(rV_r)_o$, meters^2/sec.

SPLNO(1), SPLNO(2) — Number of blade spline points given for each surface as input, maximum of 50 [these include the first and last points (dummies) that are tangent to the leading- and trailing-edge radii (Fig. 11, Ref. 15)].

SSM1 — m-coordinate where supersonic solution is to start.

SSM2 — m-coordinate where supersonic solution is to end. (Note: If SSM1 and SSM2 are both left blank, there will be no supersonic region).

TIP — Inlet stagnation temperature, K

WOWCR — Array of W/W_{cr} values at mid-channel, corresponding to the MR, RMSP, and BESP arrays. Used to calculate the reduced mass flow BESP array.

WTFL — Mass flow per blade for stream channel, kg/sec.

The remaining variables, starting with BLDAT, are used to indicate what output is desired. A value of 0 for any of these variables will cause the output associated with that variable to be omitted. A value of 1 will cause the corresponding output to be printed for the final iteration only; 2, for the first and final iterations; and 3, for all iterations. Case should be used not to call for more output than is really useful. The following list gives the output associated with each of these variables:

BLDAT — All geometrical information which does not change from iteration to iteration (i.e., coordinates and first and second derivatives of all blade surface spline points; blade coordinates, blade slopes, and blade curvatures where vertical mesh lines meet each blade surface; radii and stream-channel thickness corresponding to each vertical mesh line; m-coordinate, stream-channel radius and thickness, and blade surface angles and slopes where horizontal mesh lines intersect each blade; and ITV and IV arrays, internal variables describing the location of the blade surfaces with respect to the finite difference grid).

AANDK Coefficient array, constant vector, and indexes of all adjacent points for each point in finite-difference mesh (this information is needed for debugging the program only).

ERSOR Maximum change in stream function at any point for each iteration of SOR equation [Eq. (A8), Ref. 17].

STRFN Value of stream function at each unknown mesh point in region.

SLCRD Streamline θ-cordinates at each vertical mesh line, and streamline plot.

INTVL Velocity and flow angle at each interior mesh point for both reduced and actual weight flow.

SURVL m-coordinate, surface velocity, flow angle, distance along surface, and W/W_{cr} based on meridional velocity components where each vertical mesh line meets each blade surface; m-coordinate, surface velocity, flow angle, distance along surface, and W/W_{cr} based on tangential velocity components where each horizontal mesh line meets each blade surface, plot of blade surface velocities against meridional streamline distance, meters.

A.4.4 Dictionary of Input Variables for Subroutine PERTRB

This section presents definitions of the input variables required for the perturbation method. For a complete description of the variables and their usage see Reference 3.

A Scaling parameter in straining procedure. A = -x(1), where x(1) is x-location of first data point on lower surface (see PROGRAM DESCRIPTION, Ref. 3).

B Scaling parameter in straining procedure. B = -x(N), where x(N) is location of last data point on upper surface (see PROGRAM DESCRIPTION, Ref. 3).

LPLOT Specifies whether or not an additional plot by a peripheral device is to be made. Software must be supplied by user in subroutine DRVPLT.

LPLOT = 0 ... no peripheral plot
LPLOT = 1 ... peripheral plot

LSELCT(I) — Array of length 6 of which NSELCT elements are read in; specifies nature of points to be held invariant according to the code:

1 ... minimum point held invariant
2 ... maximum point held invariant
3 ... 1st critical point held invariant
4 ... 2nd critical point held invariant
5 ... 3rd critical point held invariant
6 ... 4th critical point held invariant

Note that critical point ordering is determined from order of occurrence starting at the lower surface at the point furthest from the leading edge and proceeding clockwise around the surface (see PROGRAM DESCRIPTION, Ref. 3).

Note that the code numbers can be assigned in any order, e.g.,

LSELCT(1) = 1		LSELCT(1) = 4
LSELCT(2) = 3	and	LSELCT(2) = 1
LSELCT(3) = 4		LSELCT(3) = 3

Are equivalent, both corresponding to NSELCT = 3, with the minimum, and first and second critical points held invariant.

LUNIT — Controls whether or not unit coordinate strainings and unit perturbation(s) are printed.

LUNIT = 0 ... no output
LUNIT = 1 ... output

NSELCT — Number of points (in addition to end points) to be held invariant in straining; note: $1 \leq \text{NSELCT} \leq 6$.

PARNAM(K) — Array of 8-character strings which identify the parameters varied. NPARAM element of the array are read in.

TITLE — Character string of length 80; identifies job. First nine characters are used to identify peripheral plot.

VNAM — Character string of length 2 which symbolizes dependent variables, e.g., "CP" for pressure coefficient.

A.5 PREPARATION OF INPUT DATA

This section describes the preparation of the card input data for the program. A description of each input item is presented, followed by a description of the card format.

A.5.1 Description of Input

The data are divided into four sections corresponding to the four major parts of the program.

A.5.1.1 COPES Input

Item 1.1 To end of COPES input. Input for the COPES optimization control subroutines is described in Section A.5.3. The user should take particular note of variable ITMAX in Block C and the array X in Block R. If IOPT = 3, ITMAX should be set equal to 1, and all values of the array X should be set to zero.

A.5.1.2 ANALIZ Input

Item 2.1 One card, containing text which identifies the ANALIZ input block; may contain up to 80 characters.

Item 2.2 One card, containing the control parameters IOPT, NDV, NCN, ITMAX3.

Item 2.3 One card, containing the constant blade parameters NB, R, ALP, and SOLID.

Item 2.4 One card, containing the variable blade parameters KICR, KOCR, T, ZM, P, TMX, THLE, THTE. The initial values of the design variables used in optimization are contained in this set.

Item 2.5 One card, containing the character strings VNAME(I), I = 1, NDV.

Item 2.6 One card, containing the integer array IDV(I), I = 1, NDV.

Item 2.7 One card, containing parameters ICALB and PSTEP.

Item 2.9 One card, optional, containing perturbation parameters. DVCALB(I), I = 1, NDV. Omit this item if ICALB = 0.

A.5.1.3 TSONIC Input

Item 3.1 One card, containing text identifying the TSONIC input block; may contain up to 80 characters.

Item 3.2 One card, containing values for GAM, AR, TIP, RHOIP, WTFL, OMEGA, and ORF.

Item 3.3 One card, containing values for geometric parameters BETAI, BETAO, FSMI, and FSMO.
Optional: LAMBDA and RVTHO may be specified instead of BETAI and BETAO. In this case, specify LRVB = 1. FSMI and FSMO are not needed.

Item 3.4 One card, containing values for REDFAC, DENTOL, SSM1, SSM2.

Item 3.5 One card, containing integral MBI, MBO, MM, NBBI, NBL, NRSP, MOPT, LOPT, LRVB.

Item 3.6 One card, containing SPLNO(1).

Item 3.7 One card, containing SPLNO(2).

Item 3.8 Set of cards, eight values per card, containing array MR(I), I = 1, NRSP.

Item 3.9 Set of cards, eight values per card, containing array RMSP(I), I = 1, NRSP.

Item 3.10 Set of cards, eight values per card, containing array BESP(I), I = 1, NRSP.

Item 3.11 Set of cards, optional, eight values per card, containing array WOWCR(I), I = 1, NRSP. This item is required only when MOPT = 1.

Item 3.12 Set of cards, optional, eight values per card, contain-array PLOSS(I), I = 1, MRSP. Omit this item if LOPT = 0.

Item 3.13 One card, containing the integer print control variables BLDAT, AANDK, ERSOR, STRFN, SLCRD, INTVL, and survl.

Item 3.14 One card, containing the character string $END in columns 1-4. This identifies the end of TSONIC input.

A.5.1.4 PERTRB Input

Item 4.1 One card of text, identifying the PERTRB input block. This item may contain up to 80 characters.

Item 4.2 One card, containing TITLE for job identification. This item may contain up to 80 characters. The first 9 characters are used to identify peripheral plot, if LPLOT = 1.

Item 4.3 One card, containing the integer control parameters NSELCT, LUNIT, and LPLOT.

Item 4.4 One card, containing NSELCT values of the integer array LSELCT.

Item 4.5 One card, containing the character string VNAM.

Item 4.6 One card, containing the character strings PARNAM(I), I = 1, NDV.

Item 4.7 One card, containing the scaling parameters A and B.

A.5.2 Format of Input Data

A.5.2. Format of Input Data

Item 1.1 - end of COPES input   see Section A.5.3

Item no.2.1   1 card (8A10)

Variable	Text
Card column	80
Format type	A

Item no.2.2   1 card (10I5)

Variable	IØPT	NDV	NCN	ITMAX3
Card column	5	10	15	20
Format type	I	I	I	I

Item no.2.3   1 card (I10,3F10.6)

Variable	NB	R	ALP	SØLID
Card column	10	20	30	40
	I	F	F	F

Item no. 2.4   1 card (8F10.0)

Variable	KICR	KØCR	T	ZM	P	TMX	THLE	THTE
Card column	10	20	30	40	50	60	70	80
Format type	F	F	F	F	F	F	F	F

Item no. 2.5   1 card (8A10)

Variable	VNAME(1)	VNAME(2)	---	---	VNAME(NDV)			
Card column	10	20	30	40	50	60	70	80
Format type	A	A	A	A	A	A	A	A

Item no. 2.6   1 card (10I5)

Variable	IDV(1)	IDV(2)	---	---	IDV(NDV)			
Card column	5	10	15	20	25	30	35	40
Format type	I	I	I	I	I	I	I	I

Item no. 2.7 1 card (I10, 3F10.6)

Variable	ICALB	PSTEP
Card column	10	20
Format type		

Item no. 2.8 1 card (8F10) Read only when ICALB not zero.

Variable	DVCALB(1)	DVCALB(2)	----	---	DVCALB(NDV)			
Card column	10	20	30	40	50	60	70	80
Format type	F	F	F	F	F	F	F	F

Item no. 3.1 1 card (8A10)

Variable	Text
Card column	80
Format type	A

Item no. 3.2 1 card (8F10.5)

Variable	GAM	AR	TIP	RHØIP	WTFL		ØMEGA	ØRF
Card column	10	20	30	40	50	60	70	80
Format type	F	F	F	F	F	F	F	F

Item no. 3.3 1 card (8F10.5)

Variable	BETAI	BETAØ			FSMI	FSMØ
Card column	10	20	30	40	50	60
Format type	F	F	F	K	F	F

Item no. 3.4 1 card (8F10.5)

Variable	REDFAC	DENTØL	SSM1	SSM2
Card column	10	20	30	
Format type	F	F	F	F

Item no. 3.5   1 card (16I5)

Variable	MBI	MBØ			MM	NBBI	NBL	NRSP	MØPT	LØPT	LRVB
Card column	5	10	15	20	25	30	35	40	45	50	55
Format type	I	I	I	I	I	I	I	I	I	I	I

Item no. 3.6   1 card (8F10.5)

Variable					SPLNØ(1)
Card column	10	20	30	40	50
Format type					F

Item no. 3.7   1 card (8F10.5)

Variable					SPLNØ(2)
Card column	10	20	30	40	50
Format type					F

Item no. 3.8   J cards, J=INT((NRSP-1)/8)+1, 8 values per card (8F10.5)

Variable	MR(1)	MR(2)	----	----	MR(NRSP)			
Card column	10	20	30	40	50	60	70	80
Format type	F	F	F	F	F	F	F	F

Item no. 3.9   J cards, J as above, 8 values per card (8F10.5)

Variable	RMSP(1)	RMSP(2)	----	----	RMSP(NRSP)			
Card column	10	20	30	40	50	60	70	80
Format type	F	F	F	F	F	F	F	F

Item no.3.10   J cards, J as above, 8 values per card (8F10.5)

Variable	BESP(1)	BESP(2)	----	----	BESP(NRSP)			
Card column	10	20	30	40	50	60	70	80
Format type	F	F	F	F	F	F	F	F

Item no.3.11 (If MØPT=1) J cards, J as above, 8 values per card (8F10.5)

Variable	WØWCR(1)	WØWCR(2)	----	----	WØWCR(NRSP)			
Card column	10	20	30	40	50	60	70	80
Format type	F	F	F	F	F	F	F	F

Item no.3.12 (If LØPT≠0) J cards, J as above, 8 values per card (8F10.5)

Variable	PLØSS(1)	PLØSS(2)	----	----	PLØSS(NRSP)			
Card column	10	20	30	40	50	60	70	80
Format type	F	F	F	F	F	F	F	F

Item no.3.13 1 card (16I5)

Variable	BLDAT	AANDK	ERSØR	STRFN	SLCRD	INTVL	SURVL
Card column	5	10	15	20	25	30	35
Format type	I	I	I	I	I	I	I

Item no.3.14 1 card (8A10)

Variable	$END
Card column	10
Format type	A

Item no. 4.1 1 card (8A10)

Variable	TEXT
Card column	80
Format type	A

Item no. 4.2 1 card (8A10)

Variable	TITLE
Card column	80
Format type	A

Item no. 4.3 1 card (16I5)

Variable	NSELCT	LUNIT	LPLØT
Card column	5	10	15
Format type	I	I	I

Item no. 4.4 1 card (16I5)

Variable	LSELCT(1)	LSELCT(2)	-----	-----	LSELCT(NSELCT)			
Card column	5	10	15	20	25	30	35	40
Format type	I	I	I	I	I	I	I	I

Item no. 4.5 1 card (2A1)

Variable	VNAM
Card column	2
Format type	A

Item no. 4.6 1 card (10A8)

Variable	PARNAM(1)	PARNAM(2)	---	----	PARNAM(NDV)			
Card column	8	16	24	32	40	48	56	64
Format type	A	A	A	A	A	A	A	A

Item no. 4.7 1 card (8F10.6)

Variable	A	B
Card column	10	20
Format type	F	F

A.5.3 Format of COPES Input Data.

DATA BLOCK A

DESCRIPTION: Title card.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
TITLE								20A4
CANTILEVERED BEAM DESIGN								

FIELD	CONTENTS
1-8	Any 80 character title may be given on this card.

DATA BLOCK B

DESCRIPTION: Program Control Parameters.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7		FORMAT
NCALC	NDV	NSV	N2VAR	NXAPRX	IPNPUT	IPDBG		7I10
2	2	3	5	2	0	0		

FIELD CONTENTS

1 NCALC: Calculation Control

0 - Read input and stop. Data of blocks A, B and V is required. Remaining data is optional.

1 - One cycle through program. The same as executing ANALIZ stand-alone. Data of blocks A, B and V is required. Remaining data is optional.

2 - Optimization. Data of blocks A-I and V is required. Remaining data is optional.

3 - Sensitivity analysis. Data of blocks A, B, P, Q and V is required. Remaining data is optional.

4 - Two variable function space. Data of blocks A, B, and R-V is required. Remaining data is optional.

FIELD		CONTENTS
1 - cont.	NCALC:	
	5 -	Optimum Sensitivity. Data of blocks A-K and V is required. Remaining data is optional.
	6 -	Optimization using approximation techniques. Data of blocks A-O and V is required. Remaining data is optional.
2	NDV:	Number of independent design variables in optimization.
3	NSV:	Number of variables on which sensitivity analysis will be performed.
4	N2VAR:	Number of objective functions in a two variable function space study.
5	NXAPRX:	Number of X-variables for approximate analysis/optimization.
6	IPNPUT:	Input print control.
	0 -	Print card images of data plus formatted print of input data.
	1 -	Formatted print only of input data.
	2 -	No print of input data.
7	IPDBG:	Debug print control.

DATA BLOCK C OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Integer optimization control parameters.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
IPRINT	ITMAX	ICNDIR	NSCAL	ITRM	LINOBJ	NACMX1	NFDG	8I10
5	0	0	0	0	0	0	0	

FIELD — CONTENTS

1 IPRINT: Print control used in the optimization program CONMIN.

0 - No print during optimization.

1 - Print initial and final optimization information.

2 - Print above plus objective function value and design variable values at each iteration.

3 - Print above plus constraint values, direction vector and move parameter at each iteration.

4 - Print above plus gradient information.

5 - Print above plus each proposed design vector, objective function and constraint values during the one-dimensional search.

FIELD	CONTENTS
2	ITMAX: Maximum number of optimization iterations allowed. DEFAULT = 20.
3	ICNDIR: Conjugate direction restart parameter. DEFAULT = NDV + 1.
4	NSCAL: Scaling parameter. GT.0 - Scale design variables to order of magnitude one every NSCAL iterations. LT.0 - Scal design variables according to user-input scaling values.
5	ITRM: Number of consecutive iterations which must satisfy relative or absolute convergence criterion before optimization process is terminated. DEFAULT = 3.
6	LINOBJ: Linear objective function identifier. If the optimization objective is known to be a linear function of the design variables, set LINOBJ = 1. DEFAULT = Non-linear.
7	NACMX1; One plus the maximum number of active constraints anticipated. DEFAULT = NDV + 2.
8	NFDG: Finite difference gradient identifier. 0 - All gradient information is computed by finite difference within CONMIN. 1 - All gradient information is computed analytically by the user-supplied code. 2 - Gradient of objective is computed analytically. Gradients of constraints are computed by finite difference within CONMIN.

REMARKS

1) Currently NFDG must be zero in COPES.

DATA BLOCK D OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Floating point optimization program parameters.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7		FORMAT
FDCH	FDCHM	CT	CTMIN	CTL	CTLMIN	THETA		7F10
0.0	0.0	0.0	0.0	0.0	0.0	0.0		
DELFUN	DABFUN	ALPHAX	ABOBJ1					4F10
0.0	0.0	0.0	0.0					

NOTE: TWO CARDS ARE READ HERE.

FIELD		CONTENTS
1	FDCH:	Relative change in design variables in calculating finite difference gradients. DEFAULT = 0.01.
2	FDCHM:	Minimum absolute step in finite difference gradient calculations. DEFAULT = 0.001.

FIELD	CONTENTS	
3	CT:	Constraint thickness parameter. DEFAULT = -0.05.
4	CTMIN:	Minimum absolute value of CT considered in the optimization process. DEFAULT = 0.004.
5	CTL:	Constraint thickness parameter for linear constraints. DEFAULT = -0.01.
6	CTLMIN:	Minimum absolute value of CTL considered in the optimization process. DEFAULT = 0.001.
7	THETA:	Mean value of push-off factor in the method of feasible directions. DEFAULT = 1.0.
1	DELFUN:	Minimum relative change in objective function to indicate convergence of the optimization process. DEFAULT = 0.001.
2	DABFUN:	Minimum absolute change in objective function to indicate convergence of the optimization process. DEFAULT = 0.001 times the initial objective value.
3	ALPHAX:	Maximum fractional change in any any design variable for first estimate of the step.in the one-dimensional search. DEFAULT = 0.1.
4	ABOBJ1:	Expected fractional change in the objective function for first estimate of the step in the one-dimensional search. DEFAULT = 0.1.

REMARKS

1) The DEFAULT values for these parameters usually work well.

DATA BLOCK E OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Total number of design variables, design objective identification and sign.

FORMAT AND EXAMPLE

1	2	3		FORMAT
NDVTOT	IOBJ	SGNOPT		2I10,F10
0	3	-1.0		

FIELD CONTENTS

1 NDVTOT: Total number of variables linked to the design variables. This option allows two or more parameters to be assigned to a single design variable. The value of each parameter is the value of the design variable times a multiplier, which may be different for each parameter. DEFAULT = NDV.

2 IOBJ: Global variable location associated with the objective function in optimization.

3 SGNOPT: Sign used to identify whether function is to be maximized or minimized. +1.0 indicates maximization. -1.0 indicates minimization. If SGNOPT is not unity in magnitude, it acts as a multiplier as well, to scale the magnitude of the objective.

DATA BLOCK F OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Design variable bounds, initial values and scaling factors.

FORMAT AND EXAMPLE

1	2	3	4		FORMAT
VLB	VUB	X	SCAL		4F10
.5	5.	0.0	0.0		

NOTE: READ ONE CARD FOR EACH OF THE NDV INDEPENDENT DESIGN VARIABLES.

FIELD		CONTENTS
1	VLB:	Lower bound on the design variable. If VLB.LT.-1.0E+15, no lower bound.
2	VUB:	Upper bound on the design variable. If VUB.GT.10.E+15, no upper bound.
3	X:	Initial value of the design variable. If X is non-zero, this will supercede the value initialized by the user-supplied subroutine ANALIZ.
4	SCAL:	Design variable scale factor. Not used if NSCAL.GE.0 in BLOCK C.

DATA BLOCK G OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Design variable identification.

FORMAT AND EXAMPLE

1	2	3		FORMAT
NDSGN	IDSGN	AMULT		2I10,F10
1	1	1.0		

NOTE: READ ONE CARD FOR EACH OF THE NDVTOT DESIGN VARIABLES.

FIELD		CONTENTS
1	NDSGN:	Design variable number associated with this variable.
2	IDSGN:	Global variable number associated with this variable.
3	AMULT:	Constant multiplier on this variable. The value of the variable will be the value of the design variable, NDSGN, times AMULT. DEFAULT = 1.0.

DATA BLOCK H OMIT IF NDV = 0 IN BLOCK B

DESCRIPTION: Number of constrained parameters.

FORMAT AND EXAMPLE

1		FORMAT
NCONS		I10
4		

FIELD	CONTENTS
1	NCONS: Number of constraint sets in the optimization problem.

REMARKS

1) If two or more adjacent parameters in the global common block have the same limits imposed, these are part of the same constraint set.

DATA BLOCK I OMIT IF NDV = 0 IN BLOCK B, OR NCONS = 0 IN BLOCK H

DESCRIPTION: Constraint identification and constraint bounds.

FORMAT AND EXAMPLE

1	2	3	4		FORMAT
ICON	JCON	LCON			3I10
4	0	0			
BL	SCAL1	BU	SCAL2		
-1.0 +20	0.0	20000.	0.0		

NOTE: READ TWO CARDS FOR EACH OF THE NCONS CONSTRAINT SETS.

FIELD		CONTENTS
1	ICON:	First global number corresponding to the constraint set.
2	ICON:	Last global number corresponding to the constraint set. DEFAULT = ICON.
3	LCON:	Linear constraint identifier for this constraint set. LCON = 1 indicates linear constraints.

FIELD		CONTENTS
1	BL:	Lower bound on the constrained variables. If BL.LT.-1.0E+15, no lower bound.
2	SCAL1:	Normalization factor on lower bound. DEFAULT = MAX of ABS(BL), 0.1.
3	BU:	Upper bound on the constrained variables. If BU.GT.1.0E+15, no upper bound.
4	SCAL2:	Normalization factor on upper bound. DEFAULT = MAX of ABS(BU), 0.1.

REMARKS

1) The normalization factor should usually be defaulted.
2) The constraint functions sent to CONMIN are of the form;
   (BL - VALUE)/SCAL1 .LE. 0.0 and (VALUE - BU)/SCAL2 .LE. 0.0.
3) Each constrained parameter is converted to two constraints in CONMIN unless ABS(BL) or ABS(BU) exceeds 1.0E+15, in which case no constraint is created for that bound.

DATA BLOCK J OMIT IF NXAPRX = 0 IN BLOCK B

DESCRIPTION: Approximate analysis/optimization control parameters.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NF	NXS	NXFS	NXA	INOM	ISCRX	ISCRXF	IPAPRX	8I10
5	1	1	1	0	0	0	1	
KMIN	KMAX	NPMAX	JNOM	INXLOC	INFLOC			6I10
0	0	0	0	0	0			

FIELD		CONTENTS
1	NF:	Number of functions to be approximated. Default = number of optimization objective and constraint functions.
2	NXS:	Number of X-vectors read as data.
3	NXFS:	Number of X-F pairs read as data.
4	NXA:	If non-zero, the design variables read by SUBROUTINE ANALIZ form an X-vector.
5	INOM:	Nominal X-vector. Default = best available.

FIELD	CONTENTS
6	ISCRX: File from which NXS X-vectors are read. Default = 5.
7	ISCRXF: File from which NXFS X-F pairs of data are read. Default = 5.
1	KMIN: Minimum number of approximation iterations.
2	KMAX: Maximum number of approximation iterations.
3	NPMAX: Maximum number of designs retained for Tayler series expansion.
4	JNOM: Number of iterations after which the best design is picked as nominal.
5	INXLOC: X-variable global location identifier. If INXLOC = 0, the Tayler series expansion is on the design variables listed in BLOCK G.
6	INFLOC: Function global location identifier. If INFLOC = 0, the Objective and constraint functions identified in BLOCKS E and I are the functions on which the Tayler series expansion is performed.

REMARKS

1) If ISCRX and/or ISCRXF file number is other than 5, the data read from that file is assumed to be binary data.

2) If NXS = NXFS = 0, NXA is defaulted to NXA = 1, even is it is read as zero. Also, a second vector of design variables is automatically defined by COPES to yield two independent designs to start the optimization.

DATA BLOCK K OMIT IF NDV = 0 IN BLOCK B, OR NXAPRX = 0 IN BLOCK B

DESCRIPTION: Bounds and multipliers for approximate optimization.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
DX1	DX2	DX3	DX4	DX5	...	...	...	8F10
.5	2.							
XFACT1	XFACT2							2F10
0.	0.							

NOTE: TWO OR MORE CARDS ARE READ HERE.

FIELD		CONTENTS
1-8	DXI:	Allowable change (in magnitude) of the Ith design variable during each approximate optimization.
1	XFACT1:	Multiplier on DXI when the diagonal elements of the H matrix are available. Default = 1.5.
2	XFACT2:	Multiplier on DXI when all elements of the H matrix are available. Default = 2.0.

DATA BLOCK L OMIT IF NXAPRX = 0 IN BLOCK B OR INXLOC = 0 IN BLOCK J

DESCRIPTION: Global locations of approximating variables.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
LOCX1	LOCX2	LOCX3	LOCX4	.....	.....	.....	.....	8I10
1	2							

NOTE: MORE THAN ONE CARD MAY BE READ HERE.

FIELD	CONTENTS
1-8	LOCI: Global location of Ith approximating variable.

REMARKS

1) If INXLOC = 0 in BLOCK J, this data is not read. In this case, the data is defaulted to be the global locations of the design variables (IDSGN values in BLOCK G).

DATA BLOCK M OMIT IF NXAPRX = 0 IN BLOCK B OR INFLOC = 0 IN BLOCK J

DESCRIPTION: Global locations of functions to be approximated.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
LOCF1	LOCF2	LOCF3	LOCF4	.....	.....	.....	.....	8I10
3	5	6	4					

NOTE: MORE THAN ONE CARD MAY BE READ HERE.

FIELD	CONTENTS
1-8	LOCI: Global location of Ith function to be approximated.

REMARKS

1) If INFLOC = 0 in BLOCK J, this data is not read. In this case, the data is defaulted to be the global locations of the objective function (IOBJ in BLOCK E) followed by the global locations of the constrained parameters (ICON, JCON in BLOCK I).

DATA BLOCK N OMIT IF NXS = 0 IN BLOCK J

DESCRIPTION: X-Vectors for approximate optimization.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
XI1	XI2	XI3	XI4	...	...	...	...	8F10
4.	15.							

NOTE: NXS SETS OF DATA ARE READ HERE.

NOTE: MORE THAN ONE CARD MAY BE READ FOR EACH SET OF DATA.

FIELD	CONTENTS
1-8	XIJ: Jth value of Ith X-vector, J = 1,NXAPRX.

DATA BLOCK O OMIT IF NXFS = 0 IN BLOCK J

DESCRIPTION: X-F pairs of information for approximate optimization.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
X1	X2	X3	X4	...	...	...	...	8F10
2.	18.							
Y1	Y2	Y3	Y4	Y5	...	...	...	
7200.	416.667	.914495	18518.519					

NOTE: NXFS SETS OF DATA ARE READ HERE.

NOTE: MORE THAN ONE CARD MAY BE REQUIRED FOR XI OR YI.

NOTE: NXAPRX VALUES OF X AND NF VALUES OF Y ARE READ FOR EACH SET OF DATA.

FIELD	CONTENTS
1-8	XI: Ith value of X, I = 1,NXAPRX.
1-8	YI: Ith value of Y, I = 1,NF.

DATA BLOCK P OMIT IF NSV = 0 IN BLOCK B

DESCRIPTION: Sensitivity objectives.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NSOBJ	IPSENS							2I10
5	0							
NSN1	NSN2	NSN3	NSN4	NSN5	...	...	...	8I10
3	4	5	6	7				

NOTE: TWO OR MORE CARDS ARE READ HERE.

FIELD	CONTENTS	
1	NSOBJ:	Number of seperate objective functions to be calculated as functions of the sensitivity variables.
2	IPSENS:	Print control. If IPSENS.GT.0, detailed print will be called at each step in the sensitivity analysis. DEFAULT = No print.
1-8	NSNI:	Global variable number associated with the sensitivity objective functions.

REMARKS

1) More than eight sensitivity objectives are allowed. Add data cards as required to contain data.

DATA BLOCK Q OMIT IF NSV = 0 IN BLOCK B

DESCRIPTION: Sensitivity variables.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8 --	FORMAT
ISENS	NSENS							2I10
9	4							
SNS1	SNS2	SNS3	SNS4	....	....	....	....	8F10
200.	100.	150.	250.					

NOTE: READ ONE SET OF DATA FOR EACH OF THE NSV SENSITIVITY VARIABLES.

NOTE: TWO OR MORE CARDS ARE READ FOR EACH SET OF DATA.

FIELD		CONTENTS
1	ISENS:	Global variable number associated with the sensitivity variable.
2	NSENS:	Number of values of this sensitivity variable to be read on the next card.
1-8	SENSI;	Values of the sensitivity variable. I = 1,NSENS. I = 1 corresponds to the nominal value.

REMARKS

1) More than eight values of the sensitivity variable are allowed. Add data cards as required to contain the data.

DATA BLOCK R OMIT IF N2VAR = 0 IN BLOCK B

DESCRIPTION: Two variable function space control parameters.

FORMAT AND EXAMPLE

1	2	3	4	5		FORMAT
N2VX	M2VX	N2VY	M2VY	IP2VAR		5I10
1	4	2	5	0		

FIELD	CONTENTS
1	N2VX: Global location of the X-variable in the two variable function space.
2	M2VX: Number of values of X to be considered.
3	N2VY: Global location of the Y-variable in the two variable function space.
4	M2VY: Number of values of Y to be considered.
5	IP2VAR: Print control. If IP2VAR.GT.0, detailed print will be called at each step (each X-Y combination). DEFAULT = No print.

DATA BLOCK S OMIT IF N2VAR = 0 IN BLOCK B

DESCRIPTION: Objective functions of the two variable function space study.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
NZ1	NZ2	NZ3	NZ4	NZ5	...	...	...	8I10
3	4	5	6	7				

FIELD CONTENTS

1-8 NZI: Global location corresponding to the Ith function of X and Y to be calculated. N2VAR values are read here.

REMARKS

1) More than eight objective functions are allowed. Add data cards as required to contain the data.

DATA BLOCK T OMIT IF N2VAR = 0 IN BLOCK B

DESCRIPTION: Values of the X-variable in a two variable function space study.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
X1	X2	X3	X4	...	...	...	...	8F10
0.5	1.0	1.5	2.0					

FIELD		CONTENTS
1-8	XI:	Values of the X-variable in the two variable function space. M2VX values are read here.

REMARKS

1) More than eight values are allowed. Add data cards as required to contain the data.

DATA BLOCK U OMIT IF N2VAR = 0 IN BLOCK B

DESCRIPTION: Values of the Y-variable in a two variable function space study.

FORMAT AND EXAMPLE

1	2	3	4	5	6	7	8	FORMAT
Y1	Y2	Y3	Y4	Y5	...	...	...	8F10
4.0	8.0	12.0	16.0	20.0				

FIELD CONTENTS

1-8 YI: Values of the Y-variable in the two variable function space. M2VY values are read here.

REMARKS

1) More than eight values are allowed. Add data cards as required to contain the data.

DATA BLOCK V

DESCRIPTION: COPES data 'END' card.

FORMAT AND EXAMPLE

1		FORMAT
END		3A1
END		

FIELD	CONTENTS
1	The word 'END' in columns 1-3.

REMARKS

1) This card MUST appear at the end of the COPES data.

2) This ends the COPES input data.

3) Data for the user-supplied routine, ANALIZ, follows this.

A.6 DESCRIPTION OF OUTPUT

The first output item consists of a banner page related to the COPES control program and followed by card images of the COPES input data. The next output items contain COPES optimization information which will be employed by the optimization program. This is followed by a display of input information required by the ANALIZ subroutine and related to the control parameters, constant blade properties, variable blade properties, active design variables, and input information for the calibration solution matrix. Next, is a display of input information required by the TSONIC blade-to-blade flow solver. Finally, a display is provided of the input information needed by the perturbation method.

The next items of output are related to the TSONIC solution for the baseline blade profile. The first item consists of the input information on design variables and constant blade properties that is provided to the blade element program which then performs the computation necessary to determine blade property characteristics required as input to the TSONIC code. The next items are the input to the TSONIC code and the output generated by the code for the flow solution. This is followed by output from OBJCON related to the objective function and active constraints. The next item is the banner page of information that will initiate the CONMIN minimization procedure. The optimization search cycles are then begun. The output that follows depends upon which IOPT option was specified. However, regardless of the IOPT option selected, each time a TSONIC flow solution is required by the optimization package, the following segment of output is produced: the input information regarding design variables and constant blade properties that is provided to the blade element program is displayed, followed by the input to the TSONIC code, the output from TSONIC, and finally, output from OBJCON related to the current objective function and active side constraint values for the TSONIC solution just calculated. Consequently, for IOPT = 1, for which TSONIC solutions are used throughout the optimization process, the above segment of information is displayed for each TSONIC solution for the gradient and search calculations. At the end of each iteration cycle, CONMIN provides output regarding the ending values of the design variables and side constraints. This continues until the ITMAX limit is reached, or the objective function has not changed within the last three iterations, at which point the final optimization results are printed, followed by a final TSONIC calculation at the design point. For the IOPT = 2 option, after the base solution is computed, the calibration solution matrix is determined based on user-supplied information. The optimization cycles then proceed using perturbation-predicted solutions for the blade surface velocities. When the optimization process is complete, a final TSONIC solution is calculated at the design point and a printer plot provided exhibiting a comparison of the perturbation-predicted surface velocity distribution with the TSONIC result. For the IOPT = 3 option, the optimization process is allowed to proceed with TSONIC solutions only as with IOPT = 1, but only for one search cycle. Then, based on the new base design point reached, design variable values for a calibration solution matrix are determined. Next, each of the TSONIC solutions for the new base and calibration solutions are separately determined, with output provided after each solution regarding number of critical points, and strained coordinate and surface pressure arrays. Following the final calibration solution, summary output is provided regarding the base coordinates, the strained coordinate arrays for each calibration solution, and the corresponding surface velocity arrays. Next, the CONMIN optimization search cycles are entered using the perturbation method to predict all flow solutions required in the gradient and search calculations. The search cycles are continued to the ITMAX3 limit or until the objective function does not change for three successive iterations, at which point the final optimization results are printed, followed by a final TSONIC calculation at the design point, and a printer plot illustrating comparisons between the perturbation method predicted surface velocities distribution and the nonlinear TSONIC result.

A.7 ERROR MESSAGES

NUMBER OF CRITICAL POINTS IN
BASE AND CALIBRATION SOLUTIONS
ARE UNEQUAL - CALCULATION ENDED

This message will be printed if critical points are specified in straining (LSPEC = 0) and the number of critical points in base and calibration solutions are unequal. The remedy is to avoid use of critical points in straining, or to use base and calibration solutions having equal numbers of critical points.

NUMBER OF CRITICAL POINTS
SELECTED EXCEEDS NUMBER
ACTUALLY LOCATED - CALCULATION
ENDED

This message will be printed if more critical points are specified in straining (LSPEC = 0) than the number located by the program. The remedy is to specify a number of points less than or equal to the actual number.

ORDER OF SPECIFIED POINTS IN
BASE AND CALIBRATION SOLUTIONS
DOES NOT CORRESPOND - CALCULATION
ENDED

This message will be printed if the fixed points specified (LSPEC = 0) occur in a different sequence in the base and calibration solutions. The remedy is to use base and calibration solutions having the same qualitative features.

A.8 SAMPLE CASE

The sample case presented in this section provides example results of the perturbation method for the blade design optimization problem described in Section 3 of the main text. The calculation is for the simultaneous six design variable (KOCR, T, ZM, P, TMX, THLE) optimization of full potential turbomachinery flows past compressor blades having NASA double circular arc blade profiles with the following initial values, lower, and upper bounds, respectively, of these parameters: (-10.0, 0.25, 0.45, 1.5, 0.05, 0.005), (-15.0, 0.20, 0.50, 0.03, 0.003), (0.00, 0.60, 0.55, 4.00, 0.10, 0.012).

The input data is tabulated in Figure A.1 with the COPES data appearing first, followed by the inputs for subroutine ANALIZ, subroutine TSONIC, and subroutine PERTURB. We note that in the input for subroutine PERTURB, the number of invariant points to be held invariant for this calculation were chosen to be one (NSELCT = 1), and that particular point was chosen as the maximum point (LSELCT(1) = 2, i.e.), the stagnation point, and that the dependent variable for print output will be symbolized by 'WM' denoting the surface speed. Examination of the sample input and comparison with the description of input data provided in Sections A.5.1 to A.5.3 provides a convenient, self-explanatory menu of how to prepare typical input data sets for future case studies.

Finally, Figure A.2 provides an abbreviated print output for this sample case. That output, together, with the information contained in Appendix A, provides a benchmark result that can be employed to completely verify the BLDOPT code on any user facility.

```
$---DATA BLOCK A
OPTIMIZATION TEST CASE 1 - MINIMIZE SUCTION SURFACE VELOCITY DIFFUSION
$---DATA BLOCK B
$    NCALC       NDV       NSV     N2VAR    NXAPRX    IPNPUT     IPDBG
         2         6
$---DATA BLOCK C
$  IPRINT      ITMAX    ICNDIR     NSCAL      ITRM    LINOBJ    NACMX1      NDFG
       5         1                    -2         3                  20
$---DATA BLOCK D
$ ALL DEFAULTS EXCEPT FDCH AND FDCHM
$     FDCH      FDCHM
      0.01    0.00100
$   DELFUN     DABFUN    ALPHAX    ABOBJ1
       0.0        0.0       0.0      0.05
$---DATA BLOCK E
$   NDVTOT      IOBJ    SGNOPT
                   1      -1.0
$---DATA BLOCK F
$      VLB        VUB         X      SCAL
$ OUTLET BLADE ANGLE - KOCR
     -15.0        0.0       0.0     -10.0
$ TRANSITION POINT LOCATION - T
       0.2       0.60       0.0      0.25
$ MAXIMUM THICKNESS LOCATION - ZM
       0.2       0.55       0.0      0.45
$ INLET/OUTLET TURNING RATE - P
       0.5        4.0       0.0       1.5
$ MAXIMUM THICKNESS - TMX
      0.03       0.10       0.0      0.05
$ LEADING EDGE RADIUS - THLE
     0.003      0.012       0.0     0.005
$---DATA BLOCK G
$    NDSGN      IDSGN     AMULT
$ OUTLET BLADE ANGLE - KOCR
         1          2       1.0
$ TRANSITION POINT LOCATION - T
         2          3       1.0
$ MAXIMUM THICKNESS LOCATION - ZM
         3          4       1.0
$ INLET/OUTLET TURNING RATE - P
         4          5       1.0
$ MAXIMUM THICKNESS - TMX
         5          6       1.0
$ LEADING EDGE RADIUS - THLE
         6          7       1.0
$---DATA BLOCK H
$    NCONS
         3
$---DATA BLOCK I
$     ICON       JCON      LCON
$       BL      SCAL1        BU     SCAL2
$ LOCAL BLADE THICKNESS CONSTRAINT  -  BLTKS
         8
       0.0                  10.
$ PRESSURE SURFACE DIFFUSION  -  DIFFP
         9
       0.0                  1.6
$ TRAILING EDGE CLOSURE  -  TECLSR
        10
       -1.                   1.
$---DATA BLOCKS J-U NOT REQUIRED
```

Figure A.1.- Card input for sample case.

```
$---DATA BLOCK V
END
     ***  INPUT FOR SUBROUTINE ANALIZ  ***
    3    6    3    6
         34   0.454     6.664     2.252
  52.0      -10.0      0.25      0.45       1.5       0.05      0.005      0.005
KOCR      T         ZM        P          TMX       THLE
    2    3    4    5    6    7
         0    0.0
     ***  INPUT FOR SUBROUTINE TSONIC  ***
1.400     1716.48   599.76    .00334586 .00570000            0.0       1.910
48.2      0.0       0.0       0.0       0.0       0.0
1.0       0.001     0.0       0.0
   24   56             72   15   34   13    0    1    0
                                        13.0
                                        13.0
-.13      -.005     .005      .02003    .04003    .08003    .10003    .12503
.15004    .17526    .18526    .22500    .27
.4540     .4540     .4540     .4540     .4540     .4540     .4540     .4540
.4540     .4540     .4540     .4540     .4540
.05       .05       .0499     .0496     .04925    .0485     .04815    .04770
.04720    .04675    .04661    .04661    .04661
0.        0.        .0015     .0059     .0110     .0215     .0300     .0335
.0400     .0470     .048      .048      .048
    0    0    0    0    0    0    1
$END
     ***  INPUT FOR SUBROUTINE PERTRB  ***
OPTIMIZATION TEST CASE 1 - MINIMIZE SUCTION SURFACE VELOCITY DIFFUSION
    1    1    0
    2
WM
  KOCR      T       ZM        P       TMX     THLE
      -1.0       1.0
```

Figure A.1.- Concluded.

```
CCCCCCC   0000000   PPPPPPP   EEEEEEE   SSSSSSS
C         0     0   P     P   E         S
C         0     0   P     P   E         S
C         0     0   PPPPPPP   EEEE      SSSSSSS
C         0     0   P         E               S
C         0     0   P         E               S
CCCCCCC   0000000   P         EEEEEEE   SSSSSSS


         C O N T R O L   P R O G R A M

                     F O R

   E N G I N E E R I N G   S Y N T H E S I S



                   T I T L E

OPTIMIZATION TEST CASE 1 - MINIMIZE SUCTION SURFACE VELOCITY DIFFUSION
```

Figure A.2.- Abbreviated print output for sample case.

CARD IMAGES OF CONTROL DATA

```
CARD                    IMAGE

  1)  $---DATA BLOCK A
  2)  OPTIMIZATION TEST CASE 1 - MINIMIZE SUCTION SURFACE VELOCITY DIFFUSION
  3)  $---DATA BLOCK B
  4)  $    NCALC       NDV        NSV      N2VAR    NXAPRX    IPNPUT     IPDBG
  5)           2         6
  6)  $---DATA BLOCK C
  7)  $   IPRINT     ITMAX    ICNDIR      NSCAL       ITRM    LINOBJ    NACMX1       NDFG
  8)           5         1                   -2          3                  20
  9)  $---DATA BLOCK D
 10)  $ ALL DEFAULTS EXCEPT FDCH AND FDCHM
 11)  $     FDCH     FDCHM
 12)        0.01   0.00100
 13)  $   DELFUN    DABFUN    ALPHAX    ABOBJ1
 14)         0.0       0.0       0.0      0.05
 15)  $---DATA BLOCK E
 16)  $   NDVTOT      IOBJ    SGNOPT
 17)                     1      -1.0
 18)  $---DATA BLOCK F
 19)  $      VLB       VUB         X      SCAL
 20)  $ OUTLET BLADE ANGLE - KOCR
 21)       -15.0       0.0       0.0     -10.0
 22)  $ TRANSITION POINT LOCATION - T
 23)         0.2      0.60       0.0      0.25
 24)  $ MAXIMUM THICKNESS LOCATION - ZM
 25)         0.2      0.55       0.0      0.45
 26)  $ INLET/OUTLET TURNING RATE - P
 27)         0.5       4.0       0.0       1.5
 28)  $ MAXIMUM THICKNESS - TMX
 29)        0.03      0.10       0.0      0.05
 30)  $ LEADING EDGE RADIUS - THLE
 31)       0.003     0.012       0.0     0.005
 32)  $---DATA BLOCK G
 33)  $    NDSGN     IDSGN    AMULT
 34)  $ OUTLET BLADE ANGLE - KOCR
 35)           1         2      1.0
 36)  $ TRANSITION POINT LOCATION - T
 37)           2         3      1.0
 38)  $ MAXIMUM THICKNESS LOCATION - ZM
 39)           3         4      1.0
 40)  $ INLET/OUTLET TURNING RATE - P
 41)           4         5      1.0
 42)  $ MAXIMUM THICKNESS - TMX
 43)           5         6      1.0
 44)  $ LEADING EDGE RADIUS - THLE
 45)           6         7      1.0
 46)  $---DATA BLOCK H
 47)  $    NCONS
 48)           3
 49)  $---DATA BLOCK I
 50)  $     ICON      JCON      LCON
 51)  $       BL     SCAL1        BU     SCAL2
 52)  $ LOCAL BLADE THICKNESS CONSTRAINT  -  BLTKS
 53)           8
 54)         0.0                 10.
 55)  $ PRESSURE SURFACE DIFFUSION  -  DIFFP
 56)           9
```

Figure A.2.- Continued.

```
        0.0                  1.6
$ TRAILING EDGE CLOSURE  -  TECLSR
         10
        -1.                  1.
$---DATA BLOCKS J-U NOT REQUIRED
$---DATA BLOCK V
END
```

Figure A.2.- Continued.

TITLE:
OPTIMIZATION TEST CASE 1 - MINIMIZE SUCTION SURFACE VELOCITY DIFFUSION

CONTROL PARAMETERS:

CALCULATION CONTROL,	NCALC	=	2
NUMBER OF GLOBAL DESIGN VARIABLES,	NDV	=	6
NUMBER OF SENSITIVITY VARIABLES,	NSV	=	0
NUMBER OF FUNCTIONS IN TWO-SPACE,	N2VAR	=	0
NUMBER OF APPROXIMATING VAR.	NXAPRX	=	0
INPUT INFORMATION PRINT CODE,	IPNPUT	=	0
DEBUG PRINT CODE,	IPDBG	=	0

CALCULATION CONTROL, NCALC

VALUE	MEANING
1	SINGLE ANALYSIS
2	OPTIMIZATION
3	SENSITIVITY
4	TWO-VARIABLE FUNCTION SPACE
5	OPTIMUM SENSITIVITY
6	APPROXIMATE OPTIMIZATION

• • OPTIMIZATION INFORMATION

GLOBAL VARIABLE NUMBER OF OBJECTIVE = 1
MULTIPLIER (NEGATIVE INDICATES MINIMIZATION) = -.1000E+01

CONMIN PARAMETERS (IF ZERO, CONMIN DEFAULT WILL OVER-RIDE)

IPRINT	ITMAX	ICNDIR	NSCAL	ITRM	LINOBJ	NACMX1	NFDG
5	1	0	-2	3	0	20	0

FDCH	FDCHM	CT	CTMIN
.10000E-01	.10000E-02	0.	0.

CTL	CTLMIN	THETA	PHI
0.	0.	0.	0.

DELFUN	DABFUN	ALPHAX	ABOBJ1
0.	0.	0.	.50000E-01

DESIGN VARIABLE INFORMATION
NON-ZERO INITIAL VALUE WILL OVER-RIDE MODULE INPUT

D. V. NO.	LOWER BOUND	UPPER BOUND	INITIAL VALUE	SCALE
1	-.15000E+02	0.	0.	-.10000E+02
2	.20000E+00	.60000E+00	0.	.25000E+00
3	.20000E+00	.55000E+00	0.	.45000E+00
4	.50000E+00	.40000E+01	0.	.15000E+01
5	.30000E-01	.10000E+00	0.	.50000E-01
6	.30000E-02	.12000E-01	0.	.50000E-02

DESIGN VARIABLES

ID	D. V. NO.	GLOBAL VAR. NO.	MULTIPLYING FACTOR
1	1	2	.10000E+01

Figure A.2.- Continued.

2	2	3	.10000E+01
3	3	4	.10000E+01
4	4	5	.10000E+01
5	5	6	.10000E+01
6	6	7	.10000E+01

CONSTRAINT INFORMATION

THERE ARE 3 CONSTRAINT SETS

ID	GLOBAL VAR. 1	GLOBAL VAR. 2	LINEAR ID	LOWER BOUND	NORMALIZATION FACTOR	UPPER BOUND	NORMALIZATION FACTOR
1	8	0	0	0.	.10000E+00	.10000E+02	.10000E+02
3	9	0	0	0.	.10000E+00	.16000E+01	.16000E+01
5	10	0	0	-.10000E+01	.10000E+01	.10000E+01	.10000E+01

TOTAL NUMBER OF CONSTRAINED PARAMETERS = 3

• • ESTIMATED DATA STORAGE REQUIREMENTS

REAL			INTEGER		
INPUT	EXECUTION	AVAILABLE	INPUT	EXECUTION	AVAILABLE
57	779	5000	34	112	1000

Figure A.2.- Continued.

--- INPUT FOR CONTROL SUBROUTINE ANALIZ ---

CONTROL PARAMETERS

IOPT = 3 NDV = 6 NCN = 3 ITMAX3 = 6

CONSTANT BLADE PARAMETERS

NB	R	ALP	SOLID
34	.4540	6.6640	2.2520

VARIABLE BLADE PARAMETERS

KICR	KOCR	T	ZM	P	TMX	THLE	THTE
52.0000	-10.0000	.2500	.4500	1.5000	.0500	.0050	.0050

ACTIVE DESIGN VARIABLES

KOCR	T	ZM	P	TMX	THLE
-10.0000	.2500	.4500	1.5000	.0500	.0050

ICALB = 0 PSTEP = 0.000

Figure A.2.- Continued.

```
--- INPUT FOR TSONIC BLADE-TO-BLADE FLOW SOLVER ---

  GAM            AR             TIP            RHOIP           WTFL                            OMEGA          ORF
1.400000       1716.480       599.7600       .3345860E-02    .5700000E-02    0.             0.             1.910000
  BETAI          BETAO          CHORDF         STGRF           FSMI           FSMO
48.20000       0.             0.             0.              0.             0.
  REDFAC         DENTOL         SSM1           SSM2
1.000000       .1000000E-02   0.             0.

MBI   MBO               MM   NBBI   NBL   NRSP   MOPT   LOPT   LRVB
 24    56    0     0    72    15    34     13     0      1      0

 BLADE SURFACE 1  --  UPPER SURFACE
  RI1            RO1            BETI1          BETO1           SPLNO1
0.             0.             0.             0.              13.00000
  MSP1  ARRAY
0.             0.             0.             0.              0.             0.             0.             0.
0.             0.             0.             0.              0.
  THSP1  ARRAY
0.             0.             0.             0.              0.             0.             0.             0.
0.             0.             0.             0.              0.

 BLADE SURFACE 2  --  LOWER SURFACE
  RI2            RO2            BETI2          BETO2           SPLNO2
0.             0.             0.             0.              13.00000
  MSP2  ARRAY
0.             0.             0.             0.              0.             0.             0.             0.
0.             0.             0.             0.              0.
  THSP2  ARRAY
0.             0.             0.             0.              0.             0.             0.             0.
0.             0.             0.             0.              0.

  MR  ARRAY
-.1300000      -.5000000E-02  .5000000E-02   .2003000E-01    .4003000E-01   .8003000E-01   .1000300       .1250300
.1500400       .1752600       .1852600       .2250000        .2700000
  RMSP  ARRAY
.4540000       .4540000       .4540000       .4540000        .4540000       .4540000       .4540000       .4540000
.4540000       .4540000       .4540000       .4540000        .4540000
  BESP  ARRAY
.5000000E-01   .5000000E-01   .4990000E-01   .4960000E-01    .4925000E-01   .4850000E-01   .4815000E-01   .4770000E-01
.4720000E-01   .4675000E-01   .4661000E-01   .4661000E-01    .4661000E-01
  PLOSS ARRAY
0.             0.             .1500000E-02   .5900000E-02    .1100000E-01   .2150000E-01   .3000000E-01   .3350000E-01
.4000000E-01   .4700000E-01   .4800000E-01   .4800000E-01    .4800000E-01

 BLDAT  AANDK  ERSOR  STRFN  SLCRD  INTVL  SURVL
   0      0      0      0      0      0      1
```

Figure A.2.- Continued.

```
--- INPUT FOR PERTURBATION METHOD ---

**************************************************************************************************************************
*                          OPTIMIZATION TEST CASE 1 - MINIMIZE SUCTION SURFACE VELOCITY DIFFUSION                           *
**************************************************************************************************************************

.....LIST OF INPUT PARAMETERS

     N =  62

     A = -1.0    B =  1.0

     NPARAM = 6


.....STRAINING OPTIONS

     NUMBER OF FIXED POINTS: 3

     FIXED POINTS WILL BE AUTOMATICALLY DETERMINED
     BY THE PROGRAM FOR ALL SOLUTIONS AS FOLLOWS:

          TWO END POINTS
          POINT OF MAXIMUM WM
```

Figure A.2.- Continued.

*** INPUT FOR BLADE ELEMENT PROGRAM ***

NO. BLADES	INLET RADIUS (L)	CONE ANGLE (DEG)	SOLIDITY	INLET BLADE ANGLE (DEG)	OUTLET BLADE ANGLE (DEG)	TRANS. LOC. /CHORD	MAX.TH. LOC. /CHORD	IN/OUT TURNING RATE	MAX. THICK. /CHORD	L.E. RAD. /CHORD	T.E. RAD. /CHORD
34	.45400	6.66400	2.25200	52.000	-10.0000	.25000	.45000	1.50000	.05000	.00500	.00500

Figure A.2.- Continued.

--- INPUT FOR TSONIC BLADE-TO-BLADE FLOW SOLVER ---

```
  GAM          AR           TIP          RHOIP          WTFL                              OMEGA         ORF
1.400000     1716.480     599.7600     .3345860E-02   .5700000E-02   0.                0.           1.910000
 BETAI        BETAO        CHORDF        STGRF          FSMI                FSMO
48.20000     0.           .1834184     .1311528       0.             0.
 REDFAC       DENTOL        SSM1         SSM2
1.000000     .1000000E-02 0.           0.

MBI   MBO                  MM   NBBI   NBL   NRSP   MOPT   LOPT   LRVB
 24    56     0      0     72    15     34    13      0      1      0

BLADE SURFACE 1  --  UPPER SURFACE
  RI1          RO1          BETI1        BETO1          SPLNO1
.9668364E-03 .9668364E-03 57.62951     -13.13649      13.00000
  MSP1  ARRAY
.1503113E-03 .5980107E-02 .1529854E-01 .2744253E-01 .4447412E-01 .6273379E-01 .8187670E-01 .1016760
.1218970     .1423000     .1585911     .1727172     .1826858
  THSP1  ARRAY
.1140045E-02 .1958139E-01 .4336945E-01 .6726818E-01 .9178661E-01 .1116067     .1269242     .1375985
.1435478     .1447487     .1423112     .1377348     .1330935

BLADE SURFACE 2  --  LOWER SURFACE
  RI2          RO2          BETI2        BETO2          SPLNO2
.9668364E-03 .9668364E-03 46.20771     -5.047831      13.00000
  MSP2  ARRAY
.1664865E-02 .8669538E-02 .1911677E-01 .3189241E-01 .4893131E-01 .6676995E-01 .8523057E-01 .1041871
.1235102     .1430679     .1587932     .1725519     .1823521
  THSP2  ARRAY
-.1473028E-02 .1370723E-01 .3337744E-01 .5345856E-01 .7484201E-01 .9279699E-01 .1074271     .1186613
.1264542     .1307850     .1317585     .1308042     .1290985

  MR  ARRAY
-.1300000    -.5000000E-02 .5000000E-02 .2003000E-01 .4003000E-01 .8003000E-01 .1000300     .1250300
.1500400     .1752600     .1852600     .2250000     .2700000
  RMSP  ARRAY
.4540000     .4540000     .4540000     .4540000     .4540000     .4540000     .4540000     .4540000
.4540000     .4540000     .4540000     .4540000     .4540000
  BESP  ARRAY
.5000000E-01 .5000000E-01 .4990000E-01 .4960000E-01 .4925000E-01 .4850000E-01 .4815000E-01 .4770000E-01
.4720000E-01 .4675000E-01 .4661000E-01 .4661000E-01 .4661000E-01
  PLOSS ARRAY
0.           0.           .1500000E-02 .5900000E-02 .1100000E-01 .2150000E-01 .3000000E-01 .3350000E-01
.4000000E-01 .4700000E-01 .4800000E-01 .4800000E-01 .4800000E-01

BLDAT  AANDK  ERSOR  STRFN  SLCRD  INTVL  SURVL
  0      0      0      0      0      0      1
```

Figure A.2.- Continued.

RELATIVE VELOCITY	MERIDIONAL VELOCITY	CRITICAL VELOCITY	REL. FLOW ANGLE
AT M = FSMI	AT M = FSMI	AT M = FSMI	AT UPSTREAM BDY.
745.90	497.17	1095.9	48.237
AT M = FSMO	AT M = FSMO	AT M = FSMO	AT DOWNSTREAM BDY.
499.45	499.45	1095.9	0.

FSMI = 0.
FSMO = .18342

CALCULATED PROGRAM CONSTANTS

PITCH	HT	HM1
.1847996	.1231997E-01	.5731826E-02

ITMIN	ITMAX
0	25

LAMBDA	DOWNSTREAM WHIRL (RVTHO)
252.4466	0.

REDUCED WEIGHT FLOW = .5700000E-02

NUMBER OF INTERIOR MESH POINTS = 1035

CALCULATED VELOCITY DIAGRAM INFORMATION

	IM	W	W/WCR	BETA
UPSTREAM BOUNDARY	1	745.47	.68022	48.237
LEADING EDGE	24	745.90	.68061	48.200
TRAILING EDGE	56	499.45	.45573	0.
DOWNSTREAM BOUNDARY	72	500.06	.45629	0.

Figure A.2.- Continued.

```
     ITERATION  NO.   1
     MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .3059       AT IM =  0,   IT = 11,   SURF = 1,   M =  .1779
     AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1105
     NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  86


     ITERATION  NO.   2
     MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .2681       AT IM =  0,   IT = 11,   SURF = 1,   M =  .1779
     AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .2909E-01
     NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  86


     ITERATION  NO.   3
     MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .4396       AT IM =  0,   IT = 11,   SURF = 1,   M =  .1779
     AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1291E-01
     NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  85


     ITERATION  NO.   4
     MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  3.182       AT IM =  0,   IT = 11,   SURF = 1,   M =  .1779
     AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .3816E-01
     NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  53
 DENSTY CALL NO.  9
 NER(1) =  1
 RHO*W IS  1.2430 TIMES THE MAXIMUM VALUE FOR RHO*W


     ITERATION  NO.   5
     MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .8418       AT IM =  0,   IT = 11,   SURF = 1,   M =  .1779
     AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1041E-01
     NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  22


     ITERATION  NO.   6
     MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .3818       AT IM =  0,   IT = 11,   SURF = 1,   M =  .1779
     AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .4669E-02
     NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  12


     ITERATION  NO.   7
     MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  1.655       AT IM =  0,   IT = 11,   SURF = 1,   M =  .1779
     AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1857E-01
     NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =   8
 DENSTY CALL NO.  9
 NER(1) =  2
 RHO*W IS  1.2455 TIMES THE MAXIMUM VALUE FOR RHO*W


     ITERATION  NO.   8
     MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .7274       AT IM =  0,   IT = 11,   SURF = 1,   M =  .1779
     AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .8167E-02
```

Figure A.2.- Continued.

```
     NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =    1


     ITERATION  NO.   9
     MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .3803       AT IM =  0,   IT = 11,   SURF = 1,   M =  .1779
     AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .4269E-02
     NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =    1


     ITERATION  NO.  10
     MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  1.635       AT IM =  0,   IT = 11,   SURF = 1,   M =  .1779
     AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1820E-01
     NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =    1
 DENSTY CALL NO.  9
 NER(1) =  3
 RHO*W IS  1.2472 TIMES THE MAXIMUM VALUE FOR RHO*W


     ITERATION  NO.  11
     MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .7251       AT IM =  0,   IT = 11,   SURF = 1,   M =  .1779
     AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .8072E-02
     NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =    1
```

Figure A.2.- Continued.

SURFACE VELOCITIES BASED ON MERIDIONAL COMPONENTS - REDUCED WEIGHT FLOW

	BLADF SURFACE 1				BLADE SURFACE 2			
M	VELOCITY	ANGLE(DEG)	SURF. LENGTH	W/WCR	VELOCITY	ANGLE(DEG)	SURF. LENGTH	W/WCR
0.	0.	90.00	0.	0.	0.	-90.00	0.	0.
.5732F-02	871.6	52.87	.1030E-01	.7953	608.5	44.24	.6680E-02	.5552
.1146E-01	909.2	48.47	.1935E-01	.8296	579.5	41.58	.1451E-01	.5287
.1720E-01	919.0	44.47	.2767E-01	.8385	554.2	39.02	.2202E-01	.5057
.2293E-01	910.1	40.64	.3545E-01	.8304	536.5	36.57	.2927E-01	.4895
.2866E-01	884.7	36.99	.4281E-01	.8073	524.8	34.15	.3630E-01	.4789
.3439E-01	848.7	33.70	.4984E-01	.7744	515.5	31.78	.4313E-01	.4704
.4012E-01	811.1	30.94	.5661E-01	.7401	509.7	29.64	.4980E-01	.4651
.4585F-01	782.6	28.82	.6322E-01	.7141	506.7	27.78	.5633E-01	.4623
.5159E-01	760.0	26.89	.6971E-01	.6935	502.7	26.21	.6277E-01	.4587
.5732E-01	743.6	24.95	.7608E-01	.6785	498.7	24.68	.6911E-01	.4550
.6305E-01	727.9	23.02	.8235E-01	.6642	493.4	23.16	.7538E-01	.4502
.6878E-01	716.3	21.09	.8854E-01	.6536	489.6	21.65	.8158E-01	.4468
.7451E-01	705.9	19.20	.9464E-01	.6441	485.1	20.15	.8772E-01	.4426
.8025E-01	694.6	17.36	.1007	.6338	482.3	18.68	.9380E-01	.4401
.8598E-01	687.5	15.55	.1067	.6273	479.9	17.23	.9982E-01	.4379
.9171E-01	680.8	13.75	.1126	.6212	476.3	15.79	.1058	.4346
.9744E-01	672.8	11.97	.1185	.6139	474.2	14.37	.1117	.4327
.1032	666.8	10.22	.1243	.6085	471.9	12.96	.1176	.4306
.1089	660.3	8.47	.1301	.6025	469.8	11.56	.1235	.4287
.1146	653.6	6.73	.1359	.5964	466.0	10.17	.1293	.4252
.1204	646.4	5.01	.1417	.5898	465.3	8.80	.1351	.4246
.1261	639.3	3.30	.1474	.5834	464.7	7.45	.1409	.4241
.1318	632.1	1.60	.1531	.5768	464.7	6.09	.1467	.4240
.1376	624.4	-.09	.1589	.5698	465.8	4.72	.1525	.4250
.1433	616.1	-1.76	.1646	.5622	468.3	3.36	.1582	.4273
.1490	606.4	-3.45	.1703	.5534	472.3	2.02	.1640	.4310
.1548	594.3	-5.17	.1761	.5423	477.9	.75	.1697	.4360
.1605	578.3	-6.92	.1819	.5277	485.3	-.48	.1754	.4428
.1662	557.6	-8.56	.1876	.5088	496.7	-1.89	.1812	.4532
.1720	531.8	-10.05	.1934	.4853	521.7	-3.55	.1869	.4760
.1777	499.1	-11.58	.1993	.4554	600.4	-4.87	.1926	.5478
.1834	0.	-90.00	.2054	0.	0.	90.00	.1984	0.

Figure A.2.- Continued.

SURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS
REDUCED WEIGHT FLOW

BLADE SURFACE 1

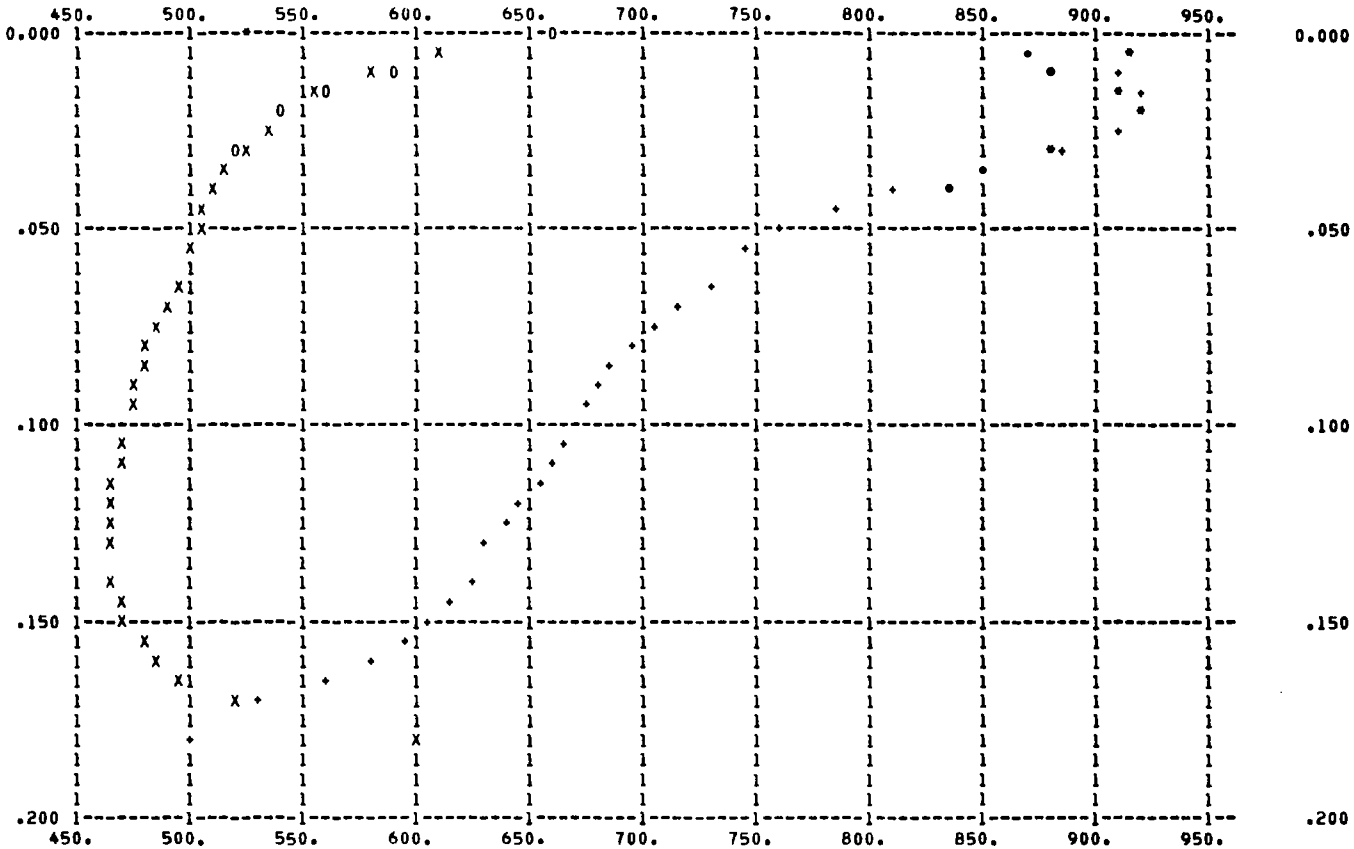
M	VELOCITY	ANGLE(DEG)	W/WCR
0.	522.7	90.00	.4769
.3555E-02	914.3	54.65	.8343
.7776E-02	880.0	51.28	.8030
.1256E-01	911.6	47.67	.8318
.1800E-01	919.8	43.93	.8393
.2424E-01	909.4	39.79	.8298
.3152E-01	879.6	35.29	.8026
.4012E-01	833.8	30.94	.7608
.5018E-01	777.0	27.36	.7090
.6197E-01	743.7	23.38	.6786
.7657E-01	708.7	18.53	.6466
.9699E-01	694.0	12.11	.6333
.1779	1096.	-11.65	1.000

BLADE SURFACE 2

M	VELOCITY	ANGLE(DEG)	W/WCR
.2310E-02	660.1	45.89	.6023
.7995E-02	592.4	43.18	.5406
.1426E-01	560.6	40.32	.5115
.2122E-01	538.8	37.30	.4916
.2901E-01	517.8	34.00	.4725
.3790E-01	509.2	30.44	.4647
.4811E-01	502.4	27.13	.4585
.5979E-01	492.8	24.03	.4496
.7348E-01	484.4	20.42	.4420
.9043E-01	474.9	16.11	.4334
.1143	465.4	10.25	.4247

Figure A.2.- Continued.

BLADE SURFACE VELOCITIES FOR FULL WEIGHT FLOW



VELOCITY(W) VS. MERIDIONAL STREAMLINE DISTANCE(M) DOWN THE PAGE

+ - BLADE SURFACE 1, BASED ON MERIDIONAL COMPONENT
• - BLADE SURFACE 1, BASED ON TANGENTIAL COMPONENT
X - BLADE SURFACE 2, BASED ON MERIDIONAL COMPONENT
0 - BLADE SURFACE 2, BASED ON TANGENTIAL COMPONENT

Figure A.2.- Continued.

BLADE SURFACE PRESSURES FOR FULL WEIGHT FLOW

IM	M	M/MC	P(1)/PT	P(2)/PT	CPT(1)	CPT(2)
24	0.	0.	.9994	.9994	-.6204E-03	-.6204E-03
25	.5732E-02	.3125E-01	.6760	.8301	-.3240	-.1699
26	.1146E-01	.6250E-01	.6507	.8435	-.3493	-.1565
27	.1720E-01	.9375E-01	.6432	.8543	-.3568	-.1457
28	.2293E-01	.1250	.6479	.8612	-.3521	-.1388
29	.2866E-01	.1563	.6631	.8652	-.3369	-.1348
30	.3439E-01	.1875	.6850	.8682	-.3150	-.1318
31	.4012E-01	.2188	.7074	.8697	-.2926	-.1303
32	.4585E-01	.2500	.7239	.8700	-.2761	-.1300
33	.5159E-01	.2813	.7365	.8707	-.2635	-.1293
34	.5732E-01	.3125	.7452	.8714	-.2548	-.1286
35	.6305E-01	.3438	.7533	.8725	-.2467	-.1275
36	.6878E-01	.3750	.7587	.8728	-.2413	-.1272
37	.7451E-01	.4063	.7631	.8731	-.2369	-.1269
38	.8025E-01	.4375	.7677	.8722	-.2323	-.1278
39	.8598E-01	.4688	.7696	.8710	-.2304	-.1290
40	.9171E-01	.5000	.7712	.8701	-.2288	-.1299
41	.9744E-01	.5313	.7736	.8689	-.2264	-.1311
42	.1032	.5625	.7755	.8684	-.2245	-.1316
43	.1089	.5938	.7782	.8683	-.2218	-.1317
44	.1146	.6250	.7813	.8693	-.2187	-.1307
45	.1204	.6563	.7846	.8691	-.2154	-.1309
46	.1261	.6875	.7877	.8687	-.2123	-.1313
47	.1318	.7188	.7906	.8677	-.2094	-.1323
48	.1376	.7500	.7933	.8660	-.2067	-.1340
49	.1433	.7813	.7961	.8635	-.2039	-.1365
50	.1490	.8125	.7995	.8602	-.2005	-.1398
51	.1548	.8438	.8039	.8563	-.1961	-.1437
52	.1605	.8750	.8101	.8516	-.1899	-.1484
53	.1662	.9063	.8184	.8453	-.1816	-.1547
54	.1720	.9375	.8289	.8334	-.1711	-.1666
55	.1777	.9688	.8423	.7961	-.1577	-.2039
56	.1834	1.000	.9521	.9521	-.4788E-01	-.4788E-01

X ARRAY

.968750	.937500	.906250	.875000	.843750
.812500	.781250	.750000	.718750	.687500
.656250	.625000	.593750	.562500	.531250
.500000	.468750	.437500	.406250	.375000
.343750	.312500	.281250	.250000	.218750
.187500	.156250	.125000	.093750	.062500
.031250	.031250	.062500	.093750	.125000
.156250	.187500	.218750	.250000	.281250
.312500	.343750	.375000	.406250	.437500
.468750	.500000	.531250	.562500	.593750
.625000	.656250	.687500	.718750	.750000
.781250	.812500	.843750	.875000	.906250
.937500	.968750			

Y ARRAY

600.354	521.669	496.727	485.325	477.873
472.312	468.339	465.802	464.706	464.733
465.289	466.016	469.834	471.882	474.222
476.253	479.869	482.348	485.093	489.643
493.390	498.688	502.725	506.666	509.703

Figure A.2.- Continued.

515.508	524.794	536.506	554.157	579.455
608.451	871.598	909.197	918.981	910.061
884.702	848.724	811.096	782.606	760.025
743.617	727.894	716.322	705.924	694.579
687.479	680.820	672.815	666.824	660.302
653.568	646.379	639.342	632.094	624.435
616.098	606.446	594.286	578.274	557.598
531.810	499.051			

```
OUTPUT FROM OBJCON

   OBJECTIVE FUNCTION   DIFFS     =  .91898E+03
                        WO        =  .49945E+03
                        DIFFS/WO  =  .18400E+01

   CONSTRAINT 1         BLTKS     =  .66765E+00

   CONSTRAINT 2         YMAX      =  .60845E+03
                        YMIN      =  .46471E+03
                        DIFFP     =  .13093E+01

   CONSTRAINT 3     WMB(MBO-2,1)  =  .53181E+03
                    WMB(MBO-2,2)  =  .52167E+03
                        TECLSR    =  .12677E+00
```

Figure A.2.- Continued.

```
 * * * * * * * * * * * * * * * * * * * * * * * * * * *
 *                                                     *
 *                   C O N M I N                       *
 *                                                     *
 *               FORTRAN PROGRAM FOR                   *
 *                                                     *
 *        CONSTRAINED FUNCTION MINIMIZATION            *
 *                                                     *
 * * * * * * * * * * * * * * * * * * * * * * * * * * *


CONSTRAINED FUNCTION MINIMIZATION

CONTROL PARAMETERS

IPRINT  NDV    ITMAX    NCON    NSIDE  ICNDIR   NSCAL   NFDG
  5      6       1        6       1       7      -2       0

LINOBJ  ITRM    N1       N2      N3      N4      N5
  0      3       8       18      20      20      40

    CT              CTMIN          CTL            CTLMIN
 -.10000E+00     .40000E-02    -.10000E-01     .10000E-02

   THETA            PHI           DELFUN         DABFUN
  .10000E+01     .50000E+01     .10000E-03     .18400E-02

   FDCH            FDCHM          ALPHAX         ABOBJ1
  .10000E-01     .10000E-02     .10000E+00     .50000E-01

LOWER BOUNDS ON DECISION VARIABLES (VLB)
  1)    -.15000E+01   .80000E+00   .44444E+00   .33333E+00   .60000E+00   .60000E+00

UPPER BOUNDS ON DECISION VARIABLES (VUB)
  1)    0.            .24000E+01   .12222E+01   .26667E+01   .20000E+01   .24000E+01

SCALING VECTOR (SCAL)
  .1000E+02   .2500E+00   .4500E+00   .1500E+01   .5000E-01   .5000E-02

ALL CONSTRAINTS ARE NON-LINEAR


INITIAL FUNCTION INFORMATION

OBJ =     .183998E+01

DECISION VARIABLES (X-VECTOR)
  1)    -.10000E+02   .25000E+00   .45000E+00   .15000E+01   .50000E-01   .50000E-02

CONSTRAINT VALUES (G-VECTOR)
  1)    -.66765E+01  -.93323E+00  -.13093E+02  -.18167E+00  -.11268E+01  -.87323E+00
```

Figure A.2.- Continued.

BEGIN ITERATION NUMBER 1

CT = -.10000E+00 CTL = -.10000E-01 PHI = .50000E+01

*** INPUT FOR BLADE ELEMENT PROGRAM ***

NO. BLADES	INLET RADIUS (L)	CONE ANGLE (DEG)	SOLIDITY	INLET BLADE ANGLE (DEG)	OUTLET BLADE ANGLE (DEG)	TRANS. LOC. /CHORD	MAX.TH. LOC. /CHORD	IN/OUT TURNING RATE	MAX. THICK. /CHORD	L.E. RAD. /CHORD	T.E. RAD. /CHORD
34	.45400	6.66400	2.25200	52.000	-9.9000	.25000	.45000	1.50000	.05000	.00500	.00500

Figure A.2.- Continued.

--- INPUT FOR TSONIC BLADE-TO-BLADE FLOW SOLVER ---

```
   GAM            AR              TIP             RHOIP           WTFL                                            OMEGA           ORF
 1.400000       1716.480        599.7600        .3345860E-02    .5700000E-02    0.                      0.                      1.910000
  BETAI          BETAO           CHORDF          STGRF           FSMI            FSMO
 48.20000       0.              .1833593        .1315224        0.              0.
  REDFAC         DENTOL          SSM1            SSM2
 1.000000       .1000000E-02    0.              0.

MBI    MBO                     MM     NBBI    NBL    NRSP   MOPT   LOPT   LRVB
 24     56      0       0      72      15      34     13      0      1      0

 BLADE SURFACE 1  --  UPPER SURFACE
   RI1            RO1             BETI1           BETO1           SPLNO1
 .9668292E-03   .9668292E-03    57.63155        -13.03485       13.00000
   MSP1  ARRAY
 .1502917E-03   .5978272E-02    .1529266E-01    .2743032E-01    .4445239E-01    .6270190E-01    .8183494E-01    .1016254
 .1218393       .1422379        .1585276        .1726546        .1826251
   THSP1  ARRAY
 .1139973E-02   .1957908E-01    .4336726E-01    .6727171E-01    .9180706E-01    .1116533        .1270053        .1377225
 .1437228       .1449823        .1425966        .1380689        .1334640

 BLADE SURFACE 2  --  LOWER SURFACE
   RI2            RO2             BETI2           BETO2           SPLNO2
 .9668292E-03   .9668292E-03    46.20589        -4.944609       13.00000
   MSP2  ARRAY
 .1664831E-02   .8668414E-02    .1911275E-01    .3188349E-01    .4891460E-01    .6674475E-01    .8519690E-01    .1041455
 .1234617       .1430141        .1587367        .1724944        .1822947
   THSP2  ARRAY
-.1473066E-02   .1370572E-01    .3337749E-01    .5346616E-01    .7486838E-01    .9285030E-01    .1075148        .1187910
 .1266333       .1310207        .1320445        .1311376        .1294676

   MR  ARRAY
-.1300000      -.5000000E-02    .5000000E-02    .2003000E-01    .4003000E-01    .8003000E-01    .1000300        .1250300
 .1500400       .1752600        .1852600        .2250000        .2700000
   RMSP  ARRAY
 .4540000       .4540000        .4540000        .4540000        .4540000        .4540000        .4540000        .4540000
 .4540000       .4540000        .4540000        .4540000        .4540000
   BESP  ARRAY
 .5000000E-01   .5000000E-01    .4990000E-01    .4960000E-01    .4925000E-01    .4850000E-01    .4815000E-01    .4770000E-01
 .4720000E-01   .4675000E-01    .4661000E-01    .4661000E-01    .4661000E-01
   PLOSS ARRAY
0.              0.              .1500000E-02    .5900000E-02    .1100000E-01    .2150000E-01    .3000000E-01    .3350000E-01
 .4000000E-01   .4700000E-01    .4800000E-01    .4800000E-01    .4800000E-01

 BLDAT  AANDK  ERSOR  STRFN  SLCRD  INTVL  SURVL
   0      0      0      0      0      0      1
```

Figure A.2.- Continued.

RELATIVE VELOCITY	MERIDIONAL VELOCITY	CRITICAL VELOCITY	REL. FLOW ANGLE
AT M = FSMI	AT M = FSMI	AT M = FSMI	AT UPSTREAM BDY.
745.90	497.17	1095.9	48.237
AT M = FSMO	AT M = FSMO	AT M = FSMO	AT DOWNSTREAM BDY.
499.44	499.44	1095.9	0.

FSMI = 0.
FSMO = .18336

CALCULATED PROGRAM CONSTANTS

PITCH	HT	HM1
.1847996	.1231997E-01	.5729979E-02

ITMIN	ITMAX
0	25

LAMBDA	DOWNSTREAM WHIRL (RVTHO)
252.4466	0.

REDUCED WEIGHT FLOW = .5700000E-02

NUMBER OF INTERIOR MESH POINTS = 1035

CALCULATED VELOCITY DIAGRAM INFORMATION

	IM	W	W/WCR	BETA
UPSTREAM BOUNDARY	1	745.47	.68022	48.237
LEADING EDGE	24	745.90	.68061	48.200
TRAILING EDGE	56	499.44	.45572	0.
DOWNSTREAM BOUNDARY	72	500.06	.45629	0.

Figure A.2.- Continued.

```
    ITERATION  NO.    1
    MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .3084        AT IM =  0,    IT = 11,    SURF = 1,    M =  .1786
    AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1106
    NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  86


    ITERATION  NO.    2
    MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .2718        AT IM =  0,    IT = 11,    SURF = 1,    M =  .1786
    AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .2915E-01
    NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  86


    ITERATION  NO.    3
    MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .4543        AT IM =  0,    IT = 11,    SURF = 1,    M =  .1786
    AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1308E-01
    NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  85


    ITERATION  NO. · 4
    MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  3.728        AT IM =  0,    IT = 11,    SURF = 1,    M =  .1786
    AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .4423E-01
    NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  53
DENSTY CALL NO.  9
NER(1) =  1
RHO*W IS  1.2462 TIMES THE MAXIMUM VALUE FOR RHO*W


    ITERATION  NO.    5
    MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .8621        AT IM =  0,    IT = 11,    SURF = 1,    M =  .1786
    AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1063E-01
    NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  23


    ITERATION  NO.    6
    MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .3878        AT IM =  0,    IT = 11,    SURF = 1,    M =  .1786
    AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .4736E-02
    NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  13


    ITERATION  NO.    7
    MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  1.740        AT IM =  0,    IT = 11,    SURF = 1,    M =  .1786
    AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1951E-01
    NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =   8
DENSTY CALL NO.  9
NER(1) =  2
RHO*W IS  1.2487 TIMES THE MAXIMUM VALUE FOR RHO*W


    ITERATION  NO.    8
    MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .7370        AT IM =  0,    IT = 11,    SURF = 1,    M =  .1786
    AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .8272E-02
```

Figure A.2.- Continued.

```
    NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =    1


    ITERATION  NO.   9
    MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .3891       AT IM =  0,   IT = 11,   SURF = 1,   M =  .1786
    AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .4369E-02
    NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =    1


    ITERATION  NO.  10
    MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  1.757       AT IM =  0,   IT = 11,   SURF = 1,   M =  .1786
    AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1955E-01
    NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =    1
DENSTY CALL NO.  9
NER(1) =  3
RHO*W IS  1.2486 TIMES THE MAXIMUM VALUE FOR RHO*W


    ITERATION  NO.  11
    MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .7389       AT IM =  0,   IT = 11,   SURF = 1,   M =  .1786
    AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .8235E-02
    NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =    1
```

Figure A.2.- Continued.

SURFACE VELOCITIES BASED ON MERIDIONAL COMPONENTS - REDUCED WEIGHT FLOW

M	BLADE SURFACE 1 VELOCITY	ANGLE(DEG)	SURF. LENGTH	W/WCR	BLADE SURFACE 2 VELOCITY	ANGLE(DEG)	SURF. LENGTH	W/WCR
0.	0.	90.00	0.	0.	0.	-90.00	0.	0.
.5730E-02	871.3	52.88	.1030E-01	.7950	608.9	44.25	.6677E-02	.5556
.1146E-01	909.0	48.49	.1935E-01	.8294	579.9	41.59	.1450E-01	.5292
.1719E-01	918.8	44.49	.2767E-01	.8384	554.6	39.03	.2202E-01	.5061
.2292E-01	910.0	40.66	.3545E-01	.8304	537.0	36.59	.2927E-01	.4900
.2865E-01	884.7	37.02	.4281E-01	.8073	525.2	34.17	.3630E-01	.4793
.3438E-01	849.2	33.73	.4983E-01	.7749	516.0	31.81	.4313E-01	.4708
.4011E-01	811.3	30.97	.5661E-01	.7402	510.1	29.67	.4979E-01	.4655
.4584E-01	782.8	28.85	.6322E-01	.7143	507.1	27.82	.5633E-01	.4627
.5157E-01	760.2	26.93	.6970E-01	.6937	503.1	26.25	.6276E-01	.4591
.5730E-01	743.9	25.00	.7608E-01	.6787	499.1	24.73	.6911E-01	.4554
.6303E-01	728.1	23.06	.8235E-01	.6644	493.7	23.21	.7538E-01	.4505
.6876E-01	716.5	21.14	.8853E-01	.6538	490.0	21.70	.8158E-01	.4471
.7449E-01	706.1	19.25	.9464E-01	.6443	485.4	20.20	.8771E-01	.4429
.8022E-01	694.8	17.40	.1007	.6339	482.6	18.73	.9379E-01	.4404
.8595E-01	687.6	15.60	.1067	.6274	480.2	17.28	.9982E-01	.4382
.9168E-01	681.0	13.81	.1126	.6214	476.6	15.85	.1058	.4349
.9741E-01	672.9	12.03	.1185	.6140	474.4	14.43	.1117	.4329
.1031	666.9	10.27	.1243	.6085	472.2	13.02	.1176	.4309
.1089	660.5	8.53	.1301	.6027	470.0	11.62	.1235	.4288
.1146	653.6	6.80	.1359	.5964	467.2	10.24	.1293	.4263
.1203	646.5	5.07	.1416	.5899	465.4	8.87	.1351	.4247
.1261	639.3	3.37	.1474	.5834	464.7	7.52	.1409	.4240
.1318	632.2	1.67	.1531	.5769	464.8	6.16	.1467	.4241
.1375	624.6	-.01	.1589	.5699	465.9	4.80	.1525	.4252
.1432	616.2	-1.68	.1646	.5623	468.3	3.43	.1582	.4273
.1490	606.5	-3.37	.1703	.5534	472.3	2.10	.1639	.4310
.1547	594.4	-5.09	.1761	.5424	477.9	.83	.1697	.4361
.1604	578.6	-6.83	.1818	.5280	485.2	-.40	.1754	.4427
.1662	558.0	-8.47	.1876	.5091	496.4	-1.80	.1811	.4530
.1719	532.3	-9.97	.1934	.4857	521.3	-3.46	.1869	.4757
.1776	499.8	-11.49	.1992	.4561	598.6	-4.77	.1926	.5462
.1834	0.	-90.00	.2053	0.	0.	90.00	.1984	0.

Figure A.2.- Continued.

SURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS
REDUCED WEIGHT FLOW

BLADE SURFACE 1

M	VELOCITY	ANGLE(DEG)	W/WCR
0.	522.3	90.00	.4766
.3554E-02	913.9	54.66	.8339
.7774E-02	879.7	51.29	.8027
.1255E-01	911.4	47.68	.8316
.1799E-01	919.7	43.95	.8392
.2423E-01	909.4	39.81	.8298
.3150E-01	879.6	35.33	.8027
.4009E-01	833.0	30.98	.7601
.5013E-01	777.3	27.41	.7092
.6189E-01	744.0	23.44	.6788
.7644E-01	708.7	18.62	.6467
.9671E-01	693.5	12.25	.6328
.1786	1096.	-11.78	1.000

BLADE SURFACE 2

M	VELOCITY	ANGLE(DEG)	W/WCR
.2310E-02	660.6	45.89	.6028
.7994E-02	592.9	43.18	.5410
.1426E-01	561.1	40.33	.5120
.2121E-01	539.3	37.31	.4921
.2900E-01	518.3	34.02	.4729
.3788E-01	509.7	30.47	.4651
.4807E-01	502.8	27.18	.4588
.5973E-01	493.1	24.08	.4499
.7337E-01	484.7	20.49	.4423
.9024E-01	475.2	16.21	.4336
.1139	466.1	10.41	.4253

Figure A.2.- Continued.

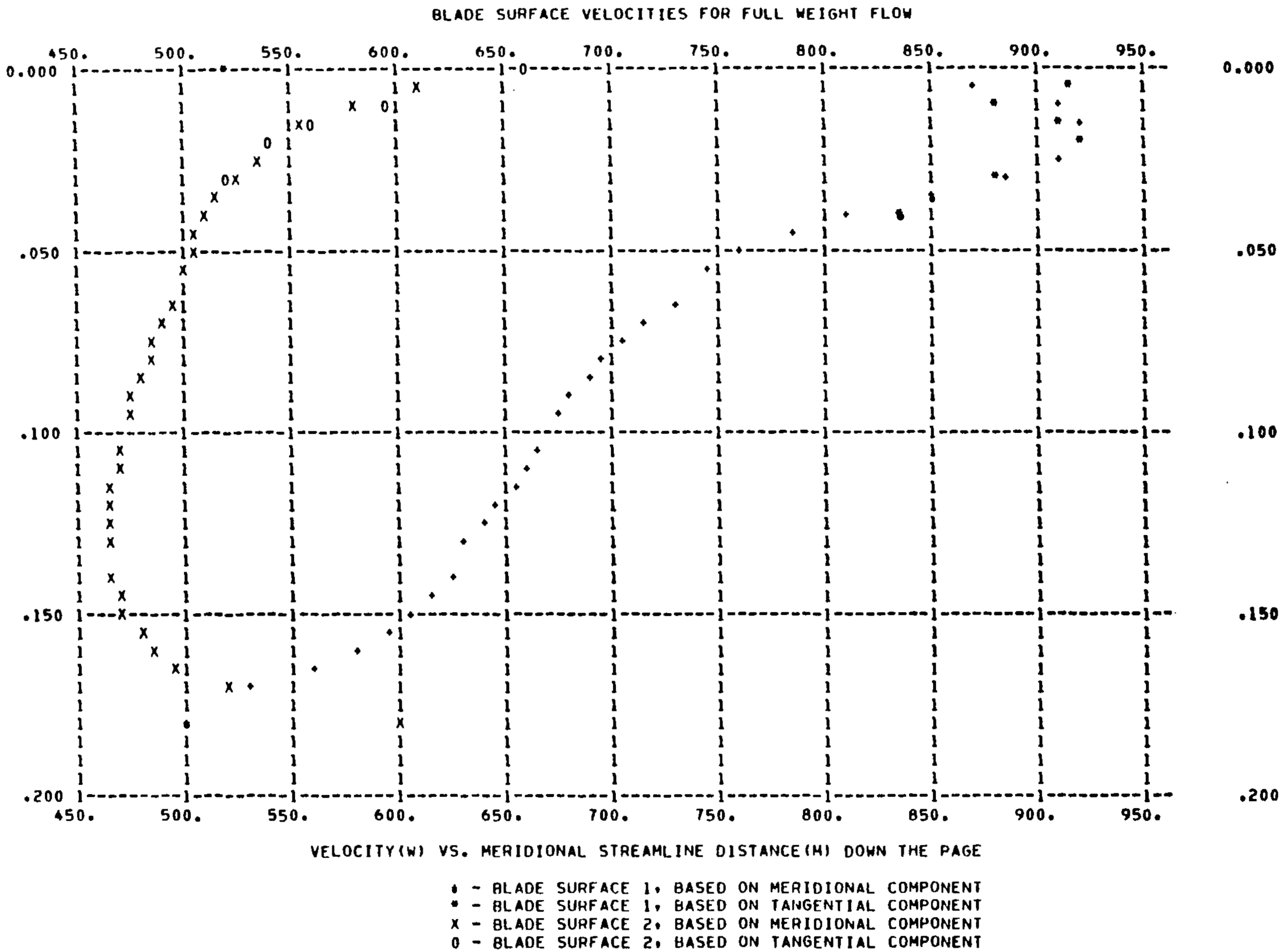


Figure A.2.- Continued.

BLADE SURFACE PRESSURES FOR FULL WEIGHT FLOW

IM	M	M/MC	P(1)/PT	P(2)/PT	CPT(1)	CPT(2)
24	0.	0.	.9994	.9994	-.6204E-03	-.6204E-03
25	.5730E-02	.3125E-01	.6762	.8298	-.3238	-.1702
26	.1146E-01	.6250E-01	.6509	.8432	-.3491	-.1568
27	.1719E-01	.9375E-01	.6433	.8541	-.3567	-.1459
28	.2292E-01	.1250	.6479	.8610	-.3521	-.1390
29	.2865E-01	.1563	.6631	.8650	-.3369	-.1350
30	.3438E-01	.1875	.6847	.8680	-.3153	-.1320
31	.4011E-01	.2188	.7073	.8695	-.2927	-.1305
32	.4584E-01	.2500	.7238	.8699	-.2762	-.1301
33	.5157E-01	.2813	.7363	.8706	-.2637	-.1294
34	.5730E-01	.3125	.7450	.8713	-.2550	-.1287
35	.6303E-01	.3438	.7531	.8724	-.2469	-.1276
36	.6876E-01	.3750	.7585	.8726	-.2415	-.1274
37	.7449E-01	.4063	.7630	.8729	-.2370	-.1271
38	.8022E-01	.4375	.7676	.8721	-.2324	-.1279
39	.8595E-01	.4688	.7695	.8708	-.2305	-.1292
40	.9168E-01	.5000	.7711	.8700	-.2289	-.1300
41	.9741E-01	.5313	.7736	.8688	-.2264	-.1312
42	.1031	.5625	.7755	.8682	-.2245	-.1318
43	.1089	.5938	.7781	.8682	-.2219	-.1318
44	.1146	.6250	.7813	.8688	-.2187	-.1312
45	.1203	.6563	.7845	.8691	-.2155	-.1309
46	.1261	.6875	.7877	.8687	-.2123	-.1313
47	.1318	.7188	.7905	.8677	-.2095	-.1323
48	.1375	.7500	.7932	.8660	-.2068	-.1340
49	.1432	.7813	.7961	.8635	-.2039	-.1365
50	.1490	.8125	.7995	.8603	-.2005	-.1397
51	.1547	.8438	.8038	.8563	-.1962	-.1437
52	.1604	.8750	.8099	.8516	-.1901	-.1484
53	.1662	.9062	.8182	.8455	-.1818	-.1545
54	.1719	.9375	.8287	.8336	-.1713	-.1664
55	.1776	.9688	.8420	.7969	-.1580	-.2031
56	.1834	1.000	.9521	.9521	-.4788E-01	-.4788E-01

X ARRAY

.968750	.937500	.906250	.875000	.843750
.812500	.781250	.750000	.718750	.687500
.656250	.625000	.593750	.562500	.531250
.500000	.468750	.437500	.406250	.375000
.343750	.312500	.281250	.250000	.218750
.187500	.156250	.125000	.093750	.062500
.031250	.031250	.062500	.093750	.125000
.156250	.187500	.218750	.250000	.281250
.312500	.343750	.375000	.406250	.437500
.468750	.500000	.531250	.562500	.593750
.625000	.656250	.687500	.718750	.750000
.781250	.812500	.843750	.875000	.906250
.937500	.968750			

Y ARRAY

598.622	521.326	496.430	485.191	477.892
472.296	468.296	465.933	464.826	464.715
465.422	467.225	469.987	472.202	474.379
476.622	480.225	482.606	485.388	489.992
493.694	499.052	503.092	507.060	510.120

Figure A.2.- Continued.

515.956	525.246	536.981	554.624	579.933
608.917	871.309	908.986	918.846	910.016
884.741	849.202	811.259	782.794	760.231
743.855	728.095	716.532	706.124	694.759
687.633	680.982	672.940	666.912	660.502
653.560	646.538	639.326	632.201	624.598
616.196	606.488	594.393	578.629	557.971
532.289	499.834			

```
OUTPUT FROM OBJCON

   OBJECTIVE FUNCTION   DIFFS      =  .91885E+03
                        WO         =  .49944E+03
                        DIFFS/WO   =  .18398E+01

   CONSTRAINT 1         BLTKS      =  .66766E+00

   CONSTRAINT 2         YMAX       =  .60892E+03
                        YMIN       =  .46472E+03
                        DIFFP      =  .13103E+01

   CONSTRAINT 3     WMB(MBO-2,1)   =  .53229E+03
                    WMB(MBO-2,2)   =  .52133E+03
                        TECLSR     =  .13704E+00
```

Figure A.2.- Continued.

```
FINAL OPTIMIZATION INFORMATION

 OBJ =     .170765E+01

 DECISION VARIABLES (X-VECTOR)
   1)    -.93094E+01   .29650E+00   .55000E+00   .95806E+00   .32891E-01   .51525E-02

 CONSTRAINT VALUES (G-VECTOR)
   1)    -.28002E+01  -.97200E+00  -.14455E+02  -.96575E-01  -.15965E+01  -.40346E+00

 THERE ARE    1 ACTIVE CONSTRAINTS
 CONSTRAINT NUMBERS ARE
     4

 THERE ARE    0 VIOLATED CONSTRAINTS

 THERE ARE    1 ACTIVE SIDE CONSTRAINTS
 DECISION VARIABLES AT LOWER OR UPPER BOUNDS (MINUS INDICATES LOWER BOUND)
     3

 TERMINATION CRITERION
      ITER EQUALS ITMAX

 NUMBER OF ITERATIONS =    1

 OBJECTIVE FUNCTION WAS EVALUATED          12  TIMES

 CONSTRAINT FUNCTIONS WERE EVALUATED       12  TIMES
```

Figure A.2.- Continued.

OPTIMIZATION RESULTS

OBJECTIVE FUNCTION
GLOBAL LOCATION 1 FUNCTION VALUE .17077E+01

DESIGN VARIABLES

ID	D. V. NO.	GLOBAL VAR. NO.	LOWER BOUND	VALUE	UPPER BOUND
1	1	2	-.15000E+02	-.93094E+01	0.
2	2	3	.20000E+00	.29650E+00	.60000E+00
3	3	4	.20000E+00	.55000E+00	.55000E+00
4	4	5	.50000E+00	.95806E+00	.40000E+01
5	5	6	.30000E-01	.32891E-01	.10000E+00
6	6	7	.30000E-02	.51525E-02	.12000E-01

DESIGN CONSTRAINTS

ID	GLOBAL VAR. NO.	LOWER BOUND	VALUE	UPPER BOUND
1	8	0.	.28002E+00	.10000E+02
3	9	0.	.14455E+01	.16000E+01
5	10	-.10000E+01	.59654E+00	.10000E+01

Figure A.2.- Continued.

*** INPUT FOR BLADE ELEMENT PROGRAM ***

NO. BLADES	INLET RADIUS (L)	CONE ANGLE (DEG)	SOLIDITY	INLET BLADE ANGLE (DEG)	OUTLET BLADE ANGLE (DEG)	TRANS. LOC. /CHORD	MAX.TH. LOC. /CHORD	IN/OUT TURNING RATE	MAX. THICK. /CHORD	L.E. RAD. /CHORD	T.E. RAD. /CHORD
34	.45400	6.66400	2.25200	52.000	-9.3094	.29650	.55000	.95806	.03289	.00515	.00500

Figure A.2.- Continued.

```
--- INPUT FOR TSONIC BLADE-TO-BLADE FLOW SOLVER ---

  GAM          AR           TIP          RHOIP         WTFL                        OMEGA        ORF
1.400000     1716.480     599.7600     .3345860E-02  .5700000E-02  0.           0.           1.910000
 BETAI        BETAO        CHORDF        STGRF         FSMI         FSMO
48.20000     0.           .1795132     .1534563      0.           0.
 REDFAC        DENTOL        SSM1          SSM2
1.000000     .1000000E-02  0.           0.

MBI   MBO                 MM   NBBI   NBL   NRSP   MOPT   LOPT   LRVB
 24    56     0     0     72    15     34    13      0      1      0

BLADE SURFACE 1  --  UPPER SURFACE
  RI1          RO1          BETI1         BETO1         SPLNO1
.9958413E-03  .9663591E-03  53.17187     -11.06885     13.00000
  MSP1  ARRAY
.1988241E-03  .6491482E-02  .1596615E-01  .2768746E-01  .4356364E-01  .6073413E-01  .7906837E-01  .9830693E-01
.1181776      .1383991      .1546368      .1687631      .1787495
  THSP1  ARRAY
.1314862E-02  .1885167E-01  .4212224E-01  .6675356E-01  .9436997E-01  .1178990      .1367454      .1506814
.1595550      .1632900      .1625744      .1592733      .1554091

BLADE SURFACE 2  --  LOWER SURFACE
  RI2          RO2          BETI2         BETO2         SPLNO2
.9958413E-03  .9663591E-03  50.81841     -5.477123     13.00000
  MSP2  ARRAY
.1767866E-02  .8225985E-02  .1791505E-01  .2985114E-01  .4593955E-01  .6312073E-01  .8117985E-01  .9994522E-01
.1192385      .1388764      .1547094      .1685705      .1784377
  THSP2  ARRAY
-.1385046E-02 .1513937E-01  .3691847E-01  .5973983E-01  .8493909E-01  .1063483      .1239432      .1375884
.1471895      .1526932      .1541361      .1532488      .1513964

  MR  ARRAY
-.1300000     -.5000000E-02 .5000000E-02  .2003000E-01  .4003000E-01  .8003000E-01  .1000300      .1250300
.1500400      .1752600      .1852600      .2250000      .2700000
  RMSP  ARRAY
.4540000      .4540000      .4540000      .4540000      .4540000      .4540000      .4540000      .4540000
.4540000      .4540000      .4540000      .4540000      .4540000
  BESP  ARRAY
.5000000E-01  .5000000E-01  .4990000E-01  .4960000E-01  .4925000E-01  .4850000E-01  .4815000E-01  .4770000E-01
.4720000E-01  .4675000E-01  .4661000E-01  .4661000E-01  .4661000E-01
  PLOSS ARRAY
0.            0.            .1500000E-02  .5900000E-02  .1100000E-01  .2150000E-01  .3000000E-01  .3350000E-01
.4000000E-01  .4700000E-01  .4800000E-01  .4800000E-01  .4800000E-01

BLDAT  AANDK  ERSOR  STRFN  SLCRD  INTVL  SURVL
  0      0      0      0      0      0      1
```

Figure A.2.- Continued.

RELATIVE VELOCITY	MERIDIONAL VELOCITY	CRITICAL VELOCITY	REL. FLOW ANGLE
AT M = FSMI	AT M = FSMI	AT M = FSMI	AT UPSTREAM BDY.
745.90	497.17	1095.9	48.257
AT M = FSMO	AT M = FSMO	AT M = FSMO	AT DOWNSTREAM BDY.
498.55	498.55	1095.9	0.

FSMI = 0.
FSMO = .17951

CALCULATED PROGRAM CONSTANTS

PITCH	HT	HM1
.1847996	.1231997E-01	.5609788E-02

ITMIN	ITMAX
0	27

LAMBDA	DOWNSTREAM WHIRL (RVTHO)
252.4466	0.

REDUCED WEIGHT FLOW = .5700000E-02

NUMBER OF INTERIOR MESH POINTS = 1046

CALCULATED VELOCITY DIAGRAM INFORMATION

	IM	W	W/WCR	BETA
UPSTREAM BOUNDARY	1	745.23	.68000	48.257
LEADING EDGE	24	745.90	.68061	48.200
TRAILING EDGE	56	498.55	.45491	0.
DOWNSTREAM BOUNDARY	72	499.77	.45602	0.

Figure A.2.- Continued.

```
ITERATION  NO.   1
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1837        AT IM =  0,   IT =  4,   SURF = 1,   M =  .1918E-01
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1045
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  90

ITERATION  NO.   2
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .7501E-01   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4069E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .2483E-01
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS.=  90

ITERATION  NO.   3
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .3724E-01   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4069E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .7438E-02
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  89

ITERATION  NO.   4
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1933E-01   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4069E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .2531E-02
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  49

ITERATION  NO.   5
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1019E-01   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4069E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .9376E-03
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  22

ITERATION  NO.   6
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .5275E-02   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4069E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .3679E-03
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  14

ITERATION  NO.   7
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .3104E-02   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4069E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1662E-03
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =   1

ITERATION  NO.   8
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1306E-02   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4069E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .5430E-04
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =   1

ITERATION  NO.   9
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .6819E-03   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4069E-02
```

Figure A.2.- Continued.

```
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .2643E-04
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =    0


ITERATION  NO.  10
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .3138E-03   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4069E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .2102E-04
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =    0
```


Figure A.2.- Continued.

SURFACE VELOCITIES BASED ON MERIDIONAL COMPONENTS - REDUCED WEIGHT FLOW

	BLADE SURFACE 1				BLADE SURFACE 2			
M	VELOCITY	ANGLE(DEG)	SURF. LENGTH	W/WCR	VELOCITY	ANGLE(DEG)	SURF. LENGTH	W/WCR
0.	0.	90.00	0.	0.	0.	-90.00	0.	0.
.5610E-02	836.7	50.59	.9360E-02	.7635	632.9	48.96	.6851E-02	.5775
.1122E-01	848.1	48.08	.1797E-01	.7739	599.0	46.37	.1518E-01	.5466
.1683E-01	851.3	45.66	.2618E-01	.7768	564.0	43.89	.2313E-01	.5147
.2244E-01	846.2	43.35	.3404E-01	.7721	540.8	41.52	.3077E-01	.4935
.2805E-01	834.5	41.16	.4162E-01	.7615	522.7	39.21	.3813E-01	.4769
.3366E-01	820.3	39.02	.4895E-01	.7485	508.1	36.98	.4526E-01	.4636
.3927E-01	805.2	36.85	.5607E-01	.7347	496.1	34.83	.5218E-01	.4527
.4488E-01	788.7	34.68	.6298E-01	.7197	487.9	32.77	.5893E-01	.4452
.5049E-01	770.7	32.50	.6972E-01	.7032	479.5	30.82	.6553E-01	.4375
.5610E-01	754.2	30.30	.7629E-01	.6882	473.5	28.92	.7200E-01	.4320
.6171E-01	739.3	28.11	.8272E-01	.6745	467.1	27.10	.7835E-01	.4263
.6732E-01	723.5	25.94	.8901E-01	.6602	462.5	25.34	.8461E-01	.4220
.7293E-01	710.8	23.83	.9520E-01	.6486	457.2	23.59	.9077E-01	.4172
.7854E-01	697.6	21.78	.1013	.6365	453.8	21.87	.9685E-01	.4141
.8415E-01	688.8	19.78	.1073	.6285	449.7	20.17	.1029	.4104
.8976E-01	680.9	17.80	.1132	.6213	447.5	18.48	.1088	.4083
.9537E-01	671.6	15.84	.1191	.6128	445.6	16.82	.1147	.4066
.1010	665.3	13.91	.1249	.6070	442.6	15.19	.1205	.4039
.1066	658.7	12.00	.1306	.6011	441.0	13.57	.1263	.4024
.1122	652.0	10.10	.1363	.5949	439.8	11.97	.1321	.4013
.1178	645.0	8.23	.1420	.5885	439.2	10.39	.1378	.4007
.1234	635.7	6.38	.1477	.5801	437.8	8.82	.1435	.3995
.1290	629.3	4.54	.1533	.5742	438.7	7.25	.1491	.4003
.1346	622.6	2.71	.1589	.5681	440.7	5.69	.1548	.4021
.1402	615.5	.90	.1646	.5616	444.0	4.12	.1604	.4052
.1459	607.4	-.92	.1702	.5542	449.0	2.60	.1660	.4097
.1515	597.2	-2.76	.1758	.5450	455.5	1.13	.1717	.4156
.1571	584.1	-4.61	.1814	.5330	463.5	-.29	.1773	.4230
.1627	567.1	-6.38	.1870	.5175	475.4	-1.92	.1829	.4338
.1683	548.2	-8.06	.1927	.5003	500.5	-3.80	.1885	.4567
.1739	533.5	-9.68	.1984	.4868	585.9	-5.24	.1941	.5347
.1795	0.	-90.00	.2043	0.	0.	90.00	.1998	0.

Figure A.2.- Continued.

SURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS
REDUCED WEIGHT FLOW

BLADE. SURFACE 1

M	VELOCITY	ANGLE(DEG)	W/WCR
0.	53A.5	90.00	.4913
.4069E-02	923.1	51.31	.8423
.8721E-02	838.6	49.19	.7652
.1374E-01	850.8	46.98	.7763
.1918E-01	851.5	44.69	.7770
.2507E-01	844.3	42.31	.7704
.3149E-01	831.4	39.85	.7587
.3852E-01	815.4	37.15	.7440
.4631E-01	787.7	34.13	.7188
.5512E-01	768.0	30.68	.7008
.6534E-01	738.9	26.70	.6742
.7768E-01	713.0	22.08	.6506
.9369E-01	687.7	16.42	.6275
.1202	656.6	7.45	.5992
.1658	603.2	-7.33	.5504

BLADE SURFACE 2

M	VELOCITY	ANGLE(DEG)	W/WCR
.2283E-02	729.0	50.57	.6652
.7074E-02	624.6	48.27	.5700
.1227E-01	581.7	45.90	.5308
.1793E-01	552.8	43.42	.5044
.2412E-01	530.1	40.82	.4837
.3092E-01	511.6	38.06	.4669
.3845E-01	496.7	35.13	.4532
.4688E-01	481.2	32.07	.4390
.5639E-01	467.9	28.82	.4270
.6734E-01	456.7	25.33	.4167
.8031E-01	447.8	21.33	.4086
.9666E-01	438.6	16.44	.4003
.1210	432.7	9.51	.3948

Figure A.2.- Continued.

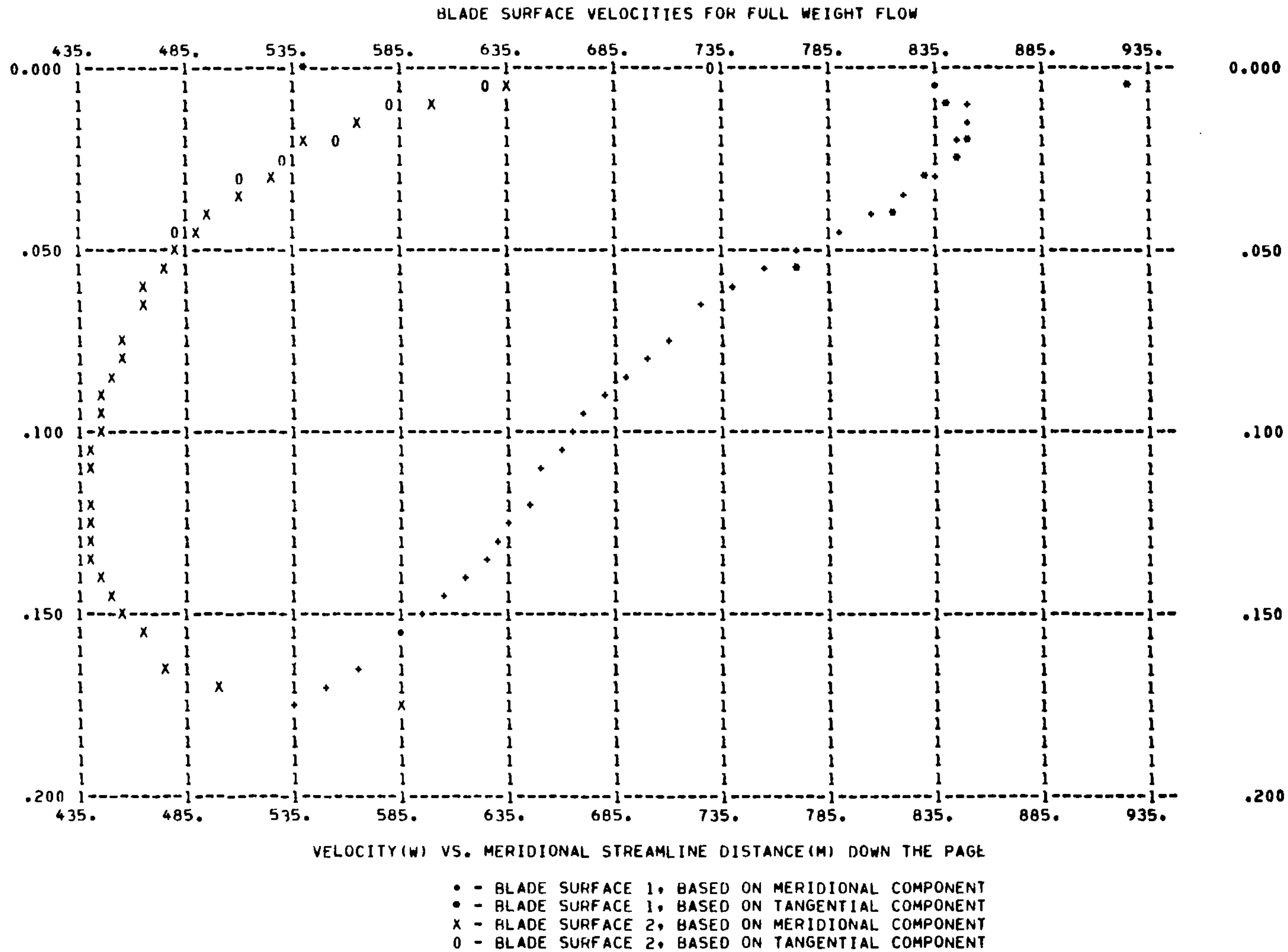


Figure A.2.- Continued.

BLADE SURFACE PRESSURES FOR FULL WEIGHT FLOW

IM	M	M/MC	P(1)/PT	P(2)/PT	CPT(1)	CPT(2)
24	0.	0.	.9994	.9994	-.6204E-03	-.6204E-03
25	.5610F-02	.3125E-01	.6981	.8173	-.3019	-.1827
26	.1122F-01	.6250E-01	.6899	.8337	-.3101	-.1663
27	.1683F-01	.9375E-01	.6867	.8496	-.3133	-.1504
28	.2244E-01	.1250	.6887	.8593	-.3113	-.1407
29	.2805E-01	.1563	.6949	.8663	-.3051	-.1337
30	.3366E-01	.1875	.7028	.8717	-.2972	-.1283
31	.3927E-01	.2188	.7111	.8759	-.2889	-.1241
32	.4488E-01	.2500	.7203	.8784	-.2797	-.1216
33	.5049E-01	.2813	.7303	.8810	-.2697	-.1190
34	.5610E-01	.3125	.7392	.8825	-.2608	-.1175
35	.6171E-01	.3438	.7469	.8839	-.2531	-.1161
36	.6732E-01	.3750	.7549	.8845	-.2451	-.1155
37	.7293F-01	.4063	.7608	.8851	-.2392	-.1149
38	.7854E-01	.4375	.7666	.8846	-.2334	-.1154
39	.8415E-01	.4688	.7695	.8840	-.2305	-.1160
40	.8976E-01	.5000	.7718	.8826	-.2282	-.1174
41	.9537E-01	.5313	.7749	.8811	-.2251	-.1189
42	.1010	.5625	.7768	.8806	-.2232	-.1194
43	.1066	.5938	.7793	.8801	-.2207	-.1199
44	.1122	.6250	.7823	.8799	-.2177	-.1201
45	.1178	.6563	.7855	.8797	-.2145	-.1203
46	.1234	.6875	.7899	.8797	-.2101	-.1203
47	.1290	.7188	.7925	.8785	-.2075	-.1215
48	.1346	.7500	.7949	.8766	-.2051	-.1234
49	.1402	.7813	.7972	.8740	-.2028	-.1260
50	.1459	.8125	.7999	.8705	-.2001	-.1295
51	.1515	.8437	.8033	.8663	-.1967	-.1337
52	.1571	.8750	.8082	.8614	-.1918	-.1386
53	.1627	.9062	.8148	.8551	-.1852	-.1449
54	.1683	.9375	.8222	.8433	-.1778	-.1567
55	.1739	.9687	.8278	.8035	-.1722	-.1965
56	.1795	1.000	.9525	.9525	-.4755E-01	-.4755E-01

Figure A.2.- Continued.

RESULTS OF COMPUTATIONS ON BASE SOLUTION:

.....MACH NUMBER,
VALUES OF PERTURBATION PARAMETERS,
CRITICAL VALUE OF WM:

MO = .1000

Q0(1) =	-9.3094	(KOCR)
Q0(2) =	.2965	(T)
Q0(3) =	.5500	(ZM)
Q0(4) =	.9581	(P)
Q0(5) =	.0329	(TMX)
Q0(6) =	.0052	(THLE)

WMCRIT =-66.8587

.....LOCATIONS OF MIN., MAX., AND CRITICAL PTS.
(* DENOTES POINT ON LOWER SURFACE)

MINIMUM AT X = .6875* (POINT NO. 10)
MAXIMUM AT X = .0937 (POINT NO. 34)

.....LOCATION OF FIXED POINTS
(* DENOTES POINT ON LOWER SURFACE)

XFIX(1) = 1.0000*
XFIX(2) = .0937
XFIX(3) = 1.0000

X ARRAY

.968750	.937500	.906250	.875000	.843750
.812500	.781250	.750000	.718750	.687500
.656250	.625000	.593750	.562500	.531250
.500000	.468750	.437500	.406250	.375000
.343750	.312500	.281250	.250000	.218750
.187500	.156250	.125000	.093750	.062500
.031250	.031250	.062500	.093750	.125000
.156250	.187500	.218750	.250000	.281250
.312500	.343750	.375000	.406250	.437500
.468750	.500000	.531250	.562500	.593750
.625000	.656250	.687500	.718750	.750000
.781250	.812500	.843750	.875000	.906250
.937500	.968750			

Y ARRAY

585.946	500.526	475.384	463.538	455.453
448.954	444.034	440.692	438.692	437.817

Figure A.2.- Continued.

439.170	439.833	441.032	442.598	445.557
447.463	449.745	453.791	457.230	462.531
467.147	473.460	479.521	487.914	496.105
508.106	522.699	540.850	564.034	599.017
632.856	836.747	848.134	851.345	846.151
834.503	820.308	805.229	788.747	770.703
754.190	739.256	723.486	710.818	697.563
688.798	680.850	671.634	665.267	658.725
651.996	644.996	635.734	629.278	622.630
615.503	607.355	597.245	584.097	567.118
548.249	533.506			

```
OUTPUT FROM OBJCON

   OBJECTIVE FUNCTION   DIFFS     =  .85135E+03
                        WO        =  .49855E+03
                        DIFFS/WO  =  .17077E+01

   CONSTRAINT 1         BLTKS     =  .28002E+00

   CONSTRAINT 2         YMAX      =  .63286E+03
                        YMIN      =  .43782E+03
                        DIFFP     =  .14455E+01

   CONSTRAINT 3     WMB(MBO-2,1)  =  .54825E+03
                    WMB(MBO-2,2)  =  .50053E+03
                        TECLSR    =  .59654E+00
```

Figure A.2.- Continued.

UNIT PERTURBATION OF WM AND UNIT STRAINING OF XBASE
FOR CALIBRATION SOLUTIONS 1 THROUGH 5

		• 1ST CALB SOLN •		• 2ND CALB SOLN •		• 3RD CALB SOLN •		• 4TH CALB SOLN •		• 5TH CALB SOLN •	
POINT	XBASE	XSTRUNIT	WMUNIT	XSTRUNIT	WMUNIT	XSTRUNIT	WMUNIT	XSTRUNIT	WMUNIT	XSTRUNIT	WMUNIT
1	.9687	.9687	-20.0835	.9687	-12.3557	.9687	.7336	.9679	49.8051	.9687	-134.3581
2	.9375	.9375	-2.0339	.9375	.3504	.9375	22.0747	.9357	31.4676	.9375	321.5647
3	.9062	.9062	-1.5704	.9062	-1.8310	.9062	32.2313	.9036	32.5262	.9062	553.3391
4	.8750	.8750	-1.1541	.8750	-2.8760	.8750	39.4171	.8714	33.1910	.8750	730.8362
5	.8437	.8437	-.5654	.8437	-3.2252	.8437	44.4573	.8393	33.0680	.8437	869.5636
6	.8125	.8125	-.0083	.8125	-3.2979	.8125	47.7861	.8071	32.3865	.8125	982.7536
7	.7813	.7813	.4519	.7813	-3.2944	.7813	49.9671	.7750	31.2294	.7813	1083.0267
8	.7500	.7500	.8409	.7500	-3.2652	.7500	51.3220	.7429	29.7367	.7500	1177.6278
9	.7188	.7188	1.1920	.7188	-3.1984	.7188	51.9009	.7107	27.8109	.7188	1266.5219
10	.6875	.6875	1.5444	.6875	-3.0260	.6875	51.5705	.6786	24.0298	.6875	1346.7744
11	.6563	.6563	1.7105	.6563	-2.7491	.6563	50.4412	.6464	22.6987	.6563	1436.6945
12	.6250	.6250	1.9518	.6250	-2.5167	.6250	48.7109	.6143	20.1833	.6250	1513.6100
13	.5938	.5938	2.1692	.5938	-2.2244	.5938	45.8657	.5821	17.5319	.5938	1584.0968
14	.5625	.5625	2.3808	.5625	-1.8467	.5625	41.5757	.5500	12.3536	.5625	1644.2614
15	.5313	.5313	2.4946	.5313	-1.4206	.5313	34.7531	.5179	9.6214	.5313	1711.4227
16	.5000	.5000	2.6598	.5000	-.9420	.5000	27.1473	.4857	4.7197	.5000	1760.2479
17	.4688	.4688	2.8328	.4688	-.3770	.4688	18.3364	.4536	-3.8576	.4688	1797.3748
18	.4375	.4375	2.9307	.4375	.3384	.4375	8.5195	.4214	-8.7994	.4375	1849.8007
19	.4063	.4063	2.8725	.4063	1.1044	.4063	-2.1194	.3893	-19.8105	.4063	1905.3349
20	.3750	.3750	-1.1870	.3750	1.6390	.3750	-18.8717	.3571	-28.8464	.3750	2411.8914
21	.3438	.3438	3.4304	.3438	1.2143	.3438	-42.9978	.3250	-49.4112	.3438	1914.4488
22	.3125	.3125	3.4417	.3125	-.2424	.3125	-82.2106	.2929	-73.9244	.3125	2377.3086
23	.2813	.2813	3.5360	.2813	1.7374	.2813	-121.3626	.2607	-100.9777	.2813	1799.7259
24	.2500	.2500	3.5962	.2500	11.6855	.2500	-170.1902	.2286	-129.5244	.2500	1750.2485
25	.2188	.2188	3.6615	.2188	28.8759	.2188	-213.2488	.1964	-168.5015	.2188	1673.8101
26	.1875	.1875	3.7119	.1875	49.8522	.1875	-259.9132	.1643	-199.0200	.1875	1603.4360
27	.1563	.1563	3.7645	.1563	63.6612	.1563	-294.5792	.1321	-227.0555	.1563	1542.7842
28	.1250	.1250	3.8190	.1250	67.6203	.1250	-314.5814	.1000	-256.8217	.1250	1493.5615
29	.0938	.0938	3.8704	.0938	67.8533	.0938	-323.8562	.0679	-313.1910	.0938	1439.4028
30	.0625	.0625	4.0452	.0625	64.1781	.0625	-318.0546	.0357	-301.6170	.0625	1395.2751
31	.0313	.0313	3.7273	.0313	59.1752	.0313	-306.1475	.0036	-390.5942	.0313	1315.1689
32	.0313	.0313	-2.0986	.0313	20.0463	.0313	38.5597	.0607	53.0857	.0313	-774.5666
33	.0625	.0625	-1.7847	.0625	59.1447	.0625	-147.0558	.0929	55.9563	.0625	-245.9485
34	.0937	.0937	-1.3110	.0937	75.5428	.0937	-243.1517	.1250	57.6540	.0937	128.3677
35	.1250	.1250	-.7343	.1250	86.0979	.1250	-298.0271	.1552	44.6364	.1250	439.0277
36	.1563	.1563	-.0684	.1563	76.0690	.1563	-316.0947	.1853	6.7466	.1563	734.5830
37	.1875	.1875	.5535	.1875	25.7422	.1875	-294.0541	.2155	-32.6152	.1875	1034.5656
38	.2188	.2188	1.0858	.2188	-37.4853	.2188	-251.8086	.2457	-67.3864	.2188	1319.4035
39	.2500	.2500	1.4408	.2500	-54.6629	.2500	-211.5466	.2759	-90.1078	.2500	1533.4164
40	.2813	.2813	1.6988	.2813	-40.9663	.2813	-164.2406	.3060	-107.1067	.2813	1716.9271
41	.3125	.3125	1.8322	.3125	-7.2238	.3125	-120.1525	.3362	-113.3012	.3125	1866.5859
42	.3438	.3438	1.8675	.3438	12.3028	.3438	-91.4976	.3664	-110.1136	.3438	1949.5734
43	.3750	.3750	1.8318	.3750	18.5201	.3750	-70.0210	.3966	-104.1926	.3750	1996.9997
44	.4063	.4063	1.7831	.4063	17.1651	.4063	-55.8055	.4267	-94.5723	.4063	2021.0725
45	.4375	.4375	1.6282	.4375	13.5137	.4375	-42.1093	.4569	-91.3018	.4375	2006.4598
46	.4688	.4688	1.4959	.4688	12.4504	.4688	-26.6607	.4871	-81.2246	.4688	2010.8281
47	.5000	.5000	1.4538	.5000	12.2531	.5000	-10.9074	.5172	-76.0135	.5000	2014.4030
48	.5313	.5313	1.2463	.5313	11.9896	.5313	4.2757	.5474	-72.4756	.5313	1961.1819
49	.5625	.5625	1.1544	.5625	11.4311	.5625	16.9513	.5776	-60.5096	.5625	1922.7421
50	.5938	.5938	1.0832	.5938	10.5534	.5938	27.4695	.6078	-54.5505	.5938	1865.9004
51	.6250	.6250	1.0269	.6250	9.3045	.6250	35.8149	.6379	-48.5193	.6250	1793.9780
52	.6563	.6563	1.4608	.6563	9.0808	.6563	43.1271	.6681	-40.0172	.6563	1744.0449
53	.6875	.6875	.9389	.6875	7.4211	.6875	45.2747	.6983	-39.3476	.6875	1578.7011
54	.7188	.7188	.9441	.7188	6.8230	.7188	48.1426	.7284	-35.3002	.7188	1464.1293

Figure A.2.- Continued.

55	.7500	.7500	1.0328	.7500	6.3076	.7500	49.1325	.7586	-31.5621	.7500	1334.1984
56	.7813	.7813	1.2671	.7813	5.8215	.7813	48.2877	.7888	-27.9952	.7813	1190.4777
57	.8125	.8125	1.6887	.8125	5.4206	.8125	45.5817	.8190	-24.4539	.8125	1030.4991
58	.8437	.8437	2.3181	.8437	5.0341	.8437	40.6758	.8491	-21.2165	.8437	846.5093
59	.8750	.8750	3.1611	.8750	4.5903	.8750	33.2001	.8793	-18.9875	.8750	630.1062
60	.9062	.9062	4.3495	.9062	3.9949	.9062	22.7723	.9095	-18.1252	.9062	376.1257
61	.9375	.9375	-.8723	.9375	4.0444	.9375	9.3163	.9397	-13.9357	.9375	73.0336
62	.9687	.9687	24.4162	.9687	8.5940	.9687	-15.1745	.9698	-38.7317	.9687	-419.8816

Figure A.2.- Continued.

UNIT PERTURBATION OF WM AND UNIT STRAINING OF XBASE
FOR CALIBRATION SOLUTIONS 6 THROUGH 6

POINT	XBASE	• 6TH CALB SOLN • XSTRUNIT	WMUNIT
1	.9687	.9687	85.4341
2	.9375	.9375	-195.5425
3	.9062	.9062	-73.2089
4	.8750	.8750	-103.2774
5	.8437	.8437	-137.1931
6	.8125	.8125	-130.1613
7	.7813	.7813	-110.6727
8	.7500	.7500	-99.2898
9	.7188	.7188	-98.7822
10	.6875	.6875	-106.3977
11	.6563	.6563	-93.3852
12	.6250	.6250	-88.7640
13	.5938	.5938	-82.9426
14	.5625	.5625	-80.9244
15	.5313	.5313	-70.7242
16	.5000	.5000	-68.5586
17	.4688	.4688	-68.1042
18	.4375	.4375	-40.0619
19	.4063	.4063	-24.6671
20	.3750	.3750	-8.0421
21	.3438	.3438	-55.7364
22	.3125	.3125	-139.4742
23	.2813	.2813	-218.3122
24	.2500	.2500	-304.8216
25	.2188	.2188	-363.2004
26	.1875	.1875	-220.2327
27	.1563	.1563	179.5793
28	.1250	.1250	972.0918
29	.0938	.0938	2388.4336
30	.0625	.0625	6013.0030
31	.0313	.0313	10829.3858
32	.0313	.0313	4319.9999
33	.0625	.0625	974.4519
34	.0937	.0937	-336.1720
35	.1250	.1250	-849.5837
36	.1563	.1563	-863.6749
37	.1875	.1875	-370.0835
38	.2188	.2188	321.6473
39	.2500	.2500	678.9298
40	.2813	.2813	973.2190
41	.3125	.3125	1123.6623
42	.3438	.3438	1043.2128
43	.3750	.3750	941.1998
44	.4063	.4063	759.0115
45	.4375	.4375	621.9917
46	.4688	.4688	507.6942
47	.5000	.5000	403.4571
48	.5313	.5313	357.8155
49	.5625	.5625	304.2645
50	.5938	.5938	258.3880
51	.6250	.6250	218.0293
52	.6563	.6563	132.6033
53	.6875	.6875	150.2109
54	.7188	.7188	125.2795

Figure A.2.- Continued.

55	.7500	.7500	109.5129
56	.7813	.7813	108.1062
57	.8125	.8125	118.9669
58	.8437	.8437	116.0773
59	.8750	.8750	66.5779
60	.9062	.9062	1.4576
61	.9375	.9375	-18.1001
62	.9687	.9687	46.1274

Figure A.2.- Continued.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                   *
*                    C O N M I N                    *
*                                                   *
*                FORTRAN PROGRAM FOR                *
*                                                   *
*       CONSTRAINED FUNCTION MINIMIZATION           *
*                                                   *
* * * * * * * * * * * * * * * * * * * * * * * * * * *


CONSTRAINED FUNCTION MINIMIZATION

CONTROL PARAMETERS

IPRINT  NDV   ITMAX   NCON   NSIDE  ICNDIR   NSCAL   NFDG
  5      6      6       6      1       7      -2       0

LINOBJ  ITRM    N1     N2     N3      N4      N5
  0      3      8      18     20      20      40

    CT              CTMIN            CTL              CTLMIN
 -.10000E+00      .40000E-02      -.10000E-01       .10000E-02

   THETA             PHI            DELFUN            DABFUN
  .10000E+01      .50000E+01       .10000E-03       .17077E-02

    FDCH             FDCHM          ALPHAX            ABOBJ1
  .10000E-01      .10000E-02       .10000E+00       .50000E-01

LOWER BOUNDS ON DECISION VARIABLES (VLB)
  1)    -.15000E+01   .80000E+00   .44444E+00   .33333E+00   .60000E+00   .60000E+00

UPPER BOUNDS ON DECISION VARIABLES (VUB)
  1)    0.            .24000E+01   .12222E+01   .26667E+01   .20000E+01   .24000E+01

SCALING VECTOR (SCAL)
  .1000E+02    .2500E+00    .4500E+00    .1500E+01    .5000E-01    .5000E-02

ALL CONSTRAINTS ARE NON-LINEAR


INITIAL FUNCTION INFORMATION

OBJ =    .170765E+01

DECISION VARIABLES (X-VECTOR)
  1)    -.93094E+01   .29650E+00   .55000E+00   .95806E+00   .32891E-01   .51525E-02

CONSTRAINT VALUES (G-VECTOR)
  1)    -.28002E+01  -.97200E+00  -.14455E+02  -.96575E-01  -.15965E+01  -.40346E+00
```

Figure A.2.- Continued.

BEGIN ITERATION NUMBER 1

CT = -.10000E+00 CTL = -.10000E-01 PHI = .50000E+01

*** INPUT FOR BLADE ELEMENT PROGRAM ***

NO. BLADES	INLET RADIUS (L)	CONE ANGLE (DEG)	SOLIDITY	INLET BLADE ANGLE (DEG)	OUTLET BLADE ANGLE (DEG)	TRANS. LOC. /CHORD	MAX.TH. LOC. /CHORD	IN/OUT TURNING RATE	MAX. THICK. /CHORD	L.E. RAD. /CHORD	T.E. RAD. /CHORD
34	.45400	6.66400	2.25200	52.000	-9.2163	.29650	.55000	.95806	.03289	.00515	.00500

X ARRAY

.968750	.937500	.906250	.875000	.843750
.812500	.781250	.750000	.718750	.687500
.656250	.625000	.593750	.562500	.531250
.500000	.468750	.437500	.406250	.375000
.343750	.312500	.281250	.250000	.218750
.187500	.156250	.125000	.093750	.062500
.031250	.031250	.062500	.093750	.125000
.156250	.187500	.218750	.250000	.281250
.312500	.343750	.375000	.406250	.437500
.468750	.500000	.531250	.562500	.593750
.625000	.656250	.687500	.718750	.750000
.781250	.812500	.843750	.875000	.906250
.937500	.968750			

Y ARRAY

584.076	500.337	475.238	463.430	455.400
448.954	444.076	440.771	438.803	437.961
439.329	440.014	441.234	442.820	445.789
447.710	450.009	454.064	457.497	462.420
467.467	473.780	479.851	488.249	496.446
508.451	523.050	541.205	564.395	599.393
633.203	836.552	847.968	851.223	846.083
834.497	820.359	805.330	788.882	770.861
754.360	739.430	723.656	710.984	697.714
688.937	680.986	671.750	665.374	658.826
652.092	645.132	635.822	629.366	622.726
615.621	607.512	597.461	584.391	567.523
548.168	535.779			

OUTPUT FROM OBJCON

```
OBJECTIVE FUNCTION    DIFFS       =  .85122E+03
                      WO          =  .49855E+03
                      DIFFS/WO    =  .17074E+01

CONSTRAINT 1          BLTKS       =  .27996E+00

CONSTRAINT 2          YMAX        =  .63320E+03
                      YMIN        =  .43796E+03
                      DIFFP       =  .14458E+01

CONSTRAINT 3       WMB(MBO-2,1)   =  .54817E+03
                   WMB(MBO-2,2)   =  .50034E+03
```

Figure A.2.- Continued.

TECLSR = .59789E+00

*** INPUT FOR BLADE ELEMENT PROGRAM ***

NO. BLADES	INLET RADIUS (L)	CONE ANGLE (DEG)	SOLIDITY	INLET BLADE ANGLE (DEG)	OUTLET BLADE ANGLE (DEG)	TRANS. LOC. /CHORD	MAX.TH. LOC. /CHORD	IN/OUT TURNING RATE	MAX. THICK. /CHORD	L.E. RAD. /CHORD	T.E. RAD. /CHORD
34	.45400	6.66400	2.25200	52.000	-9.3094	.29946	.55000	.95806	.03289	.00515	.00500

X ARRAY

.968750	.937500	.906250	.875000	.843750
.812500	.781250	.750000	.718750	.687500
.656250	.625000	.593750	.562500	.531250
.500000	.468750	.437500	.406250	.375000
.343750	.312500	.281250	.250000	.218750
.187500	.156250	.125000	.093750	.062500
.031250	.031250	.062500	.093750	.125000
.156250	.187500	.218750	.250000	.281250
.312500	.343750	.375000	.406250	.437500
.468750	.500000	.531250	.562500	.593750
.625000	.656250	.687500	.718750	.750000
.781250	.812500	.843750	.875000	.906250
.937500	.968750			

Y ARRAY

585.909	500.527	475.378	463.529	455.444
448.945	444.024	440.683	438.683	437.808
439.162	439.825	441.025	442.593	445.553
447.460	449.744	453.792	457.233	462.536
467.151	473.459	479.527	487.949	496.191
508.253	522.888	541.050	564.235	599.207
633.031	836.807	848.310	851.569	846.407
834.729	820.384	805.118	788.585	770.581
754.168	739.293	723.541	710.869	697.603
688.835	680.887	671.669	665.301	658.757
652.024	645.023	635.756	629.298	622.649
615.521	607.371	597.260	584.111	567.130
548.261	533.531			

OUTPUT FROM OBJCON

OBJECTIVE FUNCTION	DIFFS	=	.85157E+03
	WO	=	.49855E+03
	DIFFS/WO	=	.17081E+01
CONSTRAINT 1	BLTKS	=	.28326E+00
CONSTRAINT 2	YMAX	=	.63303E+03
	YMIN	=	.43781E+03
	DIFFP	=	.14459E+01
CONSTRAINT 3	WMB(MBO-2,1)	=	.54826E+03
	WMB(MBO-2,2)	=	.50053E+03

Figure A.2.- Continued.

TECLSR = .59667E+00

*** INPUT FOR BLADE ELEMENT PROGRAM ***

NO. BLADES	INLET RADIUS (L)	CONE ANGLE (DEG)	SOLIDITY	INLET BLADE ANGLE (DEG)	OUTLET BLADE ANGLE (DEG)	TRANS. LOC. /CHORD	MAX.TH. LOC. /CHORD	IN/OUT TURNING RATE	MAX. THICK. /CHORD	L.E. RAD. /CHORD	T.E. RAD. /CHORD
34	.45400	6.66400	2.25200	52.000	-9.3094	.29650	.54450	.95806	.03289	.00515	.00500

X ARRAY

.968750	.937500	.906250	.875000	.843750
.812500	.781250	.750000	.718750	.687500
.656250	.625000	.593750	.562500	.531250
.500000	.468750	.437500	.406250	.375000
.343750	.312500	.281250	.250000	.218750
.187500	.156250	.125000	.093750	.062500
.031250	.031250	.062500	.093750	.125000
.156250	.187500	.218750	.250000	.281250
.312500	.343750	.375000	.406250	.437500
.468750	.500000	.531250	.562500	.593750
.625000	.656250	.687500	.718750	.750000
.781250	.812500	.843750	.875000	.906250
.937500	.968750			

Y ARRAY

585.942	500.405	475.206	463.321	455.209
448.692	443.759	440.410	438.407	437.534
438.893	439.565	440.779	442.369	445.366
447.313	449.644	453.744	457.241	462.635
467.384	473.912	480.189	488.851	497.278
509.535	524.319	542.580	565.816	600.766
634.540	836.535	848.943	852.683	847.791
836.242	821.925	806.614	789.911	771.606
754.851	739.759	723.871	711.125	697.795
688.945	680.910	671.610	665.174	658.574
651.799	644.759	635.485	629.013	622.360
615.238	607.104	597.022	583.914	566.993
548.198	533.589			

OUTPUT FROM OBJCON

```
   OBJECTIVE FUNCTION    DIFFS      =  .85268E+03
                         WO         =  .49855E+03
                         DIFFS/WO   =  .17103E+01

   CONSTRAINT 1          BLTKS      =  .30418E+00

   CONSTRAINT 2          YMAX       =  .63454E+03
                         YMIN       =  .43753E+03
                         DIFFP      =  .14503E+01

   CONSTRAINT 3       WMB(MBO-2,1)  =  .54820E+03
                      WMB(MBO-2,2)  =  .50040E+03
```

Figure A.2.- Continued.

TECLSR = .59741E+00

*** INPUT FOR BLADE ELEMENT PROGRAM ***

NO. BLADES	INLET RADIUS (L)	CONE ANGLE (DEG)	SOLIDITY	INLET BLADE ANGLE (DEG)	OUTLET BLADE ANGLE (DEG)	TRANS. LOC. /CHORD	MAX.TH. LOC. /CHORD	IN/OUT TURNING RATE	MAX. THICK. /CHORD	L.E. RAD. /CHORD	T.E. RAD. /CHORD
34	.45400	6.66400	2.25200	52.000	-9.3094	.29650	.55000	.96764	.03289	.00515	.00500

X ARRAY

.968750	.937500	.906250	.875000	.843750
.812500	.781250	.750000	.718750	.687500
.656250	.625000	.593750	.562500	.531250
.500000	.468750	.437500	.406250	.375000
.343750	.312500	.281250	.250000	.218750
.187500	.156250	.125000	.093750	.062500
.031250	.031250	.062500	.093750	.125000
.156250	.187500	.218750	.250000	.281250
.312500	.343750	.375000	.406250	.437500
.468750	.500000	.531250	.562500	.593750
.625000	.656250	.687500	.718750	.750000
.781250	.812500	.843750	.875000	.906250
.937500	.968750			

Y ARRAY

586.337	500.777	475.659	463.823	455.737
449.235	444.309	440.961	438.951	438.061
439.395	440.040	441.220	442.758	445.677
447.543	449.777	453.767	457.140	462.344
466.803	472.880	478.743	486.863	494.787
506.576	521.012	539.030	562.063	597.128
632.376	837.637	848.781	851.710	846.170
834.089	819.507	804.075	787.352	769.212
752.705	737.799	722.181	709.606	696.498
687.856	679.943	670.825	664.577	658.097
651.430	644.488	635.279	628.867	622.258
615.166	607.047	596.962	583.832	566.876
548.079	533.116			

OUTPUT FROM OBJCON

OBJECTIVE FUNCTION	DIFFS	=	.85171E+03
	WO	=	.49855E+03
	DIFFS/WO	=	.17084E+01
CONSTRAINT 1	BLTKS	=	.27981E+00
CONSTRAINT 2	YMAX	=	.63238E+03
	YMIN	=	.43806E+03
	DIFFP	=	.14436E+01
CONSTRAINT 3	WMB(MBO-2,1)	=	.54808E+03
	WMB(MBO-2,2)	=	.50078E+03

Figure A.2.- Continued.

TECLSR = .59128E+00

*** INPUT FOR BLADE ELEMENT PROGRAM ***

NO. BLADES	INLET RADIUS (L)	CONE ANGLE (DEG)	SOLIDITY	INLET BLADE ANGLE (DEG)	OUTLET BLADE ANGLE (DEG)	TRANS. LOC. /CHORD	MAX.TH. LOC. /CHORD	IN/OUT TURNING RATE	MAX. THICK. /CHORD	L.E. RAD. /CHORD	T.E. RAD. /CHORD
34	.45400	6.66400	2.25200	52.000	-9.3094	.29650	.55000	.95806	.03389	.00515	.00500

X ARRAY

.968750	.937500	.906250	.875000	.843750
.812500	.781250	.750000	.718750	.687500
.656250	.625000	.593750	.562500	.531250
.500000	.468750	.437500	.406250	.375000
.343750	.312500	.281250	.250000	.218750
.187500	.156250	.125000	.093750	.062500
.031250	.031250	.062500	.093750	.125000
.156250	.187500	.218750	.250000	.281250
.312500	.343750	.375000	.406250	.437500
.468750	.500000	.531250	.562500	.593750
.625000	.656250	.687500	.718750	.750000
.781250	.812500	.843750	.875000	.906250
.937500	.968750			

Y ARRAY

585.812	500.848	475.937	464.269	456.323
449.937	445.117	441.870	439.959	439.164
440.607	441.346	442.616	444.242	447.268
449.223	451.543	455.641	459.135	464.943
469.062	475.837	481.321	489.665	497.779
509.709	524.242	542.343	565.474	600.412
634.171	835.973	847.888	851.474	846.590
835.238	821.342	806.548	790.281	772.420
756.056	741.206	725.483	712.839	699.569
690.809	682.865	673.595	667.190	660.591
653.790	646.740	637.313	630.742	623.964
616.694	608.385	598.092	584.727	567.494
548.322	533.086			

OUTPUT FROM OBJCON

OBJECTIVE FUNCTION	DIFFS	=	.85147E+03
	WO	=	.49855E+03
	DIFFS/WO	=	.17079E+01
CONSTRAINT 1	BLTKS	=	.29081E+00
CONSTRAINT 2	YMAX	=	.63417E+03
	YMIN	=	.43916E+03
	DIFFP	=	.14440E+01
CONSTRAINT 3	WMB(MBO-2,1)	=	.54832E+03
	WMB(MBO-2,2)	=	.50085E+03

Figure A.2.- Continued.

TECLSR = .59343E+00

*** INPUT FOR BLADE ELEMENT PROGRAM ***

NO. BLADES	INLET RADIUS (L)	CONE ANGLE (DEG)	SOLIDITY	INLET BLADE ANGLE (DEG)	OUTLET BLADE ANGLE (DEG)	TRANS. LOC. /CHORD	MAX.TH. LOC. /CHORD	IN/OUT TURNING RATE	MAX. THICK. /CHORD	L.E. RAD. /CHORD	T.E. RAD. /CHORD
34	.45400	6.66400	2.25200	52.000	-9.3094	.29650	.55000	.95806	.03289	.00615	.00500

X ARRAY

.968750	.937500	.906250	.875000	.843750
.812500	.781250	.750000	.718750	.687500
.656250	.625000	.593750	.562500	.531250
.500000	.468750	.437500	.406250	.375000
.343750	.312500	.281250	.250000	.218750
.187500	.156250	.125000	.093750	.062500
.031250	.031250	.062500	.093750	.125000
.156250	.187500	.218750	.250000	.281250
.312500	.343750	.375000	.406250	.437500
.468750	.500000	.531250	.562500	.593750
.625000	.656250	.687500	.718750	.750000
.781250	.812500	.843750	.875000	.906250
.937500	.968750			

Y ARRAY

586.031	500.331	475.311	463.435	455.316
448.824	443.923	440.593	438.594	437.711
439.077	439.744	440.949	442.517	445.486
447.394	449.677	453.751	457.205	462.523
467.092	473.320	479.303	487.610	495.742
507.885	522.879	541.822	566.423	605.030
643.685	841.067	849.109	851.009	845.302
833.639	819.938	805.551	789.426	771.676
755.314	740.299	724.427	711.577	698.185
689.306	681.254	671.991	665.571	658.984
652.214	645.128	635.885	629.403	622.739
615.612	607.474	597.362	584.164	567.120
548.231	533.552			

OUTPUT FROM OBJCON

OBJECTIVE FUNCTION	DIFFS	=	.85101E+03
	WO	=	.49855E+03
	DIFFS/WO	=	.17070E+01
CONSTRAINT 1	BLTKS	=	.42386E+00
CONSTRAINT 2	YMAX	=	.64369E+03
	YMIN	=	.43771E+03
	DIFFP	=	.14706E+01
CONSTRAINT 3	WMB(MBO-2,1)	=	.54823E+03
	WMB(MBO-2,2)	=	.50033E+03

Figure A.2.- Continued.

```
                    TECLSR   =  .59876E+00

THERE ARE    1 ACTIVE CONSTRAINTS
CONSTRAINT NUMBERS ARE
    4

THERE ARE    0 VIOLATED CONSTRAINTS

THERE ARE    1 ACTIVE SIDE CONSTRAINTS
DECISION VARIABLES AT LOWER OR UPPER BOUNDS (MINUS INDICATES LOWER BOUND)
    3

GRADIENT OF OBJ
  1)    -.26392E-01   .37807E-01  -.21940E+00   .11434E+00   .12830E-01  -.33759E-02

GRADIENTS OF ACTIVE AND VIOLATED CONSTRAINTS
CONSTRAINT NUMBER    4
  1)     .21332E-01   .22680E-01  -.24471E+00  -.18588E+00  -.44941E-01   .78414E-01

SIDE CONSTRAINT ON VARIABLE    3
  1)    0.            0.             .10000E+01  0.           0.           0.


PUSH-OFF FACTORS, (THETA(I), I=1,NAC)
  1)     .11730E-02  0.

CONSTRAINT PARAMETER, BETA =    .36299E+00

SEARCH DIRECTION (S-VECTOR)
  1)     .32046E+00  -.10000E+01   .39056E-13  -.43887E+00   .19546E+00  -.72800E+00

ONE-DIMENSIONAL SEARCH
INITIAL SLOPE =  -.9148E-01  PROPOSED ALPHA =   .1186E+00



* * CONSTRAINED ONE-DIMENSIONAL SEARCH INFORMATION * * *


PROPOSED DESIGN
ALPHA =  .11860E+00
X-VECTOR
-.8929E+01   .2668E+00   .5500E+00   .8800E+00   .3405E-01   .4721E-02
```


Figure A.2.- Continued.

*** INPUT FOR BLADE ELEMENT PROGRAM ***

NO. BLADES	INLET RADIUS (L)	CONF ANGLE (DEG)	SOLIDITY	INLET BLADE ANGLE (DEG)	OUTLET BLADE ANGLE (DEG)	TRANS. LOC. /CHORD	MAX.TH. LOC. /CHORD	IN/OUT TURNING RATE	MAX. THICK. /CHORD	L.E. RAD. /CHORD	T.E. RAD. /CHORD
34	.45400	6.66400	2.25200	52.000	-8.9293	.26685	.55000	.87999	.03405	.00472	.00500

X ARRAY

.968750	.937500	.906250	.875000	.843750
.812500	.781250	.750000	.718750	.687500
.656250	.625000	.593750	.562500	.531250
.500000	.468750	.437500	.406250	.375000
.343750	.312500	.281250	.250000	.218750
.187500	.156250	.125000	.093750	.062500
.031250	.031250	.062500	.093750	.125000
.156250	.187500	.218750	.250000	.281250
.312500	.343750	.375000	.406250	.437500
.468750	.500000	.531250	.562500	.593750
.625000	.656250	.687500	.718750	.750000
.781250	.812500	.843750	.875000	.906250
.937500	.968750			

Y ARRAY

575.226	498.998	473.581	461.860	454.132
448.015	443.428	440.389	438.698	438.254
439.690	440.765	442.260	444.293	447.409
449.890	452.824	457.082	461.385	466.261
473.241	481.773	489.466	499.183	509.712
522.635	538.120	557.091	582.208	613.227
650.082	805.736	837.247	843.322	841.747
835.030	826.076	815.978	802.800	786.405
769.349	753.299	736.954	722.919	709.390
699.074	690.470	680.975	673.220	666.054
658.757	651.200	641.678	634.454	627.364
619.856	611.397	601.083	587.877	571.010
549.410	545.087			

OUTPUT FROM OBJCON

OBJECTIVE FUNCTION	DIFFS	=	.84332E+03
	WO	=	.49855E+03
	DIFFS/WO	=	.16916E+01
CONSTRAINT 1	BLTKS	=	.19679E+00
CONSTRAINT 2	YMAX	=	.65008E+03
	YMIN	=	.43825E+03
	DIFFP	=	.14833E+01
CONSTRAINT 3	WMB(MBO-2,1)	=	.54941E+03
	WMB(MBO-2,2)	=	.49900E+03

Figure A.2.- Continued.

```
                    TECLSR   =  .63015E+00

  OBJ =   .16916E+01

  CONSTRAINT VALUFS
-.1968E+01  -.9803E+00  -.1483E+02  -.7291E-01  -.1630E+01  -.3698E+00

  TWO-POINT INTERPOLATION


  PROPOSED DESIGN
  ALPHA =  .38598E+00
  X-VECTOR
-.8072E+01   .2000E+00   .5500E+00   .7040E+00   .3666E-01   .3748E-02
```

Figure A.2.- Continued.

*** INPUT FOR BLADE ELEMENT PROGRAM ***

NO. BLADES	INLET RADIUS (L)	CONE ANGLE (DEG)	SOLIDITY	INLET BLADE ANGLE (DEG)	OUTLET BLADE ANGLE (DEG)	TRANS. LOC. /CHORD	MAX.TH. LOC. /CHORD	IN/OUT TURNING RATE	MAX. THICK. /CHORD	L.E. RAD. /CHORD	T.E. RAD. /CHORD
34	.45400	6.66400	2.25200	52.000	-8.0725	.20000	.55000	.70396	.03666	.00375	.00500

X ARRAY

.968750	.937500	.906250	.875000	.843750
.812500	.781250	.750000	.718750	.687500
.656250	.625000	.593750	.562500	.531250
.500000	.468750	.437500	.406250	.375000
.343750	.312500	.281250	.250000	.218750
.187500	.156250	.125000	.093750	.062500
.031250	.031250	.062500	.093750	.125000
.156250	.187500	.218750	.250000	.281250
.312500	.343750	.375000	.406250	.437500
.468750	.500000	.531250	.562500	.593750
.625000	.656250	.687500	.718750	.750000
.781250	.812500	.843750	.875000	.906250
.937500	.968750			

Y ARRAY

550.516	494.472	469.416	457.972	451.012
445.739	441.893	439.521	438.501	438.930
440.705	442.613	444.763	447.681	451.328
454.923	459.030	464.065	469.819	475.041
483.575	497.330	510.004	521.974	536.848
553.060	570.624	590.966	617.176	649.870
680.737	771.425	811.313	824.141	828.722
828.531	829.594	831.320	830.192	820.187
804.400	786.288	768.411	751.285	736.461
723.326	712.703	702.584	692.254	683.150
674.537	665.675	655.453	646.455	638.311
629.888	620.680	609.854	596.460	579.793
552.461	570.387			

```
OUTPUT FROM OBJCON

   OBJECTIVE FUNCTION   DIFFS     =  .83132E+03
                        WO        =  .49855E+03
                        DIFFS/WO  =  .16675E+01

   CONSTRAINT 1         BLTKS     = -.15560E-01

   CONSTRAINT 2         YMAX      =  .68074E+03
                        YMIN      =  .43850E+03
                        DIFFP     =  .15524E+01

   CONSTRAINT 3     WMB(MBO-2,1)  =  .55246E+03
                    WMB(MBO-2,2)  =  .49447E+03
```

Figure A.2.- Continued.

```
                    TECLSR    =  .72486E+00

 OBJ =    .16675E+01

 CONSTRAINT VALUES
.1556E+00  -.1002E+01  -.1552E+02  -.2974E-01  -.1725E+01  -.2751E+00

 THREE-POINT INTERPOLATION


 PROPOSED DESIGN
 ALPHA =  .34405E+00
 X-VECTOR
-.8207E+01   .2105E+00   .5500E+00   .7316E+00   .3625E-01   .3900E-02
```


Figure A.2.- Continued.

*** INPUT FOR BLADE ELEMENT PROGRAM ***

NO. BLADES	INLET RADIUS (L)	CONE ANGLE (DEG)	SOLIDITY	INLET BLADE ANGLE (DEG)	OUTLET BLADE ANGLE (DEG)	TRANS. LOC. /CHORD	MAX.TH. LOC. /CHORD	IN/OUT TURNING RATE	MAX. THICK. /CHORD	L.E. RAD. /CHORD	T.E. RAD. /CHORD
34	.45400	6.66400	2.25200	52.000	-8.2069	.21048	.55000	.73157	.03625	.00390	.00500

X ARRAY

.968750	.937500	.906250	.875000	.843750
.812500	.781250	.750000	.718750	.687500
.656250	.625000	.593750	.562500	.531250
.500000	.468750	.437500	.406250	.375000
.343750	.312500	.281250	.250000	.218750
.187500	.156250	.125000	.093750	.062500
.031250	.031250	.062500	.093750	.125000
.156250	.187500	.218750	.250000	.281250
.312500	.343750	.375000	.406250	.437500
.468750	.500000	.531250	.562500	.593750
.625000	.656250	.687500	.718750	.750000
.781250	.812500	.843750	.875000	.906250
.937500	.968750			

Y ARRAY

554.440	495.278	470.079	458.591	451.514
446.110	442.149	439.674	438.551	438.852
440.560	442.346	444.394	447.189	450.736
454.173	458.122	463.009	468.580	473.631
482.259	495.175	506.951	518.634	532.909
548.497	565.728	585.899	612.229	643.711
676.661	773.620	815.504	827.247	831.057
830.276	829.937	829.753	826.301	815.041
798.820	780.988	763.374	746.734	732.177
719.420	709.164	699.143	689.165	680.415
672.011	663.358	653.257	644.542	636.568
628.294	619.208	608.467	595.108	578.415
551.941	566.495			

OUTPUT FROM OBJCON

OBJECTIVE FUNCTION	DIFFS	=	.83106E+03
	WO	=	.49855E+03
	DIFFS/WO	=	.16670E+01
CONSTRAINT 1	BLTKS	=	.20848E-01
CONSTRAINT 2	YMAX	=	.67666E+03
	YMIN	=	.43855E+03
	DIFFP	=	.15429E+01
CONSTRAINT 3	WMB(MBO-2,1)	=	.55194E+03
	WMB(MBO-2,2)	=	.49528E+03

Figure A.2.- Continued.

```
                     TECLSR    =  .70829E+00

OBJ =    .16670E+01

CONSTRAINT VALUES
-.2085E+00  -.9979E+00  -.1543E+02  -.3566E-01  -.1708E+01  -.2917E+00

• • • END OF ONE-DIMENSIONAL SEARCH

CALCULATED ALPHA =    .34405E+00

OBJ =    .166695E+01

DECISION VARIABLES (X-VECTOR)
  1)    -.82069E+01   .21048E+00   .55000E+00   .73157E+00   .36253E-01   .39002E-02

CONSTRAINT VALUES (G-VECTOR)
  1)    -.20848E+00  -.99792E+00  -.15429E+02  -.35658E-01  -.17083E+01  -.29171E+00


BEGIN ITERATION NUMBER    2

CT =   -.10000E+00     CTL =   -.10000E-01     PHI =    .50000E+01
```

Figure A.2.- Continued.

```
FINAL OPTIMIZATION INFORMATION

 OBJ =     .166529E+01

 DECISION VARIABLES (X-VECTOR)
   1)    -.80297E+01   .20000E+00   .55000E+00   .73559E+00   .36263E-01   .39389E-02

 CONSTRAINT VALUES (G-VECTOR)
   1)    -.44211E-01  -.99956E+00  -.15409E+02  -.36960E-01  -.17108E+01  -.28922E+00

 THERE ARE     2 ACTIVE CONSTRAINTS
 CONSTRAINT NUMBERS ARE
     1    4

 THERE ARE     0 VIOLATED CONSTRAINTS

 THERE ARE     2 ACTIVE SIDE CONSTRAINTS
 DECISION VARIABLES AT LOWER OR UPPER BOUNDS (MINUS INDICATES LOWER BOUND)
    -2    3

 TERMINATION CRITERION
      ABS(OBJ(I)-OBJ(I-1))   LESS THAN DABFUN FOR  3 ITERATIONS

 NUMBER OF ITERATIONS =    4

 OBJECTIVE FUNCTION WAS EVALUATED          35  TIMES

 CONSTRAINT FUNCTIONS WERE EVALUATED       35  TIMES
```

Figure A.2.- Continued.

*** INPUT FOR BLADE ELEMENT PROGRAM ***

NO. BLADES	INLET RADIUS (L)	CONE ANGLE (DEG)	SOLIDITY	INLET BLADE ANGLE (DEG)	OUTLET BLADE ANGLE (DEG)	TRANS. LOC. /CHORD	MAX.TH. LOC. /CHORD	IN/OUT TURNING RATE	MAX. THICK. /CHORD	L.E. RAD. /CHORD	T.E. RAD. /CHORD
34	.45400	6.66400	2.25200	52.000	-8.0297	.20000	.55000	.73559	.03626	.00394	.00500

X ARRAY

.968750	.937500	.906250	.875000	.843750
.812500	.781250	.750000	.718750	.687500
.656250	.625000	.593750	.562500	.531250
.500000	.468750	.437500	.406250	.375000
.343750	.312500	.281250	.250000	.218750
.187500	.156250	.125000	.093750	.062500
.031250	.031250	.062500	.093750	.125000
.156250	.187500	.218750	.250000	.281250
.312500	.343750	.375000	.406250	.437500
.468750	.500000	.531250	.562500	.593750
.625000	.656250	.687500	.718750	.750000
.781250	.812500	.843750	.875000	.906250
.937500	.968750			

Y ARRAY

551.119	494.850	469.915	458.527	451.557
446.251	442.372	439.964	438.901	439.254
440.997	442.810	444.888	447.705	451.267
454.707	458.662	463.548	469.088	473.693
482.427	495.670	507.322	518.886	532.924
548.211	565.190	585.230	611.560	642.955
676.287	773.752	815.380	826.614	830.225
829.402	829.251	829.606	826.571	815.313
798.841	780.729	762.982	746.369	731.856
719.154	708.939	698.905	688.952	680.241
671.864	663.297	653.173	644.471	636.534
628.317	619.321	608.705	595.505	579.029
551.713	570.457			

OUTPUT FROM OBJCON

OBJECTIVE FUNCTION	DIFFS	=	.83023E+03
	WO	=	.49855E+03
	DIFFS/WO	=	.16653E+01
CONSTRAINT 1	BLTKS	=	.44211E-02
CONSTRAINT 2	YMAX	=	.67629E+03
	YMIN	=	.43890E+03
	DIFFP	=	.15409E+01
CONSTRAINT 3	WMB(MBO-2,1)	=	.55171E+03
	WMB(MBO-2,2)	=	.49485E+03

Figure A.2.- Continued.

TECLSR = .71078E+00

OPTIMIZATION RESULTS

OBJECTIVE FUNCTION
GLOBAL LOCATION 1 FUNCTION VALUE .16653E+01

DESIGN VARIABLES

ID	D. V. NO.	GLOBAL VAR. NO.	LOWER BOUND	VALUE	UPPER BOUND
1	1	2	-.15000E+02	-.80297E+01	0.
2	2	3	.20000E+00	.20000E+00	.60000E+00
3	3	4	.20000E+00	.55000E+00	.55000E+00
4	4	5	.50000E+00	.73559E+00	.40000E+01
5	5	6	.30000E-01	.36263E-01	.10000E+00
6	6	7	.30000E-02	.39389E-02	.12000E-01

DESIGN CONSTRAINTS

ID	GLOBAL VAR. NO.	LOWER BOUND	VALUE	UPPER BOUND
1	8	0.	.44211E-02	.10000E+02
3	9	0.	.15409E+01	.16000E+01
5	10	-.10000E+01	.71078E+00	.10000E+01

Figure A.2.- Continued.

*** INPUT FOR BLADE ELEMENT PROGRAM ***

NO. BLADES	INLET RADIUS (L)	CONE ANGLE (DEG)	SOLIDITY	INLET BLADE ANGLE (DEG)	OUTLET BLADE ANGLE (DEG)	TRANS. LOC. /CHORD	MAX.TH. LOC. /CHORD	IN/OUT TURNING RATE	MAX. THICK. /CHORD	L.E. RAD. /CHORD	T.E. RAD. /CHORD
34	.45400	6.66400	2.25200	52.000	-8.0297	.20000	.55000	.73559	.03626	.00394	.00500

*** OUTPUT FROM BLADE ELEMENT PROGRAM ***

CHORD (L)	ELEMENT SETTING ANGLE (DEG)
.19322	23.5522

	* MERID. LOC. FROM L.E. CENT. **				** THETA LOC. FROM L.E. CENT. **				********* BLADE ANGLES ********* (WITH RESPECT TO LOCAL CONIC RAY)				SEGMENT LENGTHS	
	INLET (L)	OUTLET (L)	TRANS. (L)	MAX.TH. (L)	INLET (RAD)	OUTLET (RAD)	TRANS. (RAD)	MAX.TH. (RAD)	INLET (DEG)	OUTLET (DEG)	TRANS. (DEG)	MAX.TH. (DEG)	FIRST (L)	SECOND (L)
SUCT.	-.00061	.17564	.02675	.08501	.00101	.16727	.06970	.15112	52.848	-11.222	44.807	20.661	.04159	.16190
CENT.	0.00000	.17546	.02807	.08624	0.00000	.16528	.06653	.14405	52.000	-8.030	42.376	20.661	.04136	.15935
PRES.	.00059	.17538	.02940	.08748	-.00105	.16325	.06334	.13699	51.166	-4.794	39.883	20.661	.04119	.15704

	**** CONIC ANGLE COORD. - є **** ** (FROM LEADING EDGE CENT.) ***				********* BLADE ANGLES ********* (WITH RESPECT TO L.E. CENT. RAY)			
	INLET (DEG)	OUTLET (DEG)	TRANS. (DEG)	MAX.TH. (DEG)	INLET (DEG)	OUTLET (DEG)	TRANS. (DEG)	MAX.TH. (DEG)
SUCT.	.007	1.112	.463	1.005	52.855	-10.110	45.271	21.666
CENT.	0.000	1.099	.442	.958	52.000	-8.030	42.819	21.619
PRES.	-.007	1.085	.421	.911	51.159	-3.709	40.304	22.530

Figure A.2.- Continued.

*** OUTPUT THAT CAN BE PUNCHED FOR TSONIC INPUT ***

CHORD (L)	STGR (RAD)
.17718	.165276

***** SUCTION SURFACE *****

RI (L)	RO (L)	BETI (DEG)	BETO (DEG)
.00076	.00097	52.8551	-10.1097

ZMSP (L)	THSP (RAD)
.00015	.001012
.00644	.018624
.01569	.042414
.02688	.068304
.04209	.097782
.05884	.122866
.07687	.143206
.09593	.158534
.11571	.168663
.13592	.173491
.15219	.173520
.16637	.170770
.17641	.167241

***** PRESSURE SURFACE *****

RI (L)	RO (L)	BETI (DEG)	BETO (DEG)
.00076	.00097	51.1587	-3.7086

ZMSP (L)	THSP (RAD)
.00135	-.001051
.00775	.015605
.01729	.037720
.02898	.061165
.04470	.087489
.06153	.110221
.07931	.129167
.09786	.144173
.11700	.155129
.13656	.161966
.15237	.164450
.16624	.164451
.17614	.163216

Figure A.2.- Continued.

*** PLOT OF BLADE SURFACE IN THETA - M COORDINATES ***

Figure A.2.- Continued.

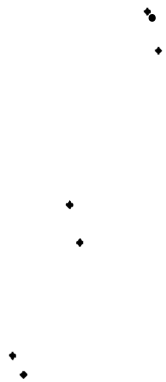


Figure A.2.- Continued.

```
--- INPUT FOR TSONIC BLADE-TO-BLADE FLOW SOLVER ---

  GAM          AR           TIP          RHOIP          WTFL                        OMEGA        ORF
1.400000     1716.480     599.7600     .3345860E-02   .5700000E-02  0.           0.           1.910000
  BETAI        BETAO        CHORDF       STGRF          FSMI         FSMO
48.20000     0.           .1771834     .1652756       0.            0.
  REDFAC       DENTOL       SSM1         SSM2
1.000000     .1000000E-02  0.           0.

MBI   MBO                  MM   NBBI   NBL   NRSP   MOPT   LOPT   LRVB
 24    56     0      0     72    15     34    13     0      1      0

BLADE SURFACE 1  --  UPPER SURFACE
  RI1          RO1          BETI1        BETO1          SPLNO1
.7610700E-03  .9660876E-03  52.85512     -10.10968      13.00000
  MSP1  ARRAY
.1544666E-03  .6437113E-02  .1568635E-01  .2687976E-01  .4208902E-01  .5883798E-01  .7687469E-01  .9592700E-01
.1157074      .1359176      .1521902      .1663708      .1764053
  THSP1  ARRAY
.1012280E-02  .1862413E-01  .4241444E-01  .6830414E-01  .9778183E-01  .1228661      .1432064      .1585341
.1686632      .1734907      .1735195      .1707699      .1672414

BLADE SURFACE 2  --  LOWER SURFACE
  RI2          RO2          BETI2        BETO2          SPLNO2
.7610700E-03  .9660876E-03  51.15866     -3.708649      13.00000
  MSP2  ARRAY
.1353915E-02  .7749708E-02  .1729088E-01  .2898442E-01  .4470456E-01  .6153388E-01  .7930961E-01  .9785967E-01
.1170045      .1365588      .1523700      .1662433      .1761365
  THSP2  ARRAY
-.1050920E-02  .1560516E-01  .3771963E-01  .6116454E-01  .8748856E-01  .1102210      .1291670      .1441733
.1551291      .1619659      .1644503      .1644505      .1632162

  MR  ARRAY
-.1300000     -.5000000E-02  .5000000E-02  .2003000E-01  .4003000E-01  .8003000E-01  .1000300      .1250300
.1500400      .1752600      .1852600      .2250000      .2700000
  RMSP  ARRAY
.4540000      .4540000      .4540000      .4540000      .4540000      .4540000      .4540000      .4540000
.4540000      .4540000      .4540000      .4540000      .4540000
  BESP  ARRAY
.5000000E-01  .5000000E-01  .4990000E-01  .4960000E-01  .4925000E-01  .4850000E-01  .4815000E-01  .4770000E-01
.4720000E-01  .4675000E-01  .4661000E-01  .4661000E-01  .4661000E-01
  PLOSS ARRAY
0.            0.            .1500000E-02  .5900000E-02  .1100000E-01  .2150000E-01  .3000000E-01  .3350000E-01
.4000000E-01  .4700000E-01  .4800000E-01  .4800000E-01  .4800000E-01

BLDAT  AANDK  ERSOR  STRFN  SLCRD  INTVL  SURVL
  0      0      0      0      0      0      1
```

Figure A.2.- Continued.

RELATIVE VELOCITY	MERIDIONAL VELOCITY	CRITICAL VELOCITY	REL. FLOW ANGLE
AT M = FSMI	AT M = FSMI	AT M = FSMI	AT UPSTREAM BDY.
745.90	497.17	1095.9	48.269
AT M = FSMO	AT M = FSMO	AT M = FSMO	AT DOWNSTREAM BDY.
497.92	497.92	1095.9	0.

FSMI = 0.
FSMO = .17718

CALCULATED PROGRAM CONSTANTS

PITCH	HT	HM1
.1847996	.1231997E-01	.5536980E-02

ITMIN	ITMAX
0	28

LAMBDA	DOWNSTREAM WHIRL (RVTHO)
252.4466	0.

REDUCED WEIGHT FLOW = .5700000E-02

NUMBER OF INTERIOR MESH POINTS = 1047

CALCULATED VELOCITY DIAGRAM INFORMATION

	IM	W	W/WCR	BETA
UPSTREAM BOUNDARY	1	745.10	.67988	48.269
LEADING EDGE	24	745.90	.68061	48.200
TRAILING EDGE	56	497.92	.45433	0.
DOWNSTREAM BOUNDARY	72	499.62	.45589	0.

Figure A.2.- Continued.

```
ITERATION  NO.   1
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1846        AT IM =  0,   IT =  7,   SURF = 1,   M =  .3571E-01
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1074
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  92

ITERATION  NO.   2
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .6432E-01   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4135E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .2615E-01
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  92

ITERATION  NO.   3
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .2959E-01   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4135E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .8037E-02
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  91

ITERATION  NO.   4
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1410E-01   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4135E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .2790E-02
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  55

ITERATION  NO.   5
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .6708E-02   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4135E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1041E-02
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  27

ITERATION  NO.   6
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .3285E-02   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4135E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .4003E-03
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =  17

ITERATION  NO.   7
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1422E-02   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4135E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .1766E-03
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =   1

ITERATION  NO.   8
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .7977E-03   AT IM =  0,   IT =  1,   SURF = 1,   M =  .4135E-02
AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .7674E-04
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS =   0

ITERATION  NO.   9
MAXIMUM RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =  .2963E-03   AT IM =  0,   IT = 15,   SURF = 2,   M =  .1739E-02
```

Figure A.2.- Continued.

AVERAGE RELATIVE CHANGE IN DENSITY AT BLADE SURFACE POINTS = .2880E-04
NUMBER OF UNCONVERGED BLADE SURFACE MESH POINTS = 0

SURFACE VELOCITIES BASED ON MERIDIONAL COMPONENTS - REDUCED WEIGHT FLOW

	BLADE SURFACE 1				BLADE SURFACE 2			
M	VELOCITY	ANGLE(DEG)	SURF. LENGTH	W/WCR	VELOCITY	ANGLE(DEG)	SURF. LENGTH	W/WCR
0.	0.	90.00	0.	0.	0.	-90.00	0.	0.
.5537E-02	797.1	51.11	.9198E-02	.7273	659.3	49.33	.7168E-02	.6016
.1107E-01	813.2	49.39	.1786E-01	.7420	631.7	47.02	.1547E-01	.5764
.1661E-01	825.6	47.83	.2623E-01	.7533	596.0	44.79	.2343E-01	.5438
.2215E-01	838.0	46.15	.3435E-01	.7646	571.5	42.65	.3109E-01	.5215
.2768E-01	845.8	44.13	.4221E-01	.7718	551.4	40.56	.3849E-01	.5031
.3322E-01	845.9	41.88	.4978E-01	.7719	534.4	38.52	.4567E-01	.4876
.3876E-01	839.4	39.46	.5708E-01	.7659	519.7	36.50	.5265E-01	.4742
.4430E-01	827.6	36.90	.6413E-01	.7551	507.1	34.53	.5946E-01	.4627
.4983E-01	809.1	34.39	.7094E-01	.7383	497.9	32.60	.6610E-01	.4544
.5537E-01	787.1	32.00	.7756E-01	.7182	488.4	30.71	.7261E-01	.4457
.6091E-01	767.5	29.79	.8401E-01	.7003	481.6	28.87	.7899E-01	.4394
.6644E-01	751.8	27.62	.9032E-01	.6860	474.2	27.07	.8526E-01	.4327
.7198E-01	735.6	25.47	.9651E-01	.6712	469.0	25.29	.9143E-01	.4279
.7752E-01	723.0	23.35	.1026	.6597	463.2	23.54	.9751E-01	.4227
.8305E-01	709.9	21.26	.1086	.6478	459.5	21.83	.1035	.4193
.8859E-01	700.7	19.21	.1145	.6394	454.9	20.14	.1094	.4151
.9413E-01	691.8	17.20	.1203	.6313	452.2	18.46	.1153	.4126
.9967E-01	681.5	15.23	.1261	.6219	449.7	16.82	.1211	.4103
.1052	673.9	13.28	.1318	.6149	446.0	15.19	.1269	.4069
.1107	666.0	11.34	.1375	.6077	444.0	13.58	.1326	.4052
.1163	658.1	9.43	.1431	.6005	442.6	12.00	.1383	.4038
.1218	650.4	7.53	.1487	.5935	441.7	10.43	.1439	.4030
.1274	642.7	5.65	.1543	.5865	441.7	8.85	.1495	.4030
.1329	632.3	3.79	.1598	.5770	441.3	7.28	.1551	.4027
.1384	624.6	1.94	.1654	.5699	443.7	5.71	.1607	.4049
.1440	615.9	.08	.1709	.5620	447.6	4.19	.1663	.4084
.1495	605.7	-1.78	.1764	.5527	452.8	2.74	.1718	.4132
.1550	593.0	-3.65	.1820	.5411	459.0	1.33	.1773	.4188
.1606	579.4	-5.46	.1875	.5287	468.2	-.33	.1829	.4272
.1661	561.2	-7.22	.1931	.5121	489.6	-2.27	.1884	.4467
.1716	549.8	-8.86	.1987	.5016	567.0	-3.66	.1940	.5173
.1772	0.	-90.00	.2045	0.	0.	90.00	.1995	0.

Figure A.2.- Continued.

SURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS
REDUCED WEIGHT FLOW

BLADE SURFACE 1

M	VELOCITY	ANGLE(DEG)	W/WCR
0.	493.5	90.00	.4503
.4135E-02	859.0	51.56	.7838
.8691E-02	801.6	50.11	.7314
.1349E-01	817.5	48.69	.7460
.1852E-01	829.3	47.29	.7567
.2384E-01	840.8	45.57	.7672
.2953E-01	847.4	43.40	.7732
.3571E-01	846.7	40.81	.7726
.4254E-01	838.8	37.72	.7654
.5026E-01	808.8	34.20	.7380
.5909E-01	783.6	30.50	.7150
.6940E-01	751.8	26.46	.6860
.8192E-01	726.3	21.69	.6627
.9848E-01	699.7	15.65	.6385
.1292	657.5	5.03	.5999
.1592	583.5	-5.01	.5324

BLADE SURFACE 2

M	VELOCITY	ANGLE(DEG)	W/WCR
.1739E-02	769.5	50.99	.7021
.6438E-02	661.6	48.95	.6037
.1149E-01	617.5	46.84	.5634
.1694E-01	587.2	44.66	.5358
.2283E-01	562.6	42.39	.5134
.2922E-01	541.6	39.99	.4942
.3620E-01	523.1	37.43	.4773
.4388E-01	507.4	34.68	.4630
.5243E-01	490.0	31.71	.4471
.6208E-01	475.2	28.48	.4336
.7321E-01	462.6	24.90	.4221
.8649E-01	453.6	20.78	.4139
.1035	444.2	15.70	.4053
.1300	433.7	8.09	.3958

Figure A.2.- Continued.

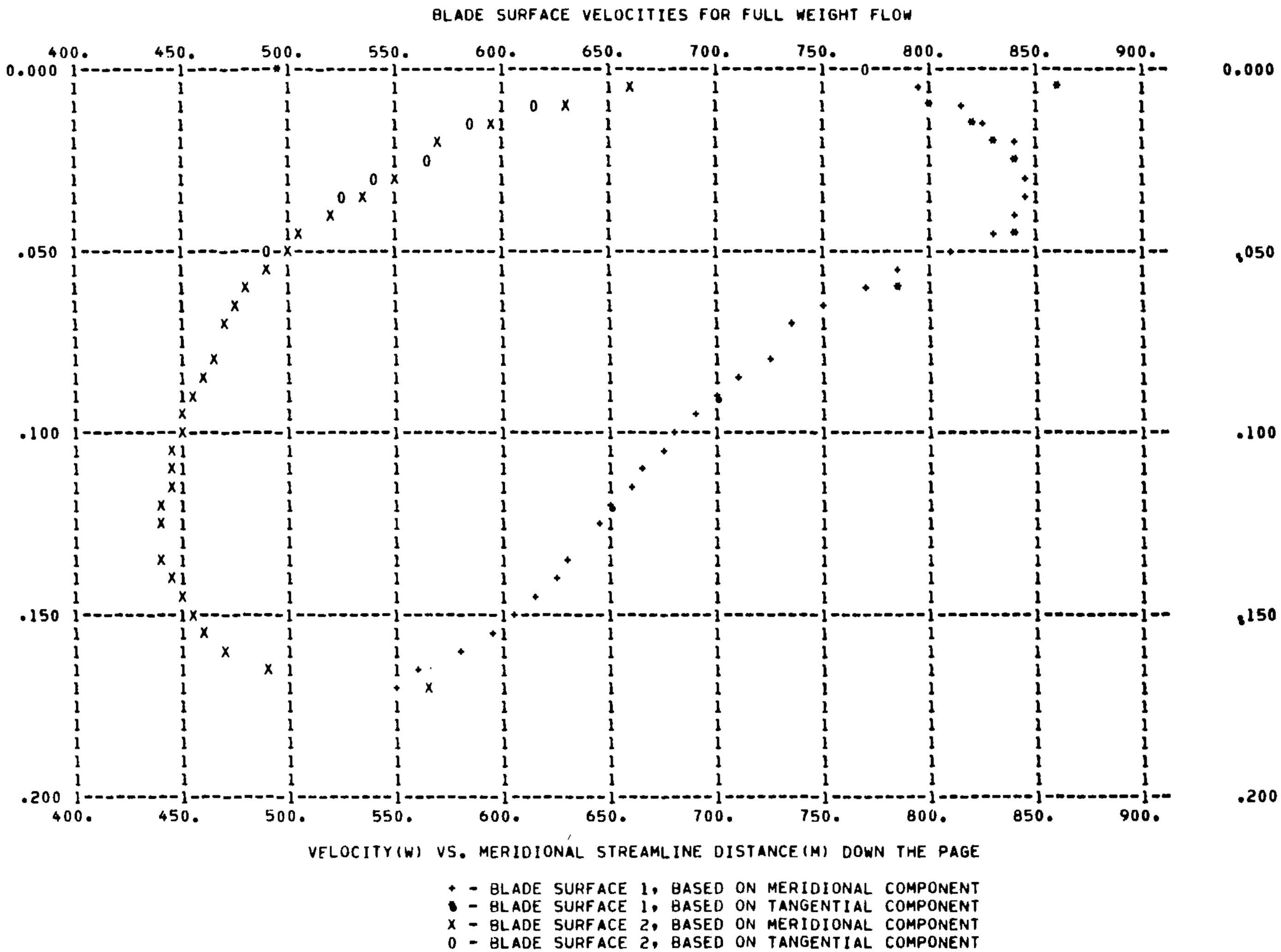


Figure A.2.- Continued.

BLADE SURFACE PRESSURES FOR FULL WEIGHT FLOW

IM	M	M/MC	P(1)/PT	P(2)/PT	CPT(1)	CPT(2)
24	0.	0.	.9994	.9994	-.6204E-03	-.6204E-03
25	.5537E-02	.3125E-01	.7228	.8030	-.2772	-.1970
26	.1107E-01	.6250E-01	.7118	.8167	-.2882	-.1833
27	.1661E-01	.9375E-01	.7029	.8338	-.2971	-.1662
28	.2215E-01	.1250	.6939	.8446	-.3061	-.1554
29	.2768E-01	.1563	.6879	.8530	-.3121	-.1470
30	.3322E-01	.1875	.6869	.8598	-.3131	-.1402
31	.3876E-01	.2188	.6901	.8655	-.3099	-.1345
32	.4430E-01	.2500	.6967	.8701	-.3033	-.1299
33	.4983E-01	.2813	.7072	.8732	-.2928	-.1268
34	.5537E-01	.3125	.7197	.8763	-.2803	-.1237
35	.6091E-01	.3438	.7305	.8781	-.2695	-.1219
36	.6644E-01	.3750	.7387	.8799	-.2613	-.1201
37	.7198E-01	.4063	.7469	.8806	-.2531	-.1194
38	.7752E-01	.4375	.7526	.8812	-.2474	-.1188
39	.8305E-01	.4688	.7582	.8806	-.2418	-.1194
40	.8859E-01	.5000	.7613	.8801	-.2387	-.1199
41	.9413E-01	.5313	.7643	.8790	-.2357	-.1210
42	.9967E-01	.5625	.7683	.8781	-.2317	-.1219
43	.1052	.5938	.7714	.8784	-.2286	-.1216
44	.1107	.6250	.7750	.8784	-.2250	-.1216
45	.1163	.6563	.7787	.8785	-.2213	-.1215
46	.1218	.6875	.7824	.8783	-.2176	-.1217
47	.1274	.7188	.7858	.8776	-.2142	-.1224
48	.1329	.7500	.7903	.8768	-.2097	-.1232
49	.1384	.7813	.7930	.8745	-.2070	-.1255
50	.1440	.8125	.7961	.8715	-.2039	-.1285
51	.1495	.8438	.7997	.8679	-.2003	-.1321
52	.1550	.8750	.8044	.8638	-.1956	-.1362
53	.1606	.9063	.8095	.8585	-.1905	-.1415
54	.1661	.9375	.8167	.8484	-.1833	-.1516
55	.1716	.9688	.8209	.8129	-.1791	-.1871
56	.1772	1.000	.9527	.9527	-.4728E-01	-.4728E-01

SURF 1		SURF 2	
XMSP	XTHSP	XMSP	XTHSP
.15447E-03	.10123E-02	.13539E-02	-.10509E-02
.64371E-02	.18624E-01	.77497E-02	.15605E-01
.15686E-01	.42414E-01	.17291E-01	.37720E-01
.26880E-01	.68304E-01	.28984E-01	.61165E-01
.42089E-01	.97782E-01	.44705E-01	.87489E-01
.58838E-01	.12287E+00	.61534E-01	.11022E+00
.76875E-01	.14321E+00	.79310E-01	.12917E+00
.95927E-01	.15853E+00	.97860E-01	.14417E+00
.11571E+00	.16866E+00	.11700E+00	.15513E+00
.13592E+00	.17349E+00	.13656E+00	.16197E+00
.15219E+00	.17352E+00	.15237E+00	.16445E+00
.16637E+00	.17077E+00	.16624E+00	.16445E+00
.17641E+00	.16724E+00	.17614E+00	.16322E+00

OUTPUT FROM OBJCON

Figure A.2.- Continued.

OBJECTIVE FUNCTION	DIFFS	=	.84594E+03
	WO	=	.49792E+03
	DIFFS/WO	=	.16990E+01
CONSTRAINT 1	BLTKS	=	.44211E-02
CONSTRAINT 2	YMAX	=	.65929E+03
	YMIN	=	.44134E+03
	DIFFP	=	.14938E+01
CONSTRAINT 3	WMB(MBO-2,1)	=	.56124E+03
	WMB(MBO-2,2)	=	.48960E+03
	TECLSR	=	.89556E+00

Figure A.2.- Continued.

```
FINAL OBJECTIVE COMPUTED BY TSONIC =  1.698965


FINAL CONSTRAINT VALUES
         ALTKS       DIFFP       TECLSR
      .44211E-02  .14938E+01  .89556E+00



X ARRAY
         .968750     .937500     .906250     .875000     .843750
         .812500     .781250     .750000     .718750     .687500
         .656250     .625000     .593750     .562500     .531250
         .500000     .468750     .437500     .406250     .375000
         .343750     .312500     .281250     .250000     .218750
         .187500     .156250     .125000     .093750     .062500
         .031250     .031250     .062500     .093750     .125000
         .156250     .187500     .218750     .250000     .281250
         .312500     .343750     .375000     .406250     .437500
         .468750     .500000     .531250     .562500     .593750
         .625000     .656250     .687500     .718750     .750000
         .781250     .812500     .843750     .875000     .906250
         .937500     .968750


LTRACE =  1

Y ARRAY BEFORE PERTRB CALL
         566.953     489.596     468.231     459.001     452.816
         447.620     443.724     441.341     441.690     441.695
         442.577     444.037     445.975     449.688     452.173
         454.893     459.487     463.246     468.972     474.208
         481.593     488.443     497.947     507.051     519.736
         534.387     551.407     571.509     596.015     631.744
         659.289     797.094     813.197     825.572     837.975
         845.829     845.941     839.362     827.555     809.126
         787.101     767.508     751.752     735.601     722.997
         709.946     700.748     691.827     681.542     673.904
         666.048     658.146     650.419     642.711     632.308
         624.559     615.941     605.686     592.951     579.399
         561.240     549.767
```

Figure A.2.- Continued.

```
*****************************************************
* COMPARISON OF AERODYNAMIC AND PERTURBATION SOLUTIONS *
*****************************************************


.....MACH NUMBER,
     VALUES OF PERTURBATION PARAMETERS,
     CRITICAL VALUE OF WM:

          M2 =  .1000

          Q2(1) =-8.0297     (  KOCR  )

          Q2(2) =  .2000     (    T   )

          Q2(3) =  .5500     (   ZM   )

          Q2(4) =  .7356     (    P   )

          Q2(5) =  .0363     (   TMX  )

          Q2(6) =  .0039     (  THLE  )

          WMCRIT =-66.8587


.....LOCATIONS OF MIN., MAX., AND CRITICAL PTS.
  (* DENOTES POINT ON LOWER SURFACE)

     PERTURBATION SOLN:

          MINIMUM AT X =  .7122*    (POINT NO.   9)
          MAXIMUM AT X =  .1194     (POINT NO.  34)

     AERODYNAMIC SOLN:

          MINIMUM AT X =  .7500*    (POINT NO.   8)
          MAXIMUM AT X =  .1875     (POINT NO.  37)


.....FINAL PRINTOUT AND GRAPHICAL DISPLAY OF WM

                                                                  H = MAXIMUM  VALUE OF WM =845.9406
                                                                  L = MINIMUM  VALUE OF WM =438.9010
                                                                  * = CRITICAL VALUE OF WM =-66.8587
                                                                  P = VALUE OF WM PREDICTED BY PERTURBATION SOLUTION
                                                                  A = VALUE OF WM IN AERODYNAMIC SOLUTION
                                                                  $ = AGREEMENT BETWEEN P AND A

PT  XBASE   WMBASE  XPERT   WMPERT  XAERO   WMAERO  WMPINT H------------------------------------------------------------------------L

 1   .9687585.9460    .9680549.7994    .9688566.9527551.1194                                                  A  P
 2   .9375500.5262    .9360492.2104    .9375489.5955494.8503                                                               PA
 3   .9062475.3838    .9041468.2692    .9063468.2308469.9154                                                                   $
 4   .8750463.5379    .8721457.5440    .8750459.0015458.5273                                                                     $
 5   .8437455.4531    .8401450.7826    .8438452.8158451.5575                                                                       $
 6   .8125448.9545    .8081445.5287    .8125447.6203446.2512                                                                        $
 7   .7813444.0337    .7761441.7684    .7813443.7239442.3717                                                                         $
```

Figure A.2.- Continued.

8	.7500440.6925	.7441439.5594	.7500441.3408439.9645
9	.7188438.6925	.7122438.7299	.7188441.6899438.9010
10	.6875437.8173	.6802439.4101	.6875441.6948439.2542
11	.6563439.1700	.6482441.5321	.6563442.5766440.9971
12	.6250439.8327	.6162443.2946	.6250444.0366442.8098
13	.5938441.0316	.5842445.5640	.5938445.9745444.8878
14	.5625442.5981	.5522448.7173	.5625449.6875447.7054
15	.5313445.5570	.5203452.6026	.5313452.1730451.2668
16	.5000447.4626	.4883455.9259	.5000454.8932454.7072
17	.4688449.7452	.4563460.4083	.4688459.4870458.6617
18	.4375453.7909	.4243465.7523	.4375463.2464463.5475
19	.4063457.2295	.3923471.6608	.4063468.9715469.0877
20	.3750462.5307	.3603475.4134	.3750474.2081473.6932
21	.3438467.1474	.3284488.9358	.3438481.5930482.4272
22	.3125473.4596	.2964502.5190	.3125488.4431495.6698
23	.2813479.5215	.2644512.6771	.2813497.9474507.3221
24	.2500487.9145	.2324526.4762	.2500507.0512518.8856
25	.2188496.1051	.2004541.5761	.2188519.7356532.9238
26	.1875508.1057	.1684557.9958	.1875534.3871548.2109
27	.1563522.6991	.1365576.8714	.1563551.4073565.1903
28	.1250540.8496	.1045600.2040	.1250571.5092585.2300
29	.0938564.0343	.0725634.0712	.0938596.0147611.5602
30	.0625599.0169	.0405662.5095	.0625631.7438642.9552
31	.0313632.8560	.0085710.1042	.0313659.2892676.2870
32	.0313836.7472	.0554812.4625	.0313797.0941773.7523
33	.0625848.1344	.0874825.6826	.0625813.1973815.3804
34	.0937851.3453	.1194830.3923	.0938825.5723826.6141
35	.1250846.1514	.1498829.4847	.1250837.9745830.2251
36	.1563834.5030	.1801829.0993	.1563845.8290829.4024
37	.1875820.3078	.2105829.7257	.1875845.9406829.2512
38	.2188805.2289	.2409829.2858	.2188839.3616829.6062
39	.2500788.7474	.2712820.2590	.2500827.5552826.5711
40	.2813770.7026	.3016805.2661	.2813809.1264815.3133
41	.3125754.1898	.3320787.3683	.3125787.1011798.8407
42	.3438739.2561	.3623770.2638	.3438767.5078780.7293
43	.3750723.4856	.3927752.8141	.3750751.7521762.9824
44	.4063710.8176	.4231738.3766	.4063735.6008746.3690
45	.4375697.5629	.4534724.6654	.4375722.9967731.8562
46	.4688688.7980	.4838713.7454	.4688709.9455719.1542
47	.5000680.8503	.5142704.7421	.5000700.7477708.9392
48	.5313671.6336	.5445694.3740	.5313691.8266698.9053
49	.5625665.2669	.5749685.2170	.5625681.5421688.9522
50	.5938658.7253	.6053677.2071	.5938673.9036680.2410
51	.6250651.9964	.6356668.9914	.6250666.0482671.8639
52	.6563644.9958	.6660660.6116	.6563658.1457663.2973
53	.6875635.7344	.6963650.1146	.6875650.4193653.1730
54	.7188629.2778	.7267642.4657	.7188642.7110644.4714
55	.7500622.6299	.7571634.7305	.7500632.3081636.5335
56	.7813615.5034	.7874626.6743	.7813624.5586628.3174
57	.8125607.3548	.8178617.7636	.8125615.9409619.3213
58	.8437597.2455	.8482607.1599	.8437605.6857608.7046
59	.8750584.0971	.8785593.9676	.8750592.9514595.5051
60	.9062567.1183	.9089577.5979	.9062579.3986579.0288
61	.9375548.2492	.9393550.1110	.9375561.2400551.7128
62	.9687533.5057	.9696571.0680	.9687549.7669570.4574

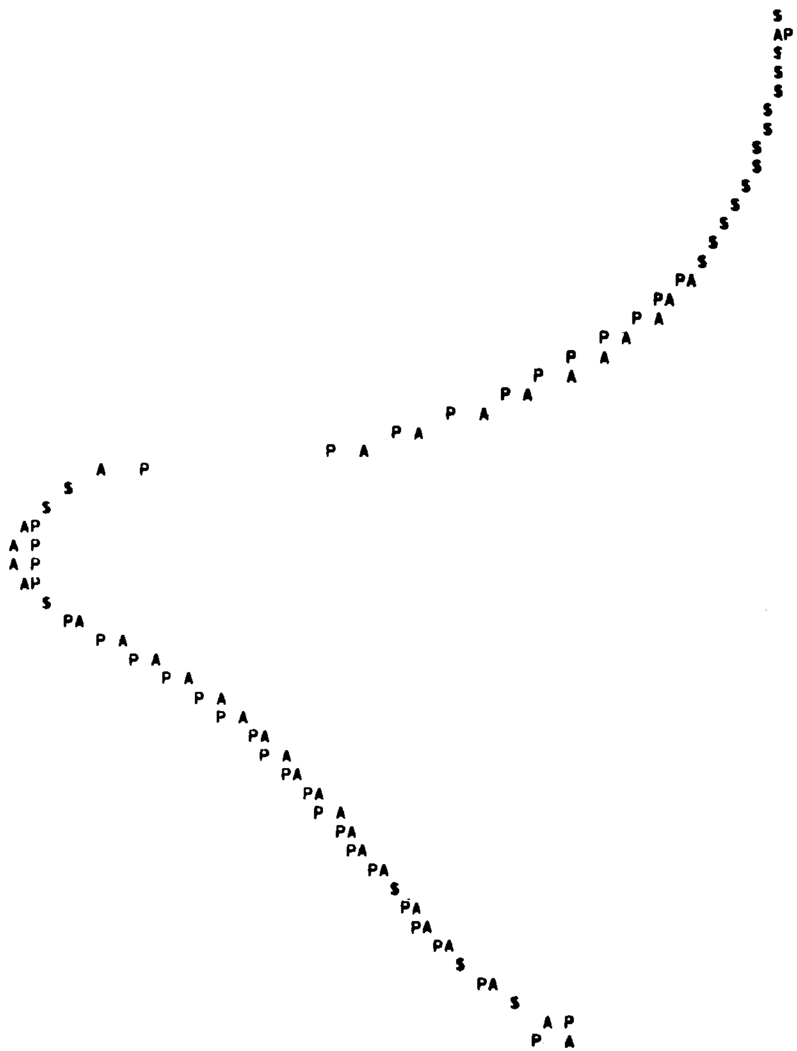


Figure A.2.- Concluded.

APPENDIX B

LISTING OF COMPUTER PROGRAM BLDOPT

```
      PROGRAM BLDOPT(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     *       TAPE1,TAPE7,TAPE9,TAPE10,TAPE11,TAPE12,TAPE13,TAPE14,
     *       TAPE20,TAPE40)
C
C  DRIVER PROGRAM FOR COPES OPTIMIZATION STUDY WITH OPTIONS
C  FOR PERTURBATION METHOD USE.
C  THE RELATIONSHIP AMONG PROGRAM COMPONENTS IS
C     DRIVER
C         - COPES
C              - ANALIZ
C                    - BLADE ROUTINES
C                    - TSONIC ROUTINES
C                    - PERTRB ROUTINE
C                    - OBJCON
C
C  PROGRAM OPTIONS
C     IOPT = 1  -  OPTIMIZATION USING TSONIC SOLUTIONS ONLY -
C                  PERTURBATION METHOD BYPASSED
C            2  -  OPTIMIZATION WITH PERTURBATION METHOD EMPLOYING
C                  USER-SPECIFIED CALIBRATION SOLUTION MATRIX
C            3  -  OPTIMIZATION WITH 1 CYCLE EMPLOYING TSONIC SOLUTIONS
C                  ONLY, FOLLOWED BY ITMAX CYCLES EMPLOYING
C                  PERTURBATION METHOD
C
      COMMON/CALB/ ICALB,DVCALB(6),PSTEP,IOPT,ITMAX3,VZERO(6)
      COMMON/CNMN1/DUM(12),NDV,DUM1(6),ITMAX,DUM2(7)
      COMMON/COPES2/ RA(5000),IA(1000)
      COMMON/COPES3/DUM3(16),LOCR(25),LOCI(25),DUM4(19)
      COMMON/GLOBCM/ OBJ,V(6),C(5),EXTRA(1488)
C
      IFLAG=1
      CALL COPES(IFLAG)
      IF(IOPT.LT.3) STOP
C
C  IOPT=3     CALCULATE CALIBRATION STEPSIZES, THEN PERFORM
C             OPTIMIZATION WITH PERTURBATION SOLUTIONS
C
      NVLB=LOCR(2)
      NVUB=LOCR(3)
      CALL STEP(NDV,V,VZERO,DVCALB,RA(NVLB),RA(NVUB))
      ICALB=1
      ITMAX=ITMAX3
      DO 10 I=1,NDV
   10 RA(I)=0.0
      IFLAG=2
      CALL COPES(IFLAG)
      STOP
      END
      SUBROUTINE STEP(NDV,V,VZERO,DVCALB,VLB,VUB)
C
C  ROUTINE TO CALCULATE CALIBRATION STEPSIZES FOR IOPT=3
C  BASED ON DESIGN RESULTS AFTER 1 OPTIMIZATION CYCLE
C  EMPLOYING TSONIC SOLUTIONS ONLY.
C
      DIMENSION V(1),VZERO(1),DVCALB(1),VLB(1),VUB(1)
      DO 20 I=1,NDV
      P=V(I)-VZERO(I)
      DVCALB(I)=V(I)+0.5*P
      IF(DVCALB(I).LT.VLB(I)) DVCALB(I)=VLB(I)
      IF(DVCALB(I).GT.VUB(I)) DVCALB(I)=VUB(I)
      IF(ABS(V(I)-VLB(I)).LT.1.E-6)
     *           DVCALB(I)=0.9*(VLB(I)-VZERO(I))+VZERO(I)
      IF(ABS(V(I)-VUB(I)).LT.1.E-6)
     *           DVCALB(I)=0.9*(VUB(I)-VZERO(I))+VZERO(I)
   20 CONTINUE
C
      RETURN
      END
      SUBROUTINE ANALIZ(ICALC)
C
C  INITIALIZES, CALCULATES, AND OUTPUTS ALL DESIGN
C  VARIABLES, OBJECTIVE FUNCTION, AND CONSTRAINTS FOR
C  OPTIMIZATION WITH COPES DRIVER.
C
      COMMON/GLOBCM/ OBJ,V(6),C(5),EXTRA(1488)
      COMMON /PERT/ Q0(10),Q1,Q2(10),XM0,XM1,XM2,NPTS,KPARAM,NPARAM,
     *              LTRACE
      COMMON /INPUTB/ ALP,KICR,KOCR,NB,P,R,SOLID,T,THLE,THTE,TMX,ZM
      COMMON /MBIMBO/ MBI,MBO
      COMMON/INPUTA/ XMSP(50,2),XTHSP(50,2)
      COMMON /CALB/ ICALB,DVCALB(6),PSTEP,IOPT,ITMAX3,VZERO(6)
      DIMENSION X(200),Y(200),VV(8),IDV(6),VNAME(6)
      REAL KICR,KOCR
      LOGICAL NONLIN
      DATA XM0/0.1/, XM1/0.1/, XM2/0.1/
      IF (ICALC .GT. 1) GO TO 100
C
C  ICALC = 1       READ INPUT AND INITIALIZE PARAMETERS
C
      READ(5,1025) COMENT
      READ(5,1005) IOPT,NDV,NCN,ITMAX3
      NPARAM=NDV
      READ(5,1010) NB,R,ALP,SOLID
      READ(5,1020) (VV(I),I=1,8)
      KICR=VV(1)
      KOCR=VV(2)
      T   =VV(3)
      ZM  =VV(4)
      P   =VV(5)
      TMX =VV(6)
      THLE=VV(7)
      THTE=VV(8)
      READ(5,1025) (VNAME(I),I=1,NDV)
      READ(5,1005) (IDV(I),I=1,NDV)
      DO 5 I=1,NDV
      J=IDV(I)
      V(I)=VV(J)
    5 VZERO(I)=VV(J)
      READ(5,1010) ICALB,PSTEP
      IF(ICALB.EQ.0) GO TO 10
      READ(5,1020) (DVCALB(I),I=1,NDV)
   10 CONTINUE
C
      WRITE(6,1000)
      WRITE(6,1055) IOPT,NDV,NCN,ITMAX3
      WRITE(6,1060) NB,R,ALP,SOLID
      WRITE(6,1065) (VV(I),I=1,8)
      WRITE(6,1070) (VNAME(I),I=1,NDV)
      WRITE(6,1072) (V(I),I=1,NDV)
      WRITE(6,1075) ICALB,PSTEP
```

```
      IF(ICALB.EQ.0) GO TO 14
      WRITE(6,1080) (I,I=1,NDV)
      WRITE(6,1085) (DVCALB(I),I=1,NDV)
   14 CONTINUE
C
      ICALL=1
      JCALL=0
      KPARAM=0
      LTRACE=0
      NONLIN=.T.
      IF(IOPT.EQ.2) NONLIN=.F.
      READ(5,1025) COMENT
      CALL TSONIC
      NPTS=2*(MBO-MBI-1)
      IF(IOPT.EQ.1) GO TO 15
      READ(5,1025) COMENT
      CALL PERTRB(ICALL,X,Y)
   15 CONTINUE
      ICALL=2
      RETURN
C
C  ICALC = 2        PERFORM OPTIMIZATION CALCULATION
C
  100 IF(ICALC.GT.2) GO TO 200
   20 CALL DESVAR(VV,V,IDV,NDV,ICALL,KPARAM)
      WRITE(6,2000) NB,R,ALP,SOLID,KICR,KOCR,T,ZM,P,TMX,THLE,THTE
      CALL BLADE(0)
      IF(.NOT.NONLIN) GO TO 22
C
C        EMPLOY TSONIC SOLUTIONS,  IOPT=1 OR 3
C
      CALL TSONIC
      CALL XY(X,Y)
      WRITE(6,2005) (X(I),I=1,NPTS)
      WRITE(6,2010) (Y(I),I=1,NPTS)
      CALL OBJCON(X,Y,OBJ,C)
      RETURN
C
C       EMPLOY BASE/CALIBRATION/PERTURBATION SOLUTIONS, IOPT=2 OR 3
C
   22 CONTINUE
      IF(ICALL.EQ.4) GO TO 24
      CALL TSONIC
      CALL XY(X,Y)
   24 CALL PERVAR(Q0,Q1,Q2,V,ICALL,KPARAM)
      CALL PERTRB(ICALL,X,Y)
      IF(ICALL.EQ.3) GO TO 26
      WRITE(6,2005) (X(I),I=1,NPTS)
      WRITE(6,2010) (Y(I),I=1,NPTS)
      CALL OBJCON(X,Y,OBJ,C)
      IF(ICALL.EQ.4) RETURN
      ICALL=3
   26 CONTINUE
      KPARAM=KPARAM+1
      IF(KPARAM.LE.NPARAM) GO TO 20
      ICALL=4
      RETURN
C
C  ICALC  =  3      CONMIN TERMINATION AND FINAL SOLUTION DETERMINATION
C
  200 CONTINUE
      JCALL=JCALL+1
      IF(IOPT.NE.3 .OR. JCALL.EQ.2) GO TO 30
      NONLIN=.FALSE.
      RETURN
   30 CONTINUE
      LTRACE=1
      CALL DESVAR(VV,V,IDV,NDV,ICALL,KPARAM)
      CALL BLADE(1)
      CALL TSONIC
      CALL XY(X,Y)
      WRITE(6,1050) (XMSP(K,1),XTHSP(K,1),XMSP(K,2),XTHSP(K,2),K=1,13)
      CALL OBJCON(X,Y,OBJ,C)
      WRITE(6,1030) OBJ
      WRITE(6,1035) (C(I),I=1,NCN)
      IF(IOPT.EQ.1) RETURN
      WRITE(6,2005) (X(I),I=1,NPTS)
      WRITE(6,1040) LTRACE,(Y(I),I=1,NPTS)
      CALL PERVAR(Q0,Q1,Q2,V,ICALL,KPARAM)
      CALL PERTRB(ICALL,X,Y)
      RETURN
C
C
 1000 FORMAT(1H1,35X,43H--- INPUT FOR CONTROL SUBROUTINE ANALIZ ---)
 1005 FORMAT(10I5)
 1010 FORMAT (I10,3F10.6)
 1020 FORMAT(8F10.0)
 1025 FORMAT(6A10)
 1030 FORMAT (37H1FINAL OBJECTIVE COMPUTED BY TSONIC =,F10.6//)
 1035 FORMAT(1H0,23HFINAL CONSTRAINT VALUES/10X,5HBLTKS,7X,5HDIFFP,
     *       6X,6HTECLSR/5X,4E12.5)
 1040 FORMAT(////* LTRACE =*,I3//* Y ARRAY BEFORE PERTRB CALL*/
     *       (5X,5F12.3))
 1045 FORMAT(////* LTRACE =*,I3//* Y ARRAY AFTER PERTRB CALL*/
     *       (5X,5F12.3))
 1050 FORMAT(////12X,*SURF 1*,29X,*SURF 2*/
     *       5X,*XMSP*,11X,*XTHSP*,15X,*XMSP*,11X,*XTHSP*/
     *       (2E15.5,5X,2E15.5))
 1055 FORMAT(//1H0,4X,18HCONTROL PARAMETERS/
     *       1H0,9X,6HIOPT =,I3,10X,5HNDV =,I3,10X,5HNCN =,I3,
     *       10X,8HITMAX3 =,I3)
 1060 FORMAT(//1H0,4X,25HCONSTANT BLADE PARAMETERS/
     *       1H0,9X,2HNB,7X,1HR,7X,3HALP,7X,5HSOLID/I12,3F10.4)
 1065 FORMAT(//1H0,4X,25HVARIABLE BLADE PARAMETERS/
     *       1H0,9X,4HKICR,6X,4HKOCR,8X,1HT,8X,2HZM,8X,1HP,9X,3HTMX,
     *       6X,4HTHLE,6X,4HTHTE/5X,8F10.4)
 1070 FORMAT(//1H0,4X,23HACTIVE DESIGN VARIABLES/1H0,9X,6A10)
 1072 FORMAT(5X,6F10.4)
 1075 FORMAT(//1H0,4X,7HICALB =,I2,6X,7HPSTEP =,F6.3)
 1080 FORMAT(1H0,4X,6HDVCALB/6I12)
 1085 FORMAT(3X,6F12.4)
 2000 FORMAT(1H1,// 46X,39H*** INPUT FOR BLADE ELEMENT PROGRAM ***   //
     */ 49X,5HINLET,4X,6HOUTLET,4X,6HTRANS.,4X,7HMAX.TH.,3X,6HIN/OUT,5X,
     * 4HMAX.,6X,4HL.E.,6X,4HT.E. / 10X,3HNO.,6X,5HINLET,5X,4HCONE,4X,
     * 8HSOLIDITY,4X,5HBLADE,5X,5HBLADE,5X,4HLOC.,6X,4HLOC.,5X,7HTURNING
     *,3X,6HTHICK.,5X,4HRAD.,6X,4HRAD. / 8X,6HBLADES,4X,6HRADIUS,5X,
     * 5HANGLE,15X,5HANGLE,5X,5HANGLE,4X,6H/CHORD,4X,6H/CHORD,5X,4HRATE,
     * 5X,6H/CHORD,4X,6H/CHORD,4X,6H/CHORD / 20X,3H(L),6X,5H(DEG),15X,
     * 5H(DEG),5X,5H(DEG) // 9X,I3,3X,3F10.5,F9.3,F10.4,6F10.5)
 2005 FORMAT(////* X ARRAY*/(5X,5F12.6))
```

```
 2010 FORMAT(1H0,* Y ARRAY*/(5X,5F12.3))                         ANALI174
C                                                                ANALI175
      END                                                        ANALI176
      SUBROUTINE DESVAR(VV,V,IDV,NDV,ICALL,KPARAM)               DESVAR 2
C                                                                DESVAR 3
C  IDENTIFIES DESIGN VARIABLES                                   DESVAR 4
C                                                                DESVAR 5
      REAL KICR,KOCR                                             DESVAR 6
      COMMON /CALB/ ICALB,DVCALB(6),PSTEP,IOPT,ITMAX3,VZERO(6)   DESVAR 7
      COMMON /INPUTB/ ALP,KICR,KOCR,NB,P,R,SOLID,T,THLE,THTE,TMX,ZM DESVAR 8
      DIMENSION VV(12),V(1),IDV(1)                               DESVAR 9
      DO 10 I=1,NDV                                              DESVAR10
      J=IDV(I)                                                   DESVAR11
      VV(J)=V(I)                                                 DESVAR12
   10 CONTINUE                                                   DESVAR13
      IF(ICALL.NE.3) GO TO 20                                    DESVAR14
      K=IDV(KPARAM)                                              DESVAR15
      IF(ICALB.EQ.0) VV(K)=(1.0+PSTEP)*VV(K)                     DESVAR16

      IF(ICALB.EQ.0) VV(K)=(1.0+PSTEP)*VV(K)                     DESVAR16
      IF(ICALB.GT.0) VV(K)=DVCALB(KPARAM)                        DESVAR17
   20 CONTINUE                                                   DESVAR18
      KICR=VV(1)                                                 DESVAR19
      KOCR=VV(2)                                                 DESVAR20
      T   =VV(3)                                                 DESVAR21
      ZM  =VV(4)                                                 DESVAR22
      P   =VV(5)                                                 DESVAR23
      TMX =VV(6)                                                 DESVAR24
      THLE=VV(7)                                                 DESVAR25
      THTE=VV(8)                                                 DESVAR26
      RETURN                                                     DESVAR27
      END                                                        DESVAR28
      SUBROUTINE PERVAR(Q0,Q1,Q2,V,ICALL,KPARAM)                 PERVAR 2
C                                                                PERVAR 3
C  SETS VALUES OF DESIGN VARIABLES FOR USE IN                    PERVAR 4
C  PERTURBATION SOLUTION CALCULATION.                            PERVAR 5
C                                                                PERVAR 6
      DIMENSION Q0(1),Q2(1),V(1)                                 PERVAR 7
      COMMON /CALB/ ICALB,DVCALB(6),PSTEP,IOPT,ITMAX3,VZERO(6)   PERVAR 8
      IF (ICALL .GT. 2) GO TO 20                                 PERVAR 9
      DO 10 I=1,6                                                PERVAR10
   10 Q0(I)=V(I)                                                 PERVAR11
      RETURN                                                     PERVAR12
   20 IF (ICALL .GT. 3) GO TO 30                                 PERVAR13
      IF(ICALB.EQ.0) Q1=(1. + PSTEP)*V(KPARAM)                   PERVAR14
      IF(ICALB.GT.0) Q1=DVCALB(KPARAM)                           PERVAR15
      RETURN                                                     PERVAR16
   30 DO 40 I=1,6                                                PERVAR17
   4  Q2(I)=5(I)                                                 PER5AR18
      RETURN                                                     PERVAR19
   40 Q2(I)=V(I)                                                 PERVAR18
      RETURN                                                     PERVAR19
      END                                                        PERVAR20
      SUBROUTINE XY(X,Y)                                         XY     2
C                                                                XY     3
C  DETERMINES SURFACE VELOCITY AND SURFACE COORDINATE ARRAYS     XY     4
C                                                                XY     5
      COMMON /VARCOM/ RDUM(400),WMB(100,2),SDUM(400),IDUM(100)   XY     6
      COMMON /MVNORM/ XMVN(100)                                  XY     7
      COMMON /MBIMBO/ MBI,MBO                                    XY     8
      DIMENSION X(200),Y(200)                                    XY     9
      ND2=MBO-MBI-1                                              XY    10
      DO 10 I=1,ND2                                              XY    11
      X(I)=XMVN(MBO-I)                                           XY    12
      X(ND2+I)=XMVN(MBI+I)                                       XY    13
      Y(I)=WMB(MBO-I,2)                                          XY    14
   10 Y(ND2+I)=WMB(MBI+I,1)                                      XY    15
      RETURN                                                     XY    16
      END                                                        XY    17
      SUBROUTINE OBJCON(X,Y,DIFFS,C)                             OBJCON 2
      REAL KICR,KOCR                                             OBJCON 3
      COMMON/INPUTA/ XMSP(50,2),XTHSP(50,2)                      OBJCON 4
      COMMON /INPUTB/ALP,KICR,KOCR,NB,P,R,SOLID,T,THLE,THTE,TMX,ZM OBJCON 5
      COMMON /MBIMBO/ MBI,MBO                                    OBJCON 6
      COMMON/WIWO/WI,WO                                          OBJCON 7
      COMMON/VARCOM/ADUM(400),WMB(100,2),BDUM(400),IDUM(100)     OBJCON 8
      DIMENSION X(200),Y(200),C(5)                               OBJCON 9
C                                                                OBJCON10
C  CALCULATION OF OBJECTIVE FUNCTION                             OBJCON11
C                                                                OBJCON12
      ND2=MBO-MBI-1                                              OBJCON13
      DIFFS=Y(ND2+1)                                             OBJCON14
      ND21=ND2-1                                                 OBJCON15
      DO 10 I=1,ND21                                             OBJCON16
   10 IF(Y(ND2+I).GT.DIFFS) DIFFS=Y(ND2+I)                       OBJCON17
      WRITE(6,1000) DIFFS,WO                                     OBJCON18
      DIFFS=DIFFS/WO                                             OBJCON19
      WRITE(6,1010) DIFFS                                        OBJCON20
C                                                                OBJCON21
C  CONSTRAINT NO. 1  -  BLADE THICKNESS                          OBJCON22
C                                                                OBJCON23
      BCHORD=2.*3.1415927*R*SOLID/FLOAT(NB)                      OBJCON24
      TEDIAM=2.*THTE*BCHORD                                      OBJCON25
      DTMIN=1.E10                                                OBJCON26
      DO 20 I=2,12                                               OBJCON27
      DRTH=(XTHSP(I,1)-XTHSP(I,2))*R                             OBJCON28
      DZ=XMSP(I,1)-XMSP(I,2)                                     OBJCON29
      DT=SQRT(DRTH*DRTH+DZ*DZ)                                   OBJCON30
      TKODTE=(DT-TEDIAM)/TEDIAM                                  OBJCON31
   20 IF(TKODTE.LT.DTMIN) DTMIN=TKODTE                           OBJCON32
      BLTKS=DTMIN                                                OBJCON33
      WRITE(6,1020) BLTKS                                        OBJCON34
      C(1)=BLTKS                                                 OBJCON35
C                                                                OBJCON36
C  CONSTRAINT NO. 2  -  PRESSURE SURFACE DIFFUSION               OBJCON37
C                                                                OBJCON38
      A=ND2/2                                                    OBJCON39
      B=2.*ND2/3                                                 OBJCON40
      N12=INT(A)                                                 OBJCON41
      N23=INT(B)                                                 OBJCON42
      YMAX=Y(ND2)                                                OBJCON43
      YMIN=YMAX                                                  OBJCON44
      DO 30 I=N12,ND2                                            OBJCON45
   30 IF(Y(I).GT.YMAX) YMAX=Y(I)                                 OBJCON46
      DO 40 I=2,N23                                              OBJCON47
   40 IF(Y(I).LT.YMIN) YMIN=Y(I)                                 OBJCON48
      DIFFP=YMAX/YMIN                                            OBJCON49
      WRITE(6,1030) YMAX,YMIN,DIFFP                              OBJCON50
      C(2)=DIFFP                                                 OBJCON51
C                                                                OBJCON52
```

```
C   CONSTRAINT NO. 3  -  TRAILING EDGE CLOSURE
C
      N1=2*(MBO-MBI)-3
      TECLSR=(Y(N1)-Y(2))/80.
      WRITE (6,1040) Y(N1),Y(2),TECLSR
      C(3)=TECLSR
C
      RETURN
 1000 FORMAT(////* OUTPUT FROM OBJCON*/1H0,3X,
     *     *OBJECTIVE FUNCTION*,3X,*DIFFS    =*,E12.5/
     *     25X,*WO       =*,E12.5)
 1010 FORMAT (25X,*DIFFS/WO =*,E12.5)
 1020 FORMAT(1H0,3X,*CONSTRAINT 1*, 9X,*BLTKS     =*,E12.5)
 1030 FORMAT(1H0,3X,*CONSTRAINT 2*, 9X,*YMAX     =*,E12.5/
     *      25X,*YMIN     =*,E12.5/
     *      25X,*DIFFP    =*,E12.5)
 1040 FORMAT(1H0,3X,*CONSTRAINT 3*, 5X,*WMB(MBO-2,1) =*,E12.5/
     *      21X,*WMB(MBO-2,2) =*,E12.5/
     *      25X,*TECLSR   =*,E12.5)
      END
      SUBROUTINE PERTRB (ICALL,X,Y)
C
C*********************** SUBROUTINE PERTURB **************************
C
C     ICALL=1 ... READ INPUT AND PRINT CONTROL PARAMETERS.
C
C     ICALL=2 ... PERFORM CALCULATIONS ON BASE SOLUTION.
C
C     ICALL=3 ... PERFORM CALCULATIONS ON CALIBRATION SOLUTION.
C
C     ICALL=4 ... RETURN PERTURBATION SOLUTION TO CALLING
C                 SUBROUTINE.
C
C**********************************************************************
C
      DIMENSION XLOC0(6),XLOC1(6),XLOC2(6),XLOC3(6),XFIX0(8),XFIX1(8)
      DIMENSION LCR0(4),LCR1(4),LCR2(4),LCR3(4),ISEQ0(8),ISEQ1(8)
      DIMENSION HEAD0(5),HEAD1(5),HEAD2(5),HEAD3(5)
      DIMENSION XOUT(8),DEL1(10),ORD(10),X(200),Y(200)
      DIMENSION FLAG(8),STRING(100),STRUNI(100)
      REAL M0,M1,M2
      COMMON /COEFF/ C(10,7),D(10,7),DELX(200)
      COMMON /HEAD/ TITLE(8),JOBKEY
      COMMON /PARAM/ PARNAM(10),LSELCT(6),LUNIT,LPLOT,NSELCT,A,B,VNAM(2)
      COMMON /PERT/ Q0(10),Q1,Q2(10),M0,M1,M2,N,KPARAM,NPARAM,LTRACE
      COMMON /MINMAX/ YMIN,YMAX,YCR2
      COMMON /SAVE/ XCSAVE(10,200),YCSAVE(10,200)
      COMMON /XY/ XBASE(200),XCALB(200),XPERT(200),XAERO(200),
     % XUNIT(200),YBASE(200),YCALB(200),YPERT(200),YAERO(200),
     % YINTP(200),YPRTI(200),DUMMY(200)
      COMMON /XUYU/ XUSAVE(10,200),YUNIT(10,200)
      LEVEL 2, XCSAVE,YCSAVE,XUSAVE,YUNIT,C,D,DELX
      LEVEL 2, XBASE,XCALB,XPERT,XAERO,XUNIT,YBASE,YCALB,YPERT,YAERO,
     % YINTP,YPRTI,DUMMY
      DATA LTERM/0/, LCORR/0/
      DATA HEAD0 /4HBASE,4H SOL,4HUTIO,4HN:  ,4H    /,
     %     HEAD1 /4HCALI,4HBRAT,4HION ,4HSOLN,4H:   /,
     %     HEAD2 /4HPERT,4HURBA,4HTION,4H SOL,4HN:  /,
     %     HEAD3 /4HAERO,4HDYNA,4HMIC ,4HSOLN,4H:   /
      DATA ORD /5H 1ST ,5H 2ND ,5H 3RD ,5H 4TH ,
     %          5H 5TH ,5H 6TH ,5H 7TH ,5H 8TH ,5H 9TH ,5H10TH /
C
C**********************************************************************
C
C     USER-SUPPLIED STATEMENT FUNCTION YCRIT(Z) DETERMINES CRITICAL
C     VALUES OF FLOW VARIABLE YCRIT AS FUNCTION OF FLOW PARAMETER Z.
C     IGRAD (+1 OR -1) IS THE USER-SUPPLIED ALGEBRAIC SIGN OF DYCRIT/DX
C     USED IN LOCATING THE CRITICAL POINT.
C
C     IN THE PRESENT VERSION OF THE CODE, YCRIT REPRESENTS THE FULL-
C     POTENTIAL CRITICAL PRESSURE COEFFICIENT FOR AIR (GAMMA = 1.4), Z
C     IS THE FREE STREAM MACH NUMBER, AND IGRAD CORRESPONDS TO POSITIVE
C     PRESSURE GRADIENT (+1).
C
      YCRIT(Z)=2.0*(((2.0+0.4*Z**2)/2.4)**(1.4/0.4)-1.0)/(1.4*Z**2)
      IGRAD=1
C
C**********************************************************************
      GO TO (501,502,503,504),ICALL
C
C.....INPUT CONTROL, GEOMETRY, AND STRAINING PARAMETERS.
C
  501 CALL INPUT (NPARAM)
C
C.....WRITE TITLE AND INPUT PARAMETERS.
C
      WRITE(6,1005)
      WRITE (6,1000) TITLE
      WRITE (6,1010) N,A,B,NPARAM
C
      NFIX=NSELCT+2
      NSEG=NFIX-1
C
C.....PRINT INFORMATION REGARDING STRAINING TO BE USED.
C
      WRITE (6,1020) NFIX
      WRITE (6,1040)
      DO 20 I=1,NSELCT
      IF (LSELCT(I) .EQ. 1) WRITE (6,1050) VNAM
      IF (LSELCT(I) .EQ. 2) WRITE (6,1060) VNAM
      IF (LSELCT(I) .LE. 2) GO TO 20
      LCORR=1
      LPR=LSELCT(I)-2
      WRITE (6,1070) VNAM,ORD(LPR)
   20 CONTINUE
      RETURN
C
C.....BEGIN CALCULATIONS ON BASE SOLUTION.
C
  502 YCRO=YCRIT(M0)
      WRITE (6,1080) HEAD0
      WRITE (6,1090) VNAM
      WRITE (6,1100) M0
      IF (NPARAM .EQ. 1) WRITE (6,1110) Q0(1),PARNAM(1)
      IF (NPARAM .GT. 1) WRITE (6,1120) (K,Q0(K),PARNAM(K),K=1,NPARAM)
      WRITE (6,1130) VNAM,YCRO
C
C.....NORMALIZE X COORDINATES AND LOCATE MINIMUM, MAXIMUM, AND CRITICAL
C     POINTS FOR BASE SOLUTION.
C
```

```
      CALL COPY12 (N,X,XBASE)
      CALL COPY12 (N,Y,YBASE)
      CALL SCALE (N,XBASE,1,A,B)
      CALL LOCATE (N,XBASE,YBASE,YCR0,IGRAD,LMN0,LMX0,NCR0,LCR0,XLOC0)
      YBCMIN=YBASE(LMN0)
      YBCMAX=YBASE(LMX0)
      WRITE (6,1140)
      WRITE (6,1150)
      CALL UPLOW (A,B,XLOC0,6,NCR0+2,XOUT,FLAG)
      WRITE (6,1160) XOUT(1),FLAG(1),LMN0,XOUT(2),FLAG(2),LMX0
      IF (NCR0 .GT. 0) WRITE (6,1170) NCR0,
     %      (ORD(I),XOUT(I+2),FLAG(I+2),LCR0(I),I=1,NCR0)
C
C.....LOAD SELECTED STRAINING POINTS INTO FIXED-POINT ARRAY FOR BASE
C     SOLUTION.
C
      XFIX0(1)=0.0
      XFIX0(NFIX)=1.0
      DO 50 I=1,NSELCT
      XFIX0(I+1)=XLOC0(LSELCT(I))
   50 CONTINUE
C
C.....ARRANGE SELECTED FIXED POINTS IN A MONOTONE SEQUENCE.
C
      CALL SORT (NFIX,XFIX0,ISEQ0)
      WRITE (6,1200)
      WRITE (6,1150)
      CALL UPLOW (A,B,XFIX0,8,NFIX,XOUT,FLAG)
      WRITE (6,1210) (I,XOUT(I),FLAG(I),I=1,NFIX)
      RETURN
C
C.....BEGIN CALCULATIONS ON CALIBRATION SOLUTIONS.
C
  503 K=KPARAM
      YCR1=YCRIT(M1)
      DEL1(K)=Q1-Q0(K)
      CALL COPY12 (N,X,XCALB)
      CALL COPY12 (N,Y,YCALB)
      CALL COPYVA (1,N,K,XCALB,XCSAVE)
      CALL COPYVA (1,N,K,YCALB,YCSAVE)
      IF (NPARAM .EQ. 1) WRITE (6,1080) HEAD1
      IF (NPARAM .GT. 1) WRITE (6,1220) ORD(K),HEAD1
      WRITE (6,1090) VNAM
      WRITE (6,1230) M1
      IF (NPARAM .GT. 1) WRITE (6,1240)
      DO 60 KK=1,NPARAM
      IF (NPARAM .EQ. 1) WRITE (6,1250) Q1,PARNAM(1)
      IF (NPARAM .GT. 1 .AND. KK .EQ. K) WRITE (6,1260) KK,Q1,PARNAM(KK)
      IF (KK .NE. K) WRITE (6,1270) KK,Q0(KK),PARNAM(KK)
   60 CONTINUE
      WRITE (6,1130) VNAM,YCR1
C
C.....NORMALIZE X COORDINATES AND LOCATE MINIMUM, MAXIMUM, AND CRITICAL
C     POINTS FOR KTH CALIBRATION SOLUTION.
C
      CALL SCALE (N,XCALB,1,A,B)
      CALL LOCATE (N,XCALB,YCALB,YCR1,IGRAD,LMN1,LMX1,NCR1,LCR1,XLOC1)
      YTMIN=YCALB(LMN1)
      YTMAX=YCALB(LMX1)
      IF (YTMIN .LT. YBCMIN) YBCMIN=YTMIN
      IF (YTMAX .GT. YBCMAX) YBCMAX=YTMAX
      WRITE (6,1140)
      WRITE (6,1150)
      CALL UPLOW (A,B,XLOC1,6,NCR1+2,XOUT,FLAG)
      WRITE (6,1160) XOUT(1),FLAG(1),LMN1,XOUT(2),FLAG(2),LMX1
      IF (NCR1 .GT. 0) WRITE (6,1170) NCR1,
     %      (ORD(I),XOUT(I+2),FLAG(I+2),LCR1(I),I=1,NCR1)
C
C.....CHECK FOR INVALID STRAINING SPECIFICATION.
C
      ICOUNT=0
      DO 70 I=1,NSELCT
      IF (LSELCT(I) .LE. 2) GO TO 70
      ICOUNT=ICOUNT+1
      IF (NCR0 .NE. NCR1) LTERM=1
   70 CONTINUE
C
C.....STOP EXECUTION IF CRITICAL POINTS ARE TO BE USED IN STRAINING AND
C     NUMBER OF CRITICAL POINTS IN BASE AND CALIBRATION SOLUTIONS ARE
C     UNEQUAL.
C
      IF (LTERM .EQ. 1) GO TO 900
C
C.....STOP EXECUTION IF NUMBER OF CRITICAL POINTS SELECTED EXCEEDS
C     NUMBER ACTUALLY LOCATED.
C
      IF (ICOUNT .GT. NCR0) GO TO 905
C
C.....LOAD SELECTED STRAINING POINTS INTO FIXED-POINT ARRAY FOR KTH
C     CALIBRATION SOLUTION.
C
      XFIX1(1)=0.0
      XFIX1(NFIX)=1.0
      DO 100 I=1,NSELCT
      XFIX1(I+1)=XLOC1(LSELCT(I))
  100 CONTINUE
C
C.....ARRANGE SELECTED FIXED POINTS IN A MONOTONE SEQUENCE.
C
      CALL SORT (NFIX,XFIX1,ISEQ1)
      WRITE (6,1200)
      WRITE (6,1150)
      CALL UPLOW (A,B,XFIX1,8,NFIX,XOUT,FLAG)
      WRITE (6,1210) (I,XOUT(I),FLAG(I),I=1,NFIX)
C
C.....STOP EXECUTION IF ORDER OF OCCURRENCE OF CRITICAL POINTS IN BASE
C     AND CALIBRATION SOLUTIONS DOES NOT CORRESPOND.
C
      DO 110 I=1,NFIX
      IF (ISEQ0(I) .NE. ISEQ1(I)) GO TO 910
  110 CONTINUE
C
C.....COMPUTE COEFFICIENTS IN KTH UNIT STRAINING OF XBASE
C
C           XSTR = C(K,I) + D(K,I)*XBASE,  I=1,2, ... ,NSEG,
C
C     WHERE NSEG IS THE NUMBER OF LINEAR SEGMENTS.
C
      DO 130 I=1,NSEG
      CNUM=XFIX1(I)*XFIX0(I+1)-XFIX1(I+1)*XFIX0(I)
```

```
      DNUM=XFIX1(I+1)-XFIX1(I)
      DENOM=XFIX0(I+1)-XFIX0(I)
      C(K,I)=CNUM/DENOM
      D(K,I)=DNUM/DENOM
  130 CONTINUE
C
C.....DETERMINE KTH UNIT STRAINING OF XBASE.
C
      CALL STRAIN (N,K,NSEG,XFIX0,XBASE,1.0)
      DO 140 I=1,N
  140 XUNIT(I)=XBASE(I)+DELX(I)
C
C.....INTERPOLATE CALIBRATION SOLUTION TO BASE FLOW POINTS CORRESPONDING
C     TO UNIT STRAINING.
C
      CALL INTERP (N,XCALB,YCALB,XUNIT,YINTP)
C
C.....CORRECT VALUES ON EITHER SIDE OF CRITICAL POINTS, IF THESE ARE
C     USED IN STRAINING.
C
      IF (LCORR .EQ. 0) GO TO 160
      DO 150 I=1,NCR1
      YINTP(LCR0(I))=YCALB(LCR1(I))
      YINTP(LCR0(I)+1)=YCALB(LCR1(I)+1)
  150 CONTINUE
  160 CONTINUE
C
C.....DETERMINE THE KTH UNIT PERTURBATION.
C
      DO 170 I=1,N
  170 YUNIT(K,I)=(YINTP(I)-YBASE(I))/DEL1(K)
C
C.....SAVE UNIT STRAINING IF REQUIRED FOR LATER PRINTOUT.
C
      IF (LUNIT .EQ. 0) GO TO 180
      CALL SCALE (N,XUNIT,2,A,B)
      CALL COPYVA (1,N,K,XUNIT,XUSAVE)
  180 IF (KPARAM .LT. NPARAM) RETURN
C
C.....PRINT UNIT PERTURBATION(S) AND UNIT STRAINING(S) IF LUNIT .NE. 0.
C
      IF (LUNIT .EQ. 0) RETURN
      CALL SCALE (N,XBASE,2,A,B)
      IRPT=0
      IF (NPARAM .GT. 5) IRPT=1
      KSTART=1
      KSTOP=5
      IF (KSTOP .GT. NPARAM) KSTOP=NPARAM
      GO TO 200
  190 KSTART=6
      KSTOP=NPARAM
  200 CONTINUE
      WRITE (6,1280) VNAM
      IF (NPARAM .GT. 1) WRITE (6,1290) KSTART,KSTOP
      IF (NPARAM .EQ. 1) WRITE (6,1300)
      IF (NPARAM .EQ. 1) GO TO 210
      NUM=KSTOP-KSTART+1
      IF (NUM .EQ. 1) WRITE (6,1310) (ORD(K),K=KSTART,KSTOP)
      IF (NUM .EQ. 2) WRITE (6,1320) (ORD(K),K=KSTART,KSTOP)
      IF (NUM .EQ. 3) WRITE (6,1330) (ORD(K),K=KSTART,KSTOP)
      IF (NUM .EQ. 4) WRITE (6,1340) (ORD(K),K=KSTART,KSTOP)
      IF (NUM .EQ. 5) WRITE (6,1345) (ORD(K),K=KSTART,KSTOP)
  210 CONTINUE
      CALL FILL (1,0,STPUNI)
      KLAST=20*(KSTOP-KSTART+1)
      WRITE (6,1350) (STRUNI(K),K=1,KLAST)
      WRITE (6,1360)
      DO 220 I=1,N
  220 WRITE (6,1370) I,XBASE(I),(XUSAVE(K,I),YUNIT(K,I),K=KSTART,KSTOP)
      IF (IPPT .EQ. 0) GO TO 230
      IRPT=0
      GO TO 190
  230 CALL SCALE (N,XBASE,1,A,B)
      RETURN
C
C.....CONSTRUCT PERTURBATION SOLUTIONS FOR REQUIRED CASES.
C
  504 YCR2=YCRIT(M2)
      YCR3=YCR2
      CALL COPY12 (N,X,XAERO)
      IF (LTRACE .EQ. 1) CALL COPY12 (N,Y,YAERO)
C
C.....INITIALIZE STRAINED COORDINATE AND PERTURBATION SOLUTION.
C
      DO 250 I=1,N
      XPERT(I)=XBASE(I)
  250 YPERT(I)=YBASE(I)
C
C.....ADD IN CONTRIBUTIONS FROM ALL PERTURBATIONS.
C
      DO 270 K=1,NPARAM
      DEL2=Q2(K)-Q0(K)
      DEL21=DEL2/DEL1(K)
      CALL STRAIN (N,K,NSEG,XFIX0,XBASE,DEL21)
      DO 260 I=1,N
      XPERT(I)=XPERT(I)+DELX(I)
  260 YPERT(I)=YPERT(I)+DEL2*YUNIT(K,I)
  270 CONTINUE
C
C.....ADJUST VALUES NEAR CRITICAL POINT FOR MONOTONE BEHAVIOR.
C
      IF (LCORR .EQ. 1) CALL MONO (NCR0,LCR0,XPERT,YPERT)
C
C.....INTERPOLATE SOLUTION IN STRAINED COORDINATES TO BASE X VALUES.
C
      CALL INTERP (N,XPERT,YPERT,XBASE,DUMMY)
      CALL COPY21 (N,DUMMY,Y)
C
      IF (LTRACE.EQ.0) RETURN
C
C.....COMPARISON OF PERTURBATION AND AERODYNAMIC SOLUTIONS.
C
C
C.....LOCATE MINIMUM, MAXIMUM, AND CRITICAL POINTS IN PERTURBATION
C     SOLUTION.
C
      CALL SCALE (N,XPERT,2,A,B)
      CALL SCALE (N,XPERT,1,A,B)
      CALL LOCATE (N,XPERT,YPERT,YCR2,IGRAD,LMN2,LMX2,NCR2,LCR2,XLOC2)
      YMIN=YPERT(LMN2)
```

```
      YMAX=YPERT(LMX2)
      YPCMIN=YMIN
      YPCMAX=YMAX
      WRITE (6,1380)
      WRITE (6,1090) VNAM
      WRITE (6,1390) M2
      IF (NPARAM .EQ. 1) WRITE (6,1400) Q2(1),PARNAM(1)
      IF (NPARAM .GT. 1) WRITE (6,1410) (K,Q2(K),PARNAM(K),K=1,NPARAM)
      WRITE (6,1130) VNAM,YCR2
      WRITE (6,1140)
      WRITE (6,1150)
      CALL UPLOW (A,B,XLOC2,6,NCR2+2,XOUT,FLAG)
      WRITE (6,1420) HEAD2,XOUT(1),FLAG(1),LMN2,XOUT(2),FLAG(2),LMX2
      IF (NCR2 .GT. 0) WRITE (6,1430) NCR2,
     %     (ORD(I),XOUT(I+2),FLAG(I+2),LCR2(I),I=1,NCR2)
      CALL SCALE (N,XBASE,2,A,B)
C
C.....LOCATE MINIMUM, MAXIMUM, AND CRITICAL POINTS IN AERODYNAMIC
C     SOLUTION.
C
      CALL SCALE (N,XAERO,1,A,B)
      CALL LOCATE (N,XAERO,YAERO,YCR3,IGRAD,LMN3,LMX3,NCR3,LCR3,XLOC3)
      YMIN=YAERO(LMN3)
      YMAX=YAERO(LMX3)
      IF (YMIN .LT. YPCMIN) YPCMIN=YMIN
      IF (YMAX .GT. YPCMAX) YPCMAX=YMAX
      CALL UPLOW (A,B,XLOC3,6,NCR3+2,XOUT,FLAG)
      WRITE (6,1420) HEAD3,XOUT(1),FLAG(1),LMN3,XOUT(2),FLAG(2),LMX3
      IF (NCR3 .GT. 0) WRITE (6,1430) NCR3,
     %     (ORD(I),XOUT(I+2),FLAG(I+2),LCR3(I),I=1,NCR3)
      CALL INTERP (N,XPERT,YPERT,XAERO,YPRTI)
      CALL LOCATE (N,XAERO,YPRTI,YCR3,IGRAD,LMN3,LMX3,NCR3,LCR3,XLOC3)
      YTMIN=YPRTI(LMN3)
      YTMAX=YPRTI(LMX3)
      IF (YTMIN .LT. YMIN) YMIN=YTMIN
      IF (YTMAX .GT. YMAX) YMAX=YTMAX
      CALL SCALE (N,XPERT,2,A,B)
      CALL SCALE (N,XAERO,2,A,B)
      CALL FILL (2,0,STRING)
      WRITE (6,1440) VNAM
      WRITE (6,1450) VNAM,YMAX,VNAM,YMIN,VNAM,YCR2,VNAM,VNAM
      WRITE (6,1460) VNAM,VNAM,VNAM,VNAM,(STRING(I),I=1,72)
      DO 280 I=1,N
      CALL FILL (3,I,STRING)
  280 WRITE (6,1470) I,XBASE(I),YBASE(I),XPERT(I),YPERT(I),
     % XAERO(I),YAERO(I),YPRTI(I),(STRING(II),II=1,72)
C
C.....IF LPLOT .NE. 0 GENERATE PERIPHERAL PLOT OF PERTURBATION AND
C     AERODYNAMIC SOLUTIONS.
C
      IF (LPLOT .EQ. 0) GO TO 320
      YMIN=YBCMIN
      YMAX=YBCMAX
      IF (YPCMIN .LT. YMIN) YMIN=YPCMIN
      IF (YPCMAX .GT. YMAX) YMAX=YPCMAX
      CALL DRVPLT (N,NPARAM,YMIN,YMAX,YCR2)
  320 CALL SCALE (N,XBASE,1,A,B)
      RETURN
C
C.....ABNORMAL TERMINATION OF COMPUTATION.
C
  900 WRITE (6,9000)
      GO TO 999
  905 WRITE (6,9050)
      GO TO 999
  910 WRITE (6,9100)
C
  999 WRITE (6,9500)
      STOP
C
C.....I/O FORMAT STATEMENTS FOLLOW.
C
 1000 FORMAT(1H0,132(1H*)/
     %          1X,1H*,25X,8A10,25X,1H*/
     %          1X,132(1H*)///)
 1005 FORMAT(1H1,35X,37H--- INPUT FOR PERTURBATION METHOD ---//)
 1010 FORMAT (1X,29H.....LIST OF INPUT PARAMETERS//
     %          6X,3HN =,I4//
     %          6X,3HA =,F5.1,4X,3HB =,F5.1//
     %          6X,8HNPARAM =,I2///)
 1020 FORMAT (1X,22H.....STRAINING OPTIONS//
     %          6X,23HNUMBER OF FIXED POINTS:,I2/)
 1040 FORMAT (6X,45HFIXED POINTS WILL BE AUTOMATICALLY DETERMINED/
     %          6X,44HBY THE PROGRAM FOR ALL SOLUTIONS AS FOLLOWS://
     %         11X,14HTWO END POINTS)
 1050 FORMAT (11X,16HPOINT OF MINIMUM,1X,2A1)
 1060 FORMAT (11X,16HPOINT OF MAXIMUM,1X,2A1)
 1070 FORMAT (11X,2A1,6HCRIT (,A5,6HPOINT))
 1080 FORMAT (1H1,26HRESULTS OF COMPUTATIONS ON,1X,5A4///)
 1090 FORMAT (1X,17H.....MACH NUMBER,/
     %          6X,34HVALUES OF PERTURBATION PARAMETERS,/
     %          6X,17HCRITICAL VALUE OF,1X,2A1,1H:/)
 1100 FORMAT (11X,4HM0 =,F7.4/)
 1110 FORMAT (11X,4HQ0 =,F10.4,5X,1H(,A8,1H)/)
 1120 FORMAT (11X,3HQ0(,I1,3H) =,F10.4,5X,1H(,A8,1H)/)
 1130 FORMAT (11X,2A1,6HCRIT =,F8.4///)
 1140 FORMAT (1X,47H.....LOCATIONS OF MIN., MAX., AND CRITICAL PTS.)
 1150 FORMAT (3X,34H(* DENOTES POINT ON LOWER SURFACE)/)
 1160 FORMAT (6X,14HMINIMUM AT X =,F7.4,A1,3X,10H(POINT NO.,I4,1H)/
     %          6X,14HMAXIMUM AT X =,F7.4,A1,3X,10H(POINT NO.,I4,1H))
 1170 FORMAT (6X,I1,1X,18HCRITICAL POINT(S):/
     %          (9X,A5,6HAT X =,1X,F6.4,A1,3X,
     %               16H(AFTER POINT NO.,I4,1H)))
 1200 FORMAT (///1X,29H.....LOCATION OF FIXED POINTS)
 1210 FORMAT (11X,5HXFIX(,I1,3H) =,F7.4,A1)
 1220 FORMAT (1H1,26HRESULTS OF COMPUTATIONS ON,1X,A5,5A4///)
 1230 FORMAT (11X,4HM1 =,F7.4/)
 1240 FORMAT (2X,41H(** DENOTES PERTURBATION FROM BASE VALUE)/)
 1250 FORMAT (11X,4HQ1 =,F10.4,5X,1H(,A8,1H)/)
 1260 FORMAT (9X,5H**Q1(,I1,3H) =,F10.4,5X,1H(,A8,1H)/)
 1270 FORMAT (11X,3HQ1(,I1,3H) =,F10.4,5X,1H(,A8,1H)/)
 1280 FORMAT (1H1,20HUNIT PERTURBATION OF,1X,2A1,1X,
     %               27HAND UNIT STRAINING OF XBASE)
 1290 FORMAT (26H FOR CALIBRATION SOLUTIONS,I2,1X,7HTHROUGH,I2)
 1300 FORMAT (//1H )
 1310 FORMAT (///19X,1(1H*,A5,11HCALB SOLN *,3X))
 1320 FORMAT (///19X,2(1H*,A5,11HCALB SOLN *,3X))
 1330 FORMAT (///19X,3(1H*,A5,11HCALB SOLN *,3X))
 1340 FORMAT (///19X,4(1H*,A5,11HCALB SOLN *,3X))
 1345 FORMAT (///19X,5(1H*,A5,11HCALB SOLN *,3X))
```

```
 1350 FORMAT(1X,5HPOINT,4X,5HXBASE,4X,100A1)
 1360 FORMAT (1X)
 1370 FORMAT (1X,I4,1X,11F10.4)
 1380 FORMAT (1H1,56(1H*)/1X,2H* ,
     % 52HCOMPARISON OF AERODYNAMIC AND PERTURBATION SOLUTIONS,
     % 2H */1X,56(1H*)///)
 1390 FORMAT (11X,4HM2 =,F7.4/)
 1400 FORMAT (11X,4HQ2 =,F7.4,5X,1H(,A8,1H)/)
 1410 FORMAT (11X,3HQ2(,I1,3H) =,F7.4,5X,1H(,A8,1H)/)
 1420 FORMAT (/6X,5A4//
     %         11X,14HMINIMUM AT X =,F7.4,A1,3X,10H(POINT NO.,I4,1H)/
     %         11X,14HMAXIMUM AT X =,F7.4,A1,3X,10H(POINT NO.,I4,1H))
 1430 FORMAT (1H ,10X,I1,1X,18HCRITICAL POINT(S):/
     %         (14X,A5,6HAT X =,F7.4,A1,3X,
     %              16H(AFTER POINT NO.,I4,1H)))
 1440 FORMAT (///1X,44H.....FINAL PRINTOUT AND GRAPHICAL DISPLAY OF,1X,
     %         2A1)
 1450 FORMAT (/72X,21HH = MAXIMUM  VALUE OF,1X,2A1,1X,1H=,F8.4/
     %          72X,21HL = MINIMUM  VALUE OF,1X,2A1,1X,1H=,F8.4/
     %          72X,21H* = CRITICAL VALUE OF,1X,2A1,1X,1H=,F8.4/
     %          72X,12HP = VALUE OF,1X,2A1,1X,
     %              34HPREDICTED BY PERTURBATION SOLUTION/
     %          72X,12HA = VALUE OF,1X,2A1,1X,
     %              23HIN AERODYNAMIC SOLUTION/
     %          72X,29H$ = AGREEMENT BETWEEN P AND A)
 1460 FORMAT (/2X,2HPT,2X,5HXBASE,3X,2A1,4HBASE,2X,5HXPERT,3X,2A1,
     %          4HPERT,2X,5HXAERO,3X,2A1,4HAERO,2X,2A1,4HPINT,1X,72A1/)
 1470 FORMAT (1X,I3,7F8.4,1X,72A1)
C
C.....ABNORMAL TERMINATION FORMATS FOLLOW.
C
 9000 FORMAT (///1X,28HNUMBER OF CRITICAL POINTS IN/
     %          1X,30HBASE AND CALIBRATION SOLUTIONS/
     %          1X,31HARE UNEQUAL - CALCULATION ENDED)
 9050 FORMAT (///1X,25HNUMBER OF CRITICAL POINTS/
     %          1X,23HSELECTED EXCEEDS NUMBER/
     %          1X,30HACTUALLY LOCATED - CALCULATION/
     %          1X,5HENDED)
 9100 FORMAT (///1X,28HORDER OF SPECIFIED POINTS IN/
     %          1X,30HBASE AND CALIBRATION SOLUTIONS/
     %          1X,39HDOES NOT CORRESPOND - CALCULATION ENDED)
 9500 FORMAT (1H1)
      END
      SUBROUTINE COPY12 (N,XIN,XOUT)
C
C.....COPIES ELEMENTS OF VECTOR XIN (LEVEL 1) TO VECTOR XOUT (LEVEL 2).
C
      DIMENSION XIN(200),XOUT(200)
      LEVEL 2, XOUT
      DO 10 I=1,N
   10 XOUT(I)=XIN(I)
      RETURN
      END
      SUBROUTINE COPY21 (N,XIN,XOUT)
C
C.....COPIES ELEMENTS OF VECTOR XIN (LEVEL 2) TO VECTOR XOUT (LEVEL 1).
C
      DIMENSION XIN(200),XOUT(200)
      LEVEL 2, XIN
      DO 10 I=1,N
   10 XOUT(I)=XIN(I)
      RETURN
      END
      SUBROUTINE COPYVA (ICALL,N,K,VECTOR,ARRAY)
C
C.....COPIES ELEMENTS OF VECTOR INTO KTH ROW OF ARRAY.
C
      DIMENSION VECTOR(200),ARRAY(10,200)
      LEVEL 2, VECTOR,ARRAY
      IF (ICALL .EQ. 2) GO TO 20
      DO 10 I=1,N
   10 ARRAY(K,I)=VECTOR(I)
      RETURN
   20 CONTINUE
      DO 30 I=1,N
   30 VECTOR(I)=ARRAY(K,I)
      RETURN
      END
      SUBROUTINE DRVPLT (N,NPARAM,YMIN,YMAX,YCR2)
      DIMENSION HLINE3(3),XPLOT(200),YPLOT(200)
      COMMON /HEAD/ ZTITLE(8),JOBKEY
      COMMON /SAVE/ XCSAVE(10,200),YCSAVE(10,200)
      COMMON /XY/ XBASE(200),XCALB(200),XPERT(200),XAERO(200),
     % XUNIT(200),YBASE(200),YCALB(200),YPERT(200),YAERO(200),
     % YINTP(200),YPRTI(200),DUMMY(200)
      LEVEL 2, XCSAVE,YCSAVE
      LEVEL 2, XBASE,XCALB,XPERT,XAERO,XUNIT,YBASE,YCALB,YPERT,YAERO,
     % YINTP,YPRTI,DUMMY
      DATA NPLOT /0/
      IF (NPLOT .EQ. 0) CALL BETA
      MIN=10.0*(YMIN-0.1)
      MAX=10.0*(YMAX+0.1)
      YMIN=0.1*MIN
      YMAX=0.1*MAX
      DO 20 K=1,NPARAM
      NPLOT=NPLOT-1
      ENCODE (22,1010,HLINE3) K,NPARAM
      CALL BGNPL (-1)
      CALL MIXALF ("L/CST")
      CALL MX3ALF ("INSTR","%")
      CALL SIMPLX
      CALL TITLE (1H ,1,1HX,1,"%E0.5)C%EX(P)$",100,6.0,8.0)
      CALL HEADIN ("PLOT (OF) C%L0.25H0.7(P)$",100,3,3)
      CALL HEADIN (JOBKEY,9,2,3)
      CALL HEADIN (HLINE3,22,2,3)
      CALL GRAF (0.0,"SCALE",1.0,YMAX,"SCALE",YMIN)
      CALL FRAME
      IF (YCR2 .GT. YMAX) GO TO 10
      CALL RLVEC (0.0,YCR2,0.2,YCR2,0000)
      CALL RLMESS ("C%L(P%BE(*)$",100,0.21,YCR2)
   10 CONTINUE
      CALL DASH
      CALL COPY21 (N,XBASE,XPLOT)
      CALL COPY21 (N,YBASE,YPLOT)
      CALL CURVE (XPLOT,YPLOT,N,0)
      CALL RESET ("DASH")
      CALL DOT
      CALL COPYVA (2,N,K,XCALB,XCSAVE)
      CALL COPYVA (2,N,K,YCALB,YCSAVE)
      CALL COPY21 (N,XCALB,XPLOT)
```

```
      CALL COPY21 (N,YCALB,YPLOT)
      CALL CURVE (XPLOT,YPLOT,N,0)
      CALL RESET ("DOT")
      CALL MARKER (1)
      CALL COPY21 (N,XPERT,XPLOT)
      CALL COPY21 (N,YPERT,YPLOT)
      CALL CURVE (XPLOT,YPLOT,N,-1)
      CALL RESET ("MARKER")
      CALL COPY21 (N,XAERO,XPLOT)
      CALL COPY21 (N,YAERO,YPLOT)
      CALL CURVE (XPLOT,YPLOT,N,0)
      CALL ENDPL (NPLOT)
   20 CONTINUE
 1010 FORMAT (16HCALIBRATION NO. ,I1,4H OF ,I1)
      RETURN
      END
      SUBROUTINE FILL (ICALL,I,STRING)
C
C.....FILLS ARRAY STRING WITH CHARACTERS FOR TABLE HEADINGS AND PRINTER
C     PLOTS.
C
      DIMENSION STRING(100),UNIT(20)
      COMMON /PARAM/ PARNAM(10),LSELCT(6),LUNIT,LPLOT,NSELCT,A,B,VNAM(2)
      COMMON /MINMAX/ YMIN,YMAX,YCR2
      COMMON /XY/ XBASE(200),XCALB(200),XPERT(200),XAERO(200),
     % XUNIT(200),YBASE(200),YCALB(200),YPERT(200),YAERO(200),
     % YINTP(200),YPRTI(200),DUMMY(200)
      LEVEL 2, XBASE,XCALB,XPERT,XAERO,XUNIT,YBASE,YCALB,YPERT,YAERO,
     % YINTP,YPRTI,DUMMY
      DATA IENT /0/
      DATA STAR/1H*/, P/1HP/, AA/1HA/, DASH/1H-/, H/1HH/, EL/1HL/,
     %     BLANK/1H /, DOLLAR/1H$/
      DATA UNIT /1HX,1HS,1HT,1HR,1HU,1HN,1HI,1HT,1H ,1H ,
     %           1H ,1H ,1H ,1HU,1HN,1HI,1HT,1H ,1H ,1H /
      GO TO (10,30,50), ICALL
   10 IENT=IENT+1
      IF (IENT .GT. 1) RETURN
      UNIT(12)=VNAM(1)
      UNIT(13)=VNAM(2)
      DO 20 J=1,20
      DO 20  K=1,5
      II=J+20*(K-1)
   20 STRING(II)=UNIT(J)
      RETURN
   30 RANGE=YMAX-YMIN
      IFLAG=0
      IF (YCR2.GT.YMAX .OR. YCR2.LT.YMIN) IFLAG=1
      DO 40 II=1,72
   40 STRING(II)=DASH
      STRING(1)=H
      STRING(72)=EL
      IF (IFLAG.EQ.1) RETURN
      NSTAR=1+(YMAX-YCR2)/RANGE*71
      STRING(NSTAR)=STAR
      RETURN
   50 CONTINUE
      DO 60 II=1,72
   60 STRING(II)=BLANK
      IF (IFLAG.EQ.0) STRING(NSTAR)=STAR
      YP=YPRTI(I)
      NPERT=1+(YMAX-YP)/RANGE*71
      STRING(NPERT)=P
      YC=YAERO(I)
      NCHEK=1+(YMAX-YC)/RANGE*71
      STRING(NCHEK)=AA
      IF (NPERT .EQ. NCHEK) STRING(NPERT)=DOLLAR
      RETURN
      END
      SUBROUTINE INPUT (NPARAM)
C
C     INPUT FOR SUBROUTINE PERTURB IS REQUIRED IN THE FOLLOWING FORM.
C     FOR DETAILS SEE ACCOMPANYING MANUAL.
C
C**** ITEM NO. 1 - ONE CARD (8A10) ************************************
C
C          TITLE      IDENTIFIES JOB - PRINTED AS HEADLINE ON FIRST PAGE
C                     OF OUTPUT. FIRST NINE CHARACTERS ARE USED TO
C                     IDENTIFY PERIPHERAL PLOT.
C
C**** ITEM NO. 2 - ONE CARD (16I5) ************************************
C
C          NSELCT     NUMBER OF POINTS (IN ADDITION TO END POINTS) TO BE
C                     HELD INVARIANT IN STRAINING.
C                           NOTE: 1 .LE. NSELCT .LE. 6.
C
C          LUNIT      CONTROLS WHETHER OR NOT UNIT COORDINATE STRAINING(S)
C                     AND UNIT PERTURBATION(S) ARE PRINTED.
C
C                     LUNIT = 0 ... NO OUTPUT
C                     LUNIT = 1 ... OUTPUT
C
C          LPLOT      SPECIFIES WHETHER OR NOT AN ADDITIONAL PLOT BY A
C                     PERIPHERAL DEVICE IS TO BE MADE (SOFTWARE MUST BE
C                     SUPPLIED BY USER IN SUBROUTINE DRVPLT).
C
C                     LPLOT = 0 ... NO PERIPHERAL PLOT
C                     LPLOT = 1 ... PERIPHERAL PLOT
C
C**** ITEM NO. 3 - ONE CARD (16I5) ************************************
C
C          LSELCT(I) ...
C                     ARRAY OF LENGTH 6 OF WHICH NSELCT ELEMENTS ARE READ
C                     IN. SPECIFIES NATURE OF POINTS TO BE HELD INVARIANT
C                     ACCORDING TO THE CODE
C
C                             1 ... MINIMUM PT. HELD INVARIANT
C                             2 ... MAXIMUM PT. HELD INVARIANT
C                             3 ... 1ST CRITICAL PT. HELD INVARIANT
C                             4 ... 2ND CRITICAL PT. HELD INVARIANT
C                             5 ... 3RD CRITICAL PT. HELD INVARIANT
C                             6 ... 4TH CRITICAL PT. HELD INVARIANT
C
C                     NOTE THAT THE CODE NUMBERS CAN BE ASSIGNED IN ANY
C                     ORDER, E.G.
C
C                                       LSELCT(1) = 1
C                                       LSELCT(2) = 3
C                                       LSELCT(3) = 4
C
C                     IS EQUIVALENT TO
```

```
C
C                                LSELCT(1) = 4
C                                LSELCT(2) = 1
C                                LSELCT(3) = 3
C
C                 BOTH CORRESPONDING TO NSELCT = 3 WITH THE MINIMUM,
C                 AND FIRST AND SECOND CRITICAL POINTS HELD INVARIANT.
C
C**** ITEM NO. 4 - ONE CARD (2A1) **************************************
C
C        VNAM     CHARACTER STRING OF LENGTH 2 WHICH SYMBOLIZES
C                 DEPENDENT VARIABLE, E.G. "CP" FOR PRESSURE
C                 COEFFICIENT.
C
C**** ITEM NO. 5 - ONE CARD (10A8) *************************************
C
C        PARNAM(K) ...
C                 ARRAY OF 8-CHARACTER STRINGS WHICH IDENTIFY THE
C                 PARAMETERS VARIED. NPARAM ELEMENTS OF THE ARRAY
C                 ARE READ IN.
C
C**** ITEM NO. 6 - ONE CARD (8F10.6) ***********************************
C
C        A        SCALING PARAMETER (A = -X(1), WHERE X(1) IS FIRST
C                 DATA POINT ON LOWER SURFACE ... SEE MANUAL).
C
C        B        SCALING PARAMETER (B = X(N), WHERE X(N) IS LAST DATA
C                 POINT ON UPPER SURFACE ... SEE MANUAL).
C
C***********************************************************************
C
      COMMON /HEAD/ TITLE(8),JOBKEY
      COMMON /PARAM/ PARNAM(10),LSELCT(6),LUNIT,LPLOT,NSELCT,A,B,VNAM(2)
      READ (5,900) TITLE
      DECODE (9,950,TITLE(1)) JOBKEY
      READ (5,1000) NSELCT,LUNIT,LPLOT
      READ (5,1000) (LSELCT(I),I=1,NSELCT)
      READ (5,1010) VNAM
      READ (5,1020) (PARNAM(I),I=1,NPARAM)
      READ (5,1030) A,B
      RETURN
  900 FORMAT (8A10)
  950 FORMAT (A9)
 1000 FORMAT (16I5)
 1010 FORMAT (2A1)
 1020 FORMAT (10A8)
 1030 FORMAT (8F10.6)
      END
      SUBROUTINE INTERP (N,X,Y,XI,YI)
C
C.....GIVEN THE SET OF POINTS X(I), Y(I), I=1,N, AND THE SET XI(J),
C     J=1,N, USES LINEAR INTERPOLATION TO COMPUTE THE SET YI(J), J=1,N.
C
      DIMENSION X(200),Y(200),XI(200),YI(200)
      LEVEL 2, X,Y,XI,YI
      NM1=N-1
      JSTART=1
      DO 70 I=1,N
      IF (XI(I) .LE. X(1)) GO TO 10
      IF (XI(I) .GE. X(N)) GO TO 20
      GO TO 30
   10 J=1
      GO TO 60
   20 J=N-1
      GO TO 60
   30 CONTINUE
      DO 50 J=JSTART,NM1
      IF (XI(I) .NE. X(J)) GO TO 40
      YI(I)=Y(J)
      GO TO 70
   40 IF (XI(I) .GT. X(J) .AND. XI(I) .LT. X(J+1)) GO TO 60
   50 CONTINUE
   60 SLOPE=(Y(J+1)-Y(J))/(X(J+1)-X(J))
      YI(I)=Y(J)+SLOPE*(XI(I)-X(J))
      JSTART=J
   70 CONTINUE
      RETURN
      END
      SUBROUTINE LOCATE (N,X,Y,YCRIT,IGRAD,LMIN,LMAX,NCRIT,LCRIT,XLOC)
C
C.....OPERATES ON THE INPUT ARRAY Y, LOCATING MINIMUM AND MAXIMUM
C     VALUES, AND ALL CRITICAL POINTS (Y=YCRIT) FOR WHICH DY/DX (IN
C     PHYSICAL COORDINATES) HAS ALGEBRAIC SIGN GIVEN BY IGRAD. NCRIT
C     IS NUMBER OF CRITICAL POINTS. POINTS FOUND ARE STORED IN THE ARRAY
C     XLOC AS FOLLOWS
C
C                          XLOC(1) = MINIMUM PT.
C                          XLOC(2) = MAXIMUM PT.
C                          XLOC(3) = CRITICAL PT. NO. 1
C                          ...     = ...
C                          XLOC(6) = CRITICAL PT. NO. 4
C
      DIMENSION X(200),Y(200),LCRIT(4),XCRIT(4),XLOC(6)
      LEVEL 2, X,Y
      COMMON /FLOREV/ IREV
      IFLOW=-1
      LMIN=1
      LMAX=1
      ISTART=2
      IF (IREV .EQ. 0) GO TO 10
      LMIN=2
      LMAX=2
      ISTART=3
   10 CONTINUE
      NCRIT=0
      DO 30 I=ISTART,N
      IF (IREV .NE. 0 .AND. I .EQ. N) GO TO 20
      IF (Y(I) .GT. Y(LMAX)) LMAX=I
      IF (Y(I) .LT. Y(LMIN)) LMIN=I
   20 CONTINUE
      IF ((Y(I) .GT. YCRIT .AND. Y(I-1) .GT. YCRIT) .OR.
     %    (Y(I) .LT. YCRIT .AND. Y(I-1) .LT. YCRIT)) GO TO 30
      IF (I .GT. IREV) IFLOW=1
      IF ((Y(I)-Y(I-1))*FLOAT(IFLOW*IGRAD) .LT. 0.0) GO TO 30
      NCRIT=NCRIT+1
      LCRIT(NCRIT)=I-1
      SLOPE=(X(I)-X(I-1))/(Y(I)-Y(I-1))
      XCRIT(NCRIT)=X(I-1)+SLOPE*(YCRIT-Y(I-1))
   30 CONTINUE
      XLOC(1)=X(LMIN)
```

```
      XLOC(2)=X(LMAX)
      IF (NCRIT .EQ. 0) RETURN
      DO 40 I=1,NCRIT
   40 XLOC(I+2)=XCRIT(I)
      RETURN
      END
      SUBROUTINE MONO (N,L,X,Y)
C
C.....CHECKS POINTS IN VICINITY OF A CRITICAL POINT FOR MONOTONE
C     BEHAVIOR, AND ADJUSTS VALUES IF NECESSARY TO GIVE A LINEAR
C     PROFILE.
C
      DIMENSION L(4),X(200),Y(200)
      LEVEL 2, X,Y
      DO 10 I=1,N
      LS=L(I)
      Y1=Y(LS-1)
      Y2=Y(LS)
      Y3=Y(LS+1)
      Y4=Y(LS+2)
      IF ((Y1 .LT. Y2) .AND. (Y2 .LT. Y3) .AND. (Y3 .LT. Y4)) GO TO 10
      IF ((Y1 .GT. Y2) .AND. (Y2 .GT. Y3) .AND. (Y3 .GT. Y4)) GO TO 10
      X1=X(LS-1)
      X2=X(LS)
      X3=X(LS+1)
      X4=X(LS+2)
      SLOPE=(Y4-Y1)/(X4-X1)
      Y(LS)=Y1+SLOPE*(X2-X1)
      Y(LS+1)=Y1+SLOPE*(X3-X1)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SCALE (N,X,M,A,B)
C
C.....ENTRY WITH M = 1 CONVERTS FROM PHYSICAL X (0 TO -A ON LOWER
C     SURFACE, 0 TO B ON UPPER SURFACE) TO NORMALIZED X (0 .LT. X .LT.
C     1) ENTRY WITH M=2 REVERSES THE PROCESS. NZ (DETERMINED WHEN M=1)
C     CORRESPONDS TO POINT AT NOSE OF BLADE OR AIRFOIL.
C
      COMMON /FLOREV/ NZ
      DIMENSION X(200)
      LEVEL 2, X
      IF (M .EQ. 2) GO TO 30
      CONTINUE
      NZ=0
      DO 10 I=2,N
      IF (X(I) .LT. X(I-1)) NZ=I
   10 CONTINUE
      DO 20 I=1,N
      IF (I .LE. NZ) T=-X(I)
      IF (I .GT. NZ) T=X(I)
      X(I)=(T-A)/(B-A)
   20 CONTINUE
      RETURN
   30 DO 40 I=1,N
      X(I)=ABS((B-A)*X(I)+A)
   40 CONTINUE
      RETURN
      END
      SUBROUTINE SORT (N,X,ISEQ)
C
C.....ARRANGES THE SET X(1), X(2), ... , X(N) IN A MONOTONE INCREASING
C     SEQUENCE. ISEQ GIVES ORDER OF SUBSCRIPTS IN REARRANGED SEQUENCE.
C
      DIMENSION X(8),ISEQ(8)
      NM1=N-1
      DO 10 I=1,N
   10 ISEQ(I)=I
   20 ITEST=0
      DO 30 I=1,NM1
      IF (X(I) .LE. X(I+1)) GO TO 30
      XSAVE=X(I)
      X(I)=X(I+1)
      X(I+1)=XSAVE
      ISAVE=ISEQ(I)
      ISEQ(I)=ISEQ(I+1)
      ISEQ(I+1)=ISAVE
      ITEST=1
   30 CONTINUE
      IF (ITEST .EQ. 1) GO TO 20
      RETURN
      END
      SUBROUTINE STRAIN (N,K,NSEG,XFIX,XIN,PARM)
C
C.....COMPUTES STRAINING INCREMENT DELX FROM INPUT ARRAY XIN, USING
C     PIECEWISE LINEAR STRAINING WITH NSEG LINEAR SEGMENTS. FOR UNIT
C     STRAINING, INPUT VALUE OF PARM IS 1.0; FOR GENERAL CASE,
C
C                         PARM = (Q2(K)-Q0(K))/(Q1-Q0(K)).
C
      DIMENSION XFIX(8),XIN(200)
      COMMON /COEFF/ C(10,7),D(10,7),DELX(200)
      LEVEL 2, C,D,DELX,XIN
      JSTART=1
      DO 30 I=1,N
      DO 10 J=JSTART,NSEG
      IF (XIN(I) .GE. XFIX(J) .AND. XIN(I) .LE. XFIX(J+1)) GO TO 20
   10 CONTINUE
   20 DELX(I)=PARM*(C(K,J)+(D(K,J)-1.0)*XIN(I))
      JSTART=J
   30 CONTINUE
      RETURN
      END
      SUBROUTINE UPLOW (A,B,XIN,K,N,XOUT,FLAG)
C
C.....CONVERTS NORMALIZED ARRAY XIN TO PHYSICAL ARRAY XOUT AND FLAGS
C     POINTS ON LOWER SURFACE WITH A "*".
C
      DIMENSION XIN(K),XOUT(8)
      DIMENSION FLAG(8)
      DATA BLANK/1H /, STAR/1H*/
      XNOSE=-A/(B-A)
      DO 10 I=1,N
      FLAG(I)=BLANK
      IF (XIN(I) .LT. XNOSE) FLAG(I)=STAR
      XOUT(I)=ABS((B-A)*XIN(I)+A)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE TSONIC
```

```
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1  LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
     2  NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
     3  INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
     4  NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
     5  PLOSS(50),MSP(50,2),THSP(50,2)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1  HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
     2  WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3  THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4  BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5  BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6  BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7  BEH(100,2),PLOSMH(100,2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      LEVEL 2,  A,U,K,RHO
      COMMON /HRBAAK/ H(4),R(4),B(4),KAK(4),KA(4),RZ,BZ,IH(4)
      COMMON /VARCOM/ RHOHB(100,2),RHOVB(100,2),WMB(100,2),WTB(100,2),
     1  WWCRM(100,2),LABEL(1,100)
      COMMON /SLCOM/ USL(100,11),TSL(100,11)
      LEVEL 2,  USL,TSL
      COMMON /BCDCOM/ INIT(2),EM(50,2),D2TDM2(100,2)
      COMMON /PLTCOM/ TSLPT(1100),XDOWN(400),YACROS(400)
      LEVEL 2,  TSLPT,XDOWN,YACROS
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1  UPPER,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      DATA ICALL/0/
      ICALL=ICALL+1
      IEND=-1
      ITER = 0
      INIT(1) = 0
      INIT(2) = 0
      CALL TINPUT
      IF(ICALL.EQ.1) RETURN
      IF (BLDAT.GE.2) CALL BLDPLT
      CALL PRECAL
   30 ITER = ITER+1
      CALL COEF
      CALL SOR
      CALL VELMER
      CALL VELTAN
      CALL STRLIN
      CALL OUTPUT
      IF(NER(2).GT.0) RETURN
      IF(IEND.LE.0) GO TO 30
      IF(REDFAC.LT.1.) CALL TVELCY
      RETURN
      END
      SUBROUTINE TINPUT
C
C  INPUT READS AND PRINTS ALL INPUT DATA CARDS AND CALCULATES HORIZONTAL
C  SPACING (MV ARRAY)
C
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1  LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
     2  NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
     3  INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
     4  NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
     5  PLOSS(50),MSP(50,2),THSP(50,2)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1  HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
     2  WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3  THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4  BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5  BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6  BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7  BEH(100,2),PLOSMH(100,2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      LEVEL 2,  A,U,K,RHO
      COMMON /MBIMBO/ MBIZ,MBOZ
      COMMON /MVNORM/ XMVN(100)
      DIMENSION SPLNO(2),CARD(8)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1  UPPER,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      DATA ICALL/0/, END/10H$END      /
C
C  READ AND PRINT ALL INPUT DATA
C
      NREAD = 1
      NWRIT = 6
      ICALL=ICALL+1
      IF (ICALL.GT.1) GO TO 2
    1 READ (5,990) CARD
      IF (CARD(1).EQ.END) GO TO 2
      WRITE(1,990) CARD
      GO TO 1
    2 REWIND 1
      WRITE(NWRIT,1000)
      WRITE(NWRIT,1110)
      READ (NREAD,1030) GAM,AR,TIP,RHOIP,WTFL,BLANK,OMEGA,ORF
      WRITE(NWRIT,1040) GAM,AR,TIP,RHOIP,WTFL,BLANK,OMEGA,ORF
      READ (NREAD,1030) BETAI,BETAO,BLANK,BLANK,FSMI,FSMO
      READ (NREAD,1030) REDFAC,DENTOL,SSM1,SSM2
      IF (DENTOL.LE.0.) DENTOL = .01
C -- MOPT = 0, NO CORRECTION TO THE BESP ARRAY (REDFAC MUST EQUAL 1.0)
C      MOPT = 1, READ IN WOWCR ARRAY FOR CALCULATING REDUCED FLOW BESP
C      MOPT = 2, REDUCED BESP ARRAY CALCULATED BY PROGRAM.
      READ (NREAD,1010) MBI,MBO,BLANK,BLANK,MM,NBBI,NBL,NRSP,MOPT,LOPT,
     1   LRVB
      MBIZ=MBI
      MBOZ=MBO
      IF (LRVB.EQ.1) GO TO 6
      WRITE(NWRIT,1120)
      WRITE(NWRIT,1040)BETAI,BETAO,CHORD(1),STGR(1),FSMI,FSMO
      GO TO 8
    6 WRITE(NWRIT,1122)
      LAMBDA = BETAI
      RVTHO  = BETAO
      WRITE(NWRIT,1040) LAMBDA,RVTHO,CHORD(1),STGR(1)
    8 WRITE(NWRIT,1125)
      WRITE(NWRIT,1040) REDFAC,DENTOL,SSM1,SSM2
      WRITE(NWRIT,1130)
      WRITE(NWRIT,1020) MBI,MBO,BLANK,BLANK,MM,NBBI,NBL,NRSP,MOPT,LOPT,
     1   LRVB
      DO 10 J=1,2
      IF (J.EQ.1) WRITE(NWRIT,1140)
```

```
      IF (J.EQ.2) WRITE(NWRIT,1150)
      WRITE(NWRIT,1180) J,J,J,J,J
      READ (NREAD,1030) BLANK,BLANK,BLANK,BLANK,SPLNO(J)
      WRITE(NWRIT,1040) RI(J),RO(J),BETI(J),BETO(J),SPLNO(J)
      NSPI(J) = SPLNO(J)
      NSP = NSPI(J)
      WRITE(NWRIT,1190) J
      WRITE(NWRIT,1040) (MSP(I,J),I=1,NSP)
      WRITE(NWRIT,1200) J
   10 WRITE(NWRIT,1040) (THSP(I,J),I=1,NSP)
      WRITE(NWRIT,1210)
      READ (NREAD,1030) (MR(I),I=1,NRSP)
      WRITE(NWRIT,1040) (MR(I),I=1,NRSP)
      WRITE(NWRIT,1220)
      READ (NREAD,1030) (RMSP(I),I=1,NRSP)
      WRITE(NWRIT,1040) (RMSP(I),I=1,NRSP)
      WRITE(NWRIT,1230)
      READ (NREAD,1030) (BESPF(I),I=1,NRSP)
      WRITE(NWRIT,1040) (BESPF(I),I=1,NRSP)
      DO  20  I=1, NRSP
      BESP(I) = BESPF(I)
   20 PLOSS(I) = 0.
      IF (MOPT.NE.1) GO TO 40
      WRITE(NWRIT,1236)
      READ (NREAD,1030) (WOWCR(I),I=1,NRSP)
      WRITE(NWRIT,1040) (WOWCR(I),I=1,NRSP)
   40 IF (LOPT.EQ.0) GO TO 60
      WRITE(NWRIT,1237)
      READ (NREAD,1030) (PLOSS(I),I=1,NRSP)
      WRITE(NWRIT,1040) (PLOSS(I),I=1,NRSP)
   60 WRITE(NWRIT,1240)
      READ (NREAD,1010) BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      WRITE(NWRIT,1020) BLDAT,AANDK,ERSOR,STRFN,SLCRD,INTVL,SURVL
      IF(ICALL.EQ.1) RETURN
      IF (MM.LE.100.AND.NBBI.LE.50.AND.NRSP.LE.50.AND.NSPI(1).LE.50
     1   .AND.NSPI(2).LE.50) GO TO 70
      WRITE(NWRIT,1250)
      STOP
   70 IF (REDFAC.EQ.1..OR.MOPT.NE.0) GO TO 75
      WRITE(NWRIT,1260)
      STOP
C
C  CALCULATE MV ARRAY
C
   75 HM1 = CHORD(1)/FLOAT(MBO-MBI)
      DO 80 IM=1,MM
      MV(IM) = FLOAT(IM-MBI)*HM1
      XMVN(IM)=MV(IM)/CHORD(1)
      IF (SSM1.EQ.SSM2) GO TO 80
      IF (MV(IM).LT.SSM1) IMS1 = IM+1
      IF (MV(IM).LE.SSM2) IMS2 = IM
   80 CONTINUE
      MV(MBO) = CHORD(1)
      XMVN(MBO)=1.0
C
C  CALCULATE MISCELLANEOUS CONSTANTS
C
      NER(1) = 0
      NER(2) = 0
      NER(1) = 0
      NER(2) = 0
      PITCH = 2.*3.1415927/FLOAT(NBL)
      HT = PITCH/FLOAT(NBBI)
      DTLR = HT/1000.
      DMLR = HM1/1000.
      BV(1) = 0.
      BV(2) = 1.
      MBIM1 = MBI-1
      MBIP1 = MBI+1
      MBOM1 = MBO-1
      MBOP1 = MBO+1
      MMM1 = MM-1
      CP = AR/(GAM-1.)*GAM
      EXPON = 1./(GAM-1.)
      TWW = 2.*OMEGA/WTFL
      CPTIP = 2.*CP*TIP
      TGROG = 2.*GAM*AR/(GAM+1.)
      CALL SPLINT(MR,RMSP,NRSP,MV,MM,RM,SAL,AAA)
      CALL SPLINT(MR,BESPF,NRSP,MV,MM,BEF,DBFDM,AAA)
      CALL SPLINT(MR,BESP,NRSP,MV,MM,BE,DBDM,AAA)
      CALL SPLINT(MR,PLOSS,NRSP,MV,MM,PLOSIM,BBB,AAA)
C
C  CALCULATE GEOMETRICAL CONSTANTS
C
      CHORD(2) = CHORD(1)
      STGR(2) = STGR(1)
      MLE(1) = 0.
      MLE(2) = 0.
      THLE(1) = 0.
      THLE(2) = PITCH
      RMI(1) = RM(MBI)
      RMI(2) = RM(MBI)
      RMO(1) = RM(MBO)
      RMO(2) = RM(MBO)
C
C  INITIALIZE U AND K ARRAYS AND SURFACE DENSITY ARRAYS
C
      DO 90 I=1,2500
      U(I) = 1.
      K(I)   = 0.
   90 RHO(I) = RHOIP
C     INITIALIZE A ARRAY

      DO 91 KDUM = 1,4
      DO 91 I = 1,2500
   91 A(I,KDUM) = 0.
      NREAD=5
      RETURN
C
C  FORMAT STATEMENTS
C
  990 FORMAT (8A10)
 1000 FORMAT(1H1,35X,
     *       51H--- INPUT FOR TSONIC BLADE-TO-BLADE FLOW SOLVER ---//)
 1010 FORMAT (16I5)
 1020 FORMAT (1X,16I7)
 1030 FORMAT (8F10.5)
 1040 FORMAT (1X,8G16.7)
```

```
1100 FORMAT (20A4)
1105 FORMAT (1X,20A4)
1110 FORMAT (7X,3HGAM,14X,2HAR,13X,3HTIP,12X,5HRHOIP,12X,4HWTFL,11X,6H
    1      ,10X,5HOMEGA,12X,3HORF)
1120 FORMAT (6X,5HBETAI,11X,5HBETAO,11X,6HCHORDF,11X,5HSTGRF,12X,4HFSMI
    1   ,12X,4HFSMO)
1122 FORMAT (6X,6HLAMBDA,10X,5HRVTHO,11X,6HCHORDF,10X,5HSTGRF)
1125 FORMAT (6X,6HREDFAC,10X,6HDENTOL,11X,4HSSM1,12X,4HSSM2)
1130 FORMAT(1H0,3X,3HMBI,4X,3HMBO,19X,2HMM,3X,4HNBBI,4X,
    *       3HNBL,3X,4HNRSP,3X,4HMOPT,3X,4HLOPT,3X,4HLRVB)
1140 FORMAT(39H0    BLADE SURFACE 1  --  UPPER SURFACE)
1150 FORMAT(39H0    BLADE SURFACE 2  --  LOWER SURFACE)
1180 FORMAT (7X,2HRI,I1,12X,2HRO,I1,12X,4HBETI,I1,11X,4HBETO,I1,11X,5HS
    1PLNO,I1)
1190 FORMAT (7X,3HMSP,I1,2X,5HARRAY)
1200 FORMAT (7X,4HTHSP,I1,2X,5HARRAY)
1210 FORMAT(16H0       MR  ARRAY)
1220 FORMAT (7X,11HRMSP  ARRAY)
1230 FORMAT (7X,11HBESP  ARRAY)
1235 FORMAT (7X,11HBESPF ARRAY)
1236 FORMAT (7X,11HWOWCR ARRAY)
1237 FORMAT (7X,11HPLOSS ARRAY)
1240 FORMAT(52H0    BLDAT  AANDK  ERSOR  STRFN  SLCRD  INTVL  SURVL)
1250 FORMAT (41H1 MM,NBBI,NRSP,OR SOME SPLNO IS TOO LARGE)
1260 FORMAT (56H1  WHEN REDFAC IS LESS THAN 1.0,  MOPT MUST EQUAL 1 OR
    12)
     END
     SUBROUTINE PRECAL
C
C  PRECAL CALCULATES ALL REQUIRED FIXED CONSTANTS
C
     COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
     COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
    1   LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
    2   NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
    3   INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
    4   NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
    5   PLOSS(50),MSP(50,2),THSP(50,2)
     COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
    1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
    2   WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
    3   THLE(2),RMI(2),RHO(2),BESP(50),MV(100),RM(100),BE(100),
    4   BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
    5   BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
    6   BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
    7   BEH(100,2),PLOSMH(100,2)
     COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
     LEVEL 2,  A,U,K,RHO
     COMMON /VARCOM/ RHOHB(100,2),RHOVB(100,2),WMB(100,2),WTB(100,2),
    1   WWCRM(100,2),LABEL(1,100)
     COMMON /BCDCOM/ INIT(2),EM(50,2),D2TDM2(100,2)
     COMMON /TBBC/ TBIBC,TBOBC
     COMMON/WIWO/ WISV,WOSV
     DIMENSION CURV(100,2),ZMR(100),BEZMR(100),WIMOP(4),WWCR(4),
    1   BETAIM(4),IMOP(4)
     INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
    1   UPPER,S1,ST
     REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
     EXTERNAL BL1,BL2
     DEGRAD = 180./3.1415927
     NEGBE = 0
     IERROR = 0
C
C  CALCULATE ITV, IV, TV, AND DTDMV ARRAYS
C
     ITMIN = 0
     ITMAX = NBBI-1
C  ITV UPSTREAM OF BLADE
     FIRST = 0
     LAST = NBBI-1
     DO 10 IM=1,MBIM1
     ITV(IM,1) = FIRST
  10 ITV(IM,2) = LAST
C  ITV, TV, AND DTDMV ON BLADE
     DO 20 IM=MBI,MBO
     LER(2) = 1
C    BLCD CALL NO. 1
     CALL BL1(MV(IM),TV(IM,1),DTDMV(IM,1),INF)
     ITV(IM,1) = INT((TV(IM,1)+DTLR)/HT)
     IF (TV(IM,1).GT.-DTLR) ITV(IM,1)=ITV(IM,1)+1
     ITMIN = MIN0(ITMIN,ITV(IM,1))
     LER(2) = 2
C    BLCD CALL NO. 2
     CALL BL2(MV(IM),TV(IM,2),DTDMV(IM,2),INF)
     ITV(IM,2) = INT((TV(IM,2)-DTLR)/HT)
     IF (TV(IM,2).LT.DTLR) ITV(IM,2)=ITV(IM,2)-1
  20 ITMAX = MAX0(ITMAX,ITV(IM,2))
C  ITV DOWNSTREAM OF BLADE
     LAST = ITV(MBO,2)
     FIRST = LAST+1-NBBI
     DO 40 IM=MBOP1,MM
     ITV(IM,1) = FIRST
  40 ITV(IM,2) = LAST
     ITMIN = MIN0(ITMIN,ITV(MM,1))
C  IV ARRAY
     IV(1) = 1
     DO 50 IM=1,MM
  50 IV(IM+1) = IV(IM)+ITV(IM,2)-ITV(IM,1)+1
C
C  CALCULATE BETAV AND CURV ARRAYS
C
     DO 60 SURF=1,2
     DO 60 IM=MBI,MBO
     CURV(IM,SURF) = (RM(IM)*D2TDM2(IM,SURF)+SAL(IM)*DTDMV(IM,SURF)) /
    1    (1.+(RM(IM)*DTDMV(IM,SURF))**2)**1.5
  60 BETAV(IM,SURF) = ATAN(DTDMV(IM,SURF)*RM(IM))*DEGRAD
     NIP = IV(MM)+NBBI-1
     IF(NIP.GT.2500) WRITE(NWRIT,1150)
C
C  CALCULATE MH AND DTDMH ARRAYS
C
     ITO = ITV(1,1)
     MRTS = 1
     IMS(1) = 1
     MH(1,1) = 0.
     DTDMH(1,1) = 1.E10
     LER(2) = 3
C    BLCD AND ROOT (VIA MHORIZ) CALL NO. 3
     CALL MHORIZ(MV,ITV(1,1),BL1,MBI,MBO,ITO,HT,DTLR,0,IMS(1),MH(1,1),
    1   DTDMH(1,1),MRTS)
```

```
      IF (ITV(MBO,1)-ITV(MBO,2)+NBBI.NE.2) GO TO 70
      IMSL = IMS(1)+1
      MH(IMSL,1) = MV(MBO)
      DTDMH(IMSL,1) = -1.E10
      IMS(1) = IMSL
   70 IMS(2) = 0
      MRTS = 1
      LER(2) = 4
C     BLCD AND ROOT (VIA MHORIZ) CALL NO. 4
      CALL MHORIZ(MV,ITV(1,2),BL2,MBI,MBO,ITO,HT,DTLR,1,IMS(2),MH(1,2),
     1   DTDMH(1,2),MRTS)
      IMSMAX = MAX0(IMS(1),IMS(2))
      IF (IMSMAX.GT.100) WRITE(NWRIT,1100) IMSMAX
C
C  CALCULATE RMH ARRAY
C
      DO 80 SURF=1,2
      CALL SPLINT(MR,RMSP,NRSP,MH(1,SURF),IMS(SURF),RMH(1,SURF),AAA,BBB)
   80 CALL SPLINT(MR,PLOSS,NRSP,MH(1,SURF),IMS(SURF),PLOSMH(1,SURF),
     1   AAA,BBB)
C
C  CALCULATE LAMBDA WHEN BETAI IS GIVEN AS INPUT
C
      IF (LRVB.EQ.1) GO TO 130
      BETAI = BETAI/DEGRAD
      BETAO = BETAO/DEGRAD
      IF (FSMI.NE.FSMO) GO TO 90
      FSMI = 0.
      FSMO = CHORD(1)
   90 CALL SPLINT(MR,RMSP,NRSP,FSMI,1,RMFSMI,BLANK,AAA)
      CALL SPLINT(MR,RMSP,NRSP,FSMO,1,RMFSMO,BLANK,AAA)
      CALL SPLINT(MR,BESPF,NRSP,FSMI,1,BEFSMI,BLANK,AAA)
      CALL SPLINT(MR,BESPF,NRSP,FSMO,1,BEFSMO,BLANK,AAA)
      CALL SPLINT(MR,PLOSS,NRSP,FSMI,1,PLFSMI,BLANK,AAA)
      CALL SPLINT(MR,PLOSS,NRSP,FSMO,1,PLFSMO,BLANK,AAA)
      RHOVI = WTFL/BEFSMI/PITCH/COS(BETAI)/RMFSMI
      WI = 0.
      DELWMX = SQRT(TGROG*TIP)/10.
  100 TTIP = 1.-(WI**2+2.*OMEGA*RMFSMI*WI*SIN(BETAI)+(OMEGA*RMFSMI)
     1   **2)/CPTIP
      IF (TTIP.LE.0.) GO TO 110
      TEMP = TTIP**(EXPON-1.)
      RHOIPL = RHOIP*(1.-PLFSMI)
      RHOT = RHOIPL*TEMP*TTIP
      FPRIME = RHOT-RHOIPL/GAM*WI/AR*(WI+OMEGA*RMFSMI*SIN(BETAI))/TIP
     1   *TEMP
      IF (FPRIME.LE.0.) GO TO 110
      WINEW = WI+(RHOVI-RHOT*WI)/FPRIME
      IF (WINEW-WI.GT.DELWMX) WINEW = WI+DELWMX
      IF (ABS((WINEW-WI)/WINEW).LT..000005) GO TO 120
      WI = WINEW
      GO TO 100
  110 WRITE(NWRIT,1020)
      IERROR = 1
  120 LAMBDA = RMFSMI*(WI*SIN(BETAI)+OMEGA*RMFSMI)
      WMI = WI*COS(BETAI)
      AA = (2.*OMEGA*LAMBDA-(OMEGA*RMFSMI)**2)/CPTIP
      WCRI = SQRT(TGROG*TIP*(1.-AA))
C
C   CALCULATE RVTHO WHEN BETAO IS GIVEN AS INPUT
C
      RHOVO = WTFL/BEFSMO/PITCH/COS(BETAO)/RMFSMO
      TWLMR = 2.*OMEGA*LAMBDA-(OMEGA*RMFSMO)**2
      TTIP = 1.-TWLMR/CPTIP
      RHOIPL = RHOIP*(1.-PLFSMO)
      RHOMB2 = RHOIPL*TTIP**EXPON
      LER(1) = 1
C     DENSTY CALL NO. 1
      NERT = NER(1)
      JZ = 1
      IF (FSMO.GE.SSM1.AND.FSMO.LE.SSM2) JZ=2
      CALL DENSTY(RHOVO,RHOMB2,WO,TWLMR,CPTIP,EXPON,RHOIPL,GAM,AR,TIP,
     1   JZ)
      IF (NERT.NE.NER(1)) WRITE(NWRIT,1022)
      RVTHO = RMFSMO*(WO*SIN(BETAO)+OMEGA*RMFSMO)
      WMO = WO*COS(BETAO)
      WCRO = SQRT(TGROG*TIP*(1.-TWLMR/CPTIP))
  130 TWL = 2.*OMEGA*LAMBDA
C
C  INITIALIZE DENSITY WHERE HORIZONTAL MESH LINES INTERSECT BLADE
C  AND CALCULATE BETAH ARRAY
C
      DO 150 SURF=1,2
      IMSS = IMS(SURF)
      IF (IMSS.LT.1) GO TO 150
      DO 140 IHS=1,IMSS
      RHOHB(IHS,SURF) = RHOIP*(1.-PLOSMH(IHS,SURF))*(1.-(2.*OMEGA*
     1   LAMBDA-(OMEGA*RMH(IHS,SURF))**2)/CPTIP)**EXPON
  140 BETAH(IHS,SURF) = ATAN(DTDMH(IHS,SURF)*RMH(IHS,SURF))*DEGRAD
  150 CONTINUE
C
C  INITIALIZE DENSITY WHERE VERTICAL MESH LINES INTERSECT BLADE
C  AND AT INTERIOR POINTS
C
      DO 170 IM=1,MM
      TTIP = 1.-(TWL-(OMEGA*RM(IM))**2)/CPTIP
      RHOT = RHOIP*(1.-PLOSIM(IM))*TTIP**EXPON
      DO 160 SURF=1,2
  160 RHOVB(IM,SURF) = RHOT
      IPU = IV(IM)
      IPL = IV(IM+1)-1
      DO 170 IP=IPU,IPL
  170 RHO(IP) = RHOT
C
C   CALCULATE VELOCITY DIAGRAM INFORMATION, AND
C   TAN BETA (TBI AND TBO) AT UPSTREAM AND DOWNSTREAM BOUNDARIES
C
      IMOP(1) = 1
      IMOP(2) = MBI
      IMOP(3) = MBO
      IMOP(4) = MM
      WHIRL = LAMBDA
      DELTWL = 0.
      DO 220 I=1,4
      IM = IMOP(I)
      VTHIM = WHIRL/RM(IM)
      VTHIM2 = VTHIM**2
      WTHIM = VTHIM-OMEGA*RM(IM)
      WTHTWL = VTHIM2+DELTWL
      RHOWIM = WTFL/RM(IM)/PITCH/BEF(IM)
```

```
      RHOIM = RHOVB(IM,1)
      RHOIPL = RHOIP*(1.-PLOSIM(IM))
      LER(1) = 2
C     DENSTY CALL NO. 2
      NERT = NER(1)
      JZ = 1
      CALL DENSTY(RHOWIM,RHOIM,WMIM,WTHTWL,CPTIP,EXPON,RHOIPL,GAM,AR,
     1   TIP,JZ)
      IF (NERT.NE.NER(1)) WRITE(NWRIT,1024) MV(IM),IM
      TBETA = WTHIM/WMIM
      BETAIM(I) = ATAN(TBETA)*DEGRAD
      AA = (TWL-(OMEGA*RM(IM))**2)/CPTIP
      WCRIM = SQRT(TGROG*TIP*(1.-AA))
      WIMOP(I) = SQRT(WMIM**2+WTHIM**2)
      WWCR(I) = WIMOP(I)/WCRIM
      IF (I.EQ.2) DELTWL = 2.*OMEGA*(LAMBDA-RVTHO)
      IF (I.EQ.2) WHIRL = RVTHO
  220 IF (I.EQ.1) TBI = TBETA
      TBO = TBETA
      BTAIN = ATAN(TBI)*DEGRAD
      BTAOUT = ATAN(TBO)*DEGRAD
      IF (LRVB.NE.1) GO TO 225
      FSMI = MV(1)
      FSMO = MV(MM)
      RMFSMI = RM(1)
      RMFSMO = RM(MM)
      WI = WIMOP(1)
      WMI = WI/SQRT(1.+TBI**2)
      WCRI = WI/WWCR(1)
      WO = WIMOP(4)
      WMO = WO/SQRT(1.+TBO**2)
      WCRO = WO/WWCR(4)
  225 CONTINUE
C
C   CALAULATE TAN BETA (TBIBC AND TBOBC) AT ONE-HALF MESH SPACE INSIDE
C   UPSTREAM AND DOWNSTREAM BOUNDARIES
C
      RMIM = (RM(1)+RM(2))/2.
      VTHIM = LAMBDA/RMIM
      WTHTWL = VTHIM**2
      WTHIM = VTHIM-OMEGA*RMIM
      RHOWIM = WTFL/RMIM/PITCH/(BEF(1)+BEF(2))*2.
      RHOIM = RHOVB(1,1)
      RHOIPL = RHOIP*(1.-(PLOSIM(1)+PLOSIM(2))/2.)
      LER(1) = 2
C--DENSTY CALL NO. 2-A
      NERT = NER(1)
      JZ = 1
      CALL DENSTY(RHOWIM,RHOIM,WMIM,WTHTWL,CPTIP,EXPON,RHOIPL,GAM,AR,
     1   TIP,JZ)
      IF(NERT.NE.NER(1)) WRITE(NWRIT,1026)
      TBIBC = WTHIM/WMIM
      RMIM = (RM(MM)+RM(MM-1))/2.
      VTHIM = RVTHO/RMIM
      WTHTWL = VTHIM**2+2.*OMEGA*(LAMBDA-RVTHO)
      WTHIM = VTHIM-OMEGA*RMIM
      RHOWIM = WTFL/RMIM/PITCH/(BEF(MM)+BEF(MM-1))*2.
      RHOIM = RHOVB(MM,1)
      RHOIPL = RHOIP*(1.-(PLOSIM(MM)+PLOSIM(MM-1))/2.)
      LER(1) = 2
C--DENSTY CALL NO. 2-B
      NERT = NER(1)
      JZ = 1
      CALL DENSTY(RHOWIM,RHOIM,WMIM,WTHTWL,CPTIP,EXPON,RHOIPL,GAM,AR,
     1   TIP,JZ)
      IF(NERT.NE.NER(1)) WRITE(NWRIT,1026)
      TBOBC = WTHIM/WMIM
C
C  CALCULATE REDUCED BESP WHEN REDFAC IS LESS THAN 1.0
C
      IF (REDFAC.EQ.1.) GO TO 300
C
C  CALCULATE REDUCED BESP WHEN W/WCR IS GIVEN AS INPUT
C
      IF (MOPT.NE.1) GO TO 240
      DO 230 I=1, NRSP
      ZMR(I) = MR(I)
      AA = (2.*OMEGA*LAMBDA-(OMEGA*RMSP(I))**2)/ CPTIP
      TPPRAT = (1.-AA)/(1.-REDFAC**2*AA)
      AA = (GAM-1.)/(GAM+1.)*WOWCR(I)**2
      AA = (TPPRAT*(1.-AA)/(1.-REDFAC**2*TPPRAT*AA))**EXPON
      IF (BESPF(I).LE.0.) NEGBE = 1
  230 BEZMR(I) = BESPF(I)*AA
      NZMR = NRSP
      IF (NEGBE.EQ.1) WRITE(NWRIT,1160)
      GO TO 290
C
C  CALCULATE REDUCED BESP WHEN W/WCR IS NOT GIVEN AS INPUT
C
C  CALCULATE SOLIDITY AND FAIRING DISTANCE FROM L.E. AND T.E.
  240 BLDCRD = SQRT(((RM(MBI)+RM(MBO))/2.*STGR(1))**2+CHORD(1)**2)
      SOLDTY = BLDCRD/PITCH/RM(MBI)
      DISTLE = AMIN1(.5,AMAX1(1./6.,(11.-4.*SOLDTY)/18.))*CHORD(1)
      SOLDTY = BLDCRD/PITCH/RM(MBO)
      DISTTE = AMIN1(.5,AMAX1(1./6.,(11.-4.*SOLDTY)/18.))*CHORD(1)
C  CALCULATE REDUCED BESP UPSTREAM
      RHOIM = RHOVB(1,1)
      DO 250 IM=1,MBIM1
      ZMR(IM) = MV(IM)
      VTH2IM = (LAMBDA/RM(IM))**2
      RHOWM = WTFL/RM(IM)/PITCH/BEF(IM)
      RHOIPL = RHOIP*(1.-PLOSIM(IM))
      LER(1) = 3
C     DENSTY CALL NO. 3
      NERT = NER(1)
      JZ = 1
      CALL DENSTY(RHOWM,RHOIM,WMIM,VTH2IM,CPTIP,EXPON,RHOIPL,GAM,AR,TIP,
     1   JZ)
      IF (NERT.NE.NER(1)) WRITE(NWRIT,1024) MV(IM),IM
      ENTHF = CPTIP-WMIM**2-VTH2IM
      ENTHRE = CPTIP-(WMIM**2+VTH2IM)*REDFAC**2
  250 BEZMR(IM) = BEF(IM)*(ENTHF/ENTHRE)**EXPON
C  CALCULATE REDUCED BESP IN BLADE
      IMLP = MBIM1
      IM1 = MBI+INT(DISTLE/HM1)+1
      IML = MBI+INT((CHORD(1)-DISTTE)/HM1)
      IF (IM1.GT.IML) GO TO 270
      DO 260 IM=IM1,IML
      IMLP = IMLP+1
      ZMR(IMLP) = MV(IM)
```

```
      BETIM = (BETAV(IM,1)+BETAV(IM,2))/2./DEGRAD
      RHOWIM = WTFL/BEF(IM)/RM(IM)/(TV(IM,2)-TV(IM,1))/COS(BETIM)
      RHOIPL = RHOIP*(1.-PLOSIM(IM))
      TWLMR = TWL-(OMEGA*RM(IM))**2
      LER(1) = 4
C     DENSTY CALL NO. 4
      NERT = NER(1)
      JZ = 1
      IF (IM.GE.IMS1.AND.IM.LE.IMS2) JZ=2
      CALL DENSTY(RHOWIM,RHOIM,WIM,TWLMR,CPTIP,EXPON,RHOIPL,GAM,AR,TIP,
     1   JZ)
      IF (NERT.NE.NER(1)) WRITE(NWRIT,1024) MV(IM),IM
      ENTHF = CPTIP-WIM**2-TWLMR
      ENTHRE = CPTIP-(WIM**2+TWLMR)*REDFAC**2
  260 BEZMR(IMLP) = BEF(IM)*(ENTHF/ENTHRE)**EXPON
  270 CONTINUE
C  CALCULATE REDUCED BESP DOWNSTREAM
      DO 280 IM=MBOP1,MM
      IMLP = IMLP+1
      ZMR(IMLP) = MV(IM)
      WTHTWL = (RVTHO/RM(IM))**2+2.*OMEGA*(LAMBDA-RVTHO)
      RHOWM = WTFL/RM(IM)/PITCH/BEF(IM)
      RHOIPL = RHOIP*(1.-PLOSIM(IM))
      LER(1) = 5
C     DENSTY CALL NO. 5
      NERT = NER(1)
      JZ = 1
      CALL DENSTY(RHOWM,RHOIM,WMIM,WTHTWL,CPTIP,EXPON,RHOIPL,GAM,AR,TIP,
     1   JZ)
      IF (NERT.NE.NER(1)) WRITE(NWRIT,1024) MV(IM),IM
      ENTHF = CPTIP-WMIM**2-WTHTWL
      ENTHRE = CPTIP-(WMIM**2+WTHTWL)*REDFAC**2
  280 BEZMR(IMLP) = BEF(IM)*(ENTHF/ENTHRE)**EXPON
      NZMR = IMLP
  290 CALL SPLINT(ZMR,BEZMR,NZMR,MV,MM,BE,DBDM,AAA)
C
C  CALCULATE BEH ARRAY
C
  300 DO 320 SURF=1,2
      IF (REDFAC.NE.1.) CALL SPLINT(ZMR,BEZMR,NZMR,MH(1,SURF),IMS(SURF),
     1   BEH(1,SURF),AAA,BBB)
      IF (REDFAC.EQ.1.) CALL SPLINT(MR,BESP,NRSP,MH(1,SURF),IMS(SURF),
     1   BEH(1,SURF),AAA,BBB)
      IMSS = IMS(SURF)
      IF (IMSS.LT.1) GO TO 320
      DO 310 IHS=1,IMSS
  310 IF (BEH(IHS,SURF).LE.0.) NEGBE = 2
  320 CONTINUE
C
C **********************************************************************
C  NOTE ** WTFL, OMEGA, AND LAMBDA ARE ALL REDUCED AT THIS POINT,
C  AND REMAIN REDUCED FOR THE REST OF THE PROGRAM, EXCEPT THAT
C  LAMBDA IS RESTORED TO FULL VALUE IN TVELCY
C
      ACTWT = WTFL
      ACTOMG = OMEGA
      ACTLAM = LAMBDA
      WTFL = REDFAC*WTFL
      OMEGA = REDFAC*OMEGA
      LAMBDA = REDFAC*LAMBDA
      TWL = 2.*OMEGA*LAMBDA
      TWW = 2.*OMEGA/WTFL
C
C  WRITE OUTPUT CALCULATED BY PRECAL
C
      WRITE(NWRIT,1030) WI,WMI,WCRI,BTAIN,WO,WMO,WCRO,BTAOUT,FSMI,FSMO
      WOSV=WO
      WISV=WI
      WRITE(NWRIT,1040) PITCH,HT,HM1
      WRITE(NWRIT,1050) ITMIN,ITMAX,ACTLAM,RVTHO,WTFL,NIP
      WRITE(NWRIT,1060) (IMOP(I),WIMOP(I),WWCR(I),BETAIM(I),I=1,4)
      GO TO 5000
      WRITE(NWRIT,1070)
      WRITE(NWRIT,1080) (MV(IM),RM(IM),TV(IM,1),DTDMV(IM,1),CURV(IM,1),
     1   TV(IM,2),DTDMV(IM,2),CURV(IM,2),IM=MBI,MBO)
      WRITE(NWRIT,1085) DISTLE,IM1,DISTTE,IML
      WRITE(NWRIT,1090) (IM,MV(IM),RM(IM),SAL(IM),BE(IM),DBDM(IM),
     1BEF(IM),DBFDM(IM),IM=1,MM)
      WRITE(NWRIT,1120)
      DO 330 SURF=1,2
      IMSS = IMS(SURF)
  330 WRITE(NWRIT,1130) SURF,(MH(IM,SURF),RMH(IM,SURF),BEH(IM,SURF),
     1   BETAH(IM,SURF),DTDMH(IM,SURF),IM=1,IMSS)
      WRITE(NWRIT,1110)  (IM,IV(IM),(ITV(IM,SURF),SURF=1,2),IM=1,MM)
      WRITE(NWRIT,1140)
 5000 CONTINUE
      IT = ITMIN
  340 IF (IT.GT.ITMAX) GO TO 350
      TH = FLOAT(IT)*HT
C     WRITE(NWRIT,1010) IT,TH
      IT = IT+1
      GO TO 340
C
C  STOP PROGRAM IF FATAL ERROR HAS OCCURRED IN PRECAL
C
  350 IF(NIP.GT.2500) STOP
      IF(IMSMAX.GT.100) STOP
      IF(IERROR.NE.0) STOP
      IF(NER(1).NE.0) STOP
      IF(NEGBE.NE.0) STOP
      WRITE(NWRIT,1000)
      RETURN
C
C  FORMAT STATEMENTS
C
 1000 FORMAT (1H1)
 1010 FORMAT (4X,I4,G16.5)
 1020 FORMAT(60H0INPUT WEIGHT FLOW (WTFL) IS TOO LARGE AT BLADE LEADING
     1EDGE)
 1022 FORMAT(61H0INPUT WEIGHT FLOW (WTFL) IS TOO LARGE AT BLADE TRAILING
     1 EDGE)
 1024 FORMAT(45H0INPUT WEIGHT FLOW (WTFL) IS TOO LARGE AT M =,G15.5,7H (
     1IM = ,I3,1H))
 1026 FORMAT(97H0INPUT WEIGHT FLOW (WTFL) IS TOO LARGE ONE-HALF MESH SPA
     1CE INSIDE UPSTREAM OR DOWNSTREAM BOUNDARY)
 1030 FORMAT(1H1,10X,8HRELATIVE,11X,10HMERIDIONAL,11X,8HCRITICAL,
     1   11X,9HREL. FLOW/11X,3(8HVELOCITY,12X),6H ANGLE/
     2   3(9X,11HAT M = FSMI),7X,16HAT UPSTREAM BDY./1X,4G20.5/
     3   3(9X,11HAT M = FSMO),6X,18HAT DOWNSTREAM BDY./1X,4G20.5//
     4   9X,6HFSMI =,G14.5/9X,6HFSMO =,G14.5)
```

```
1040 FORMAT(//30H  CALCULATED PROGRAM CONSTANTS//5X,5HPITCH,13X,
    1   2HHT,13X,3HHM1/1X,5G16.7)
1050 FORMAT (/5X,5HITMIN,10X,5HITMAX/4X,I5,10X,I5//5X,6HLAMBDA,12X,
    1   29H  DOWNSTREAM WHIRL (RVTHO)   /1X,G16.7,12X,G16.7/26H0     RED
    2UCED WEIGHT FLOW =,G16.7/38H0    NUMBER OF INTERIOR MESH POINTS =
    3,I5)
1060 FORMAT(40H0CALCULATED VELOCITY DIAGRAM INFORMATION/24X,2HIM,9X,1HW
    1,17X,5HW/WCR,16X,4HBETA/1X,17HUPSTREAM BOUNDARY,5X,I3,3G20.5/1X,
    212HLEADING EDGE,10X,I3,3G20.5/1X,13HTRAILING EDGE,9X,I3,3G20.5/
    31X,19HDOWNSTREAM BOUNDARY,3X,I3,3G20.5)
1070 FORMAT (1H1,6X,62HBLADE DATA AT INTERSECTIONS OF VERTICAL MESH LIN
    1ES WITH BLADES)
1080 FORMAT (/37X,15HBLADE SURFACE 1,30X,15HBLADE SURFACE 2/7X,1HM,14X,
    1   1HR,14X,2HTV,11X,5HDTDMV,11X,4HCURV,12X,2HTV,11X,5HDTDMV,11X,
    2   4HCURV/(8G15.5))
1085 FORMAT(38H0FAIRING DISTANCE FROM LEADING EDGE IS,G20.5,8H (IM1 = ,
    1I3,1H)/39H FAIRING DISTANCE FROM TRAILING EDGE IS,G20.5,8H (IML =
    2,I3,1H))
1090 FORMAT (1H1,13X,44HSTREAM SHEET COORDINATES AND THICKNESS TABLE /
    1   2X,2HIM,7X,1HM,14X,1HR,13X,3HSAL,13X,1HB,12X,5HDB/DM,14X,2HBF,
    211X,6HDBF/DM/(1X,I3,7G15.5))
1100 FORMAT(34HLONE OF THE MH ARRAYS IS TOO LARGE/7H IT HAS,I5, 8H  POI
    1NTS)
1110 FORMAT (4H1 IM,9X,8HIV ARRAY,25X,9HITV ARRAY/38X,5HBLADE/37X,7HSUR
    1FACE,3X,1H1,5X,1H2/39X,3HNO./(1X,I3,5X,I10,25X,2(I4,2X)))
1120 FORMAT (67H1M COORDINATES OF INTERSECTIONS OF HORIZONTAL MESH LINE
    1S WITH BLADE)
1130 FORMAT (25H0MH ARRAY - BLADE SURFACE,I2//15X,2HMH,19X,3HRMH,19X,
    1   3HBEH,18X,5HBETAH,17X,5HDTDMH/(5G22.4))
1140 FORMAT (43H1THETA COORDINATES OF HORIZONTAL MESH LINES//6X,2HIT,
    15X,5HTHETA)
1150 FORMAT(48H0THE NUMBER OF INTERIOR MESH POINTS EXCEEDS 2500)
1160 FORMAT(80H0THE INPUT BESP ARRAY RESULTED IN A NEGATIVE VALUE OF BE
    1 AT A VERTICAL MESH LINE/5X,47HTHE PROGRAM WILL TERMINATE AT THE E
    2ND OF PRECAL)
1170 FORMAT(82H0THE INPUT BESP ARRAY RESULTED IN A NEGATIVE VALUE OF BE
    1 AT A HORIZONTAL MESH LINE/5X,47HTHE PROGRAM WILL TERMINATE AT THE
    2 END OF PRECAL)
     END
//
```

```
      SUBROUTINE PLOTER (KK1,KK2,KK3,KK4)
      ENTRY BLDPLT
      RETURN
      ENTRY VEPLOT
      RETURN
      ENTRY TVPLOT
      RETURN
      END
      SUBROUTINE COEF
C
C  COEF CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANTS, K,
C  AT ALL UNKNOWN MESH POINTS FOR THE ENTIRE REGION
C
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
     2   NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
     3   INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
     4   NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
     5   PLOSS(50),MSP(50,2),THSP(50,2)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
     2   WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3   THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4   BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5   BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6   BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7   BEH(100,2),PLOSMH(100,2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      LEVEL 2,  A,U,K,RHO
      COMMON /HRBAAK/H(4),R(4),B(4),KAK(4),KA(4),RZ,BZ,IH(4)
      COMMON /TBBC/ TBIBC,TBOBC
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
C  INITIALIZE ARRAYS
      IH(1) = 1
      IH(2) = 0
C  INCOMPRESSIBLE CASE
      IF (GAM.NE.1.5.OR.AR.NE.1000..OR.TIP.NE.1.E6) GO TO 20
      IEND = 1
      GO TO 40
C  ADJUSTMENT OF PRINTING CONTROL VARIABLES
   20 IF(ITER.NE.1.AND.ITER.NE.2) GO TO 30
      AANDK = AANDK-1
      ERSOR = ERSOR-1
      STRFN = STRFN-1
      SLCRD = SLCRD-1
      INTVL = INTVL-1
      SURVL = SURVL-1
   30 IF(IEND.NE.0) GO TO 40
      AANDK = AANDK+2
      ERSOR = EPSOR+2
      STRFN = STRFN+2
      SLCRD = SLCRD+2
      INTVL = INTVL+2
      SURVL = SURVL+2
C
C  FIRST VERTICAL MESH LINE
C
```

```
   40 DO 50 IP=1,NBBI
      A(IP,1) = 0.
      A(IP,2) = 0.
      A(IP,3) = 0.
      A(IP,4) = 1.
   50 K(IP) = HM1*TBIBC/PITCH*2./(RM(1)+RM(2))
C
C   UPSTREAM OF BLADE, EXCEPT FOR FIRST VERTICAL MESH LINE
C
      IF (2.GT.MBIM1) GO TO 70
      DO 60 IM=2,MBIM1
   60 CALL COEFP(IM)
C
C   BETWEEN BLADES
C
   70 DO 80 IM=MBI,MBO
   80 CALL COEFB(IM)
C
C  DOWNSTREAM OF BLADES EXCEPT FOR FINAL MESH LINE
C
  150 IF (MBOP1.GT.MMM1) GO TO 170
      DO 160 IM=MBOP1,MMM1
  160 CALL COEFP(IM)
C
C  FINAL VERTICAL MESH LINE
C
  170 IVMM = IV(MM)
      DO 180 IP=IVMM,NIP
      A(IP,1) = 0.
      A(IP,2) = 0.
      A(IP,3) = 1.
      A(IP,4) = 0.
  180 K(IP) = -HM1*TBOBC/PITCH*2./(RM(MM)+RM(MM-1))
C
C  TAKE CARE OF POINTS ADJACENT TO B, AND CASES WHEN POINTS J,C,E, OR F
C  ARE GRID POINTS
C
C  POINT B
      IP = IV(MBIM1)
      A(IP,4) = 0.
C  POINT C
      IF (ITV(MBO,1)-ITV(MBO,2)+NBBI.NE.2) RETURN
      IT = ITV(MBO,1)-1
      IP = IPF(MBOP1,IT)
      A(IP,3) = 0.
      RETURN
      END
      SUBROUTINE COSUB (IM)
C
C  COSUB CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANTS, K
C  ALONG ALL VERTICAL MESH LINES WHICH INTERSECT BLADES
C
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1  LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
     2  NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
     3  INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
     4  NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
     5  PLOSS(50),MSP(50,2),THSP(50,2)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1  HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
     2  WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3  THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4  BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5  BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6  BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7  BEH(100,2),PLOSMH(100,2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      LEVEL 2,  A,U,K,RHO
      COMMON /HRBAAK/H(4),R(4),B(4),KAK(4),KA(4),RZ,BZ,IH(4)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1  UPPER,S1,ST
      INTEGER DUMM
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      DATA IAOVER/0/
C
C  COEFB CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANTS, K,
C  ALONG VERTICAL MESH LINES WHICH INTERSECT BLADES
C
      ENTRY COEFB
      IF (ITV(IM,1).GT.ITV(IM,2)) RETURN
      ITVU = ITV(IM,1)
      IT = ITVU - 1
      ITVL = ITV(IM,2)
      IPU = IPF(IM,ITVU)
      IPL = IPU+ITVL-ITVU
      DO 90 IP=IPU,IPL
      IT = IT+1
      CALL HRB (DUMM,IM,DUMM,IT,IP)
      DO 10 I=1,4
      KAK(I) = 0.
   10 KA(I) = 0
C  FIX HRB VALUES FOR CASES WHERE MESH LINES INTERSECT BLADES
   60 IF (IT.EQ.ITV(IM,1)) CALL BDRY12(1,IM,DUMM,IT)
      IF (IT.EQ.ITV(IM,2)) CALL BDRY12(2,IM,DUMM,IT)
      ITVM1 = ITV(IM-1,1)
      ITVP1 = ITV(IM+1,1)
      IF (IT.LT.ITVM1) CALL BDRY34(3,IM,1)
      IF (IT.LT.ITVP1) CALL BDRY34(4,IM,1)
      IF (IT.GT.ITV(IM-1,2)) CALL BDRY34(3,IM,2)
      IF (IT.GT.ITV(IM+1,2)) CALL BDRY34(4,IM,2)
C  COMPUTE A AND K COEFFICIENTS
   80 CALL AAK (DUMM,IM,DUMM,DUMM,IP)
      DO 90 I=1,4
      K(IP) = K(IP)+KAK(I)*A(IP,I)
      IF (KA(I).EQ.1) A(IP,I) = 0.
      IF (ABS(A(IP,I)).LE.1.) GO TO 90
      A(IP,I) = SIGN(1.,A(IP,I))
      IAOVER = IAOVER+1
      IF (IAOVER.LT.50) GO TO 90
      WRITE(NWRIT,1000)
      STOP
   90 CONTINUE
      RETURN
C
C  COEFP CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANTS, K,
C  ALONG ALL VERTICAL MESH LINES WHICH DO NOT INTERSECT BLADES
C
      ENTRY COEFP
      ITVU = ITV(IM,1)
```

```
      IT = ITVU-1
      ITVL = ITV(IM,2)
      IPL = IV(IM+1)-1
      IPU = IV(IM)
      DO 100 IP=IPU,IPL
      IT = IT+1
      CALL HRB (DUMM,IM,DUMM,IT,IP)
      IF (IT.EQ.ITVU) R(1) = RHO(IPL)
      IF (IT.EQ.ITVL) R(2) = RHO(IPU)
      CALL AAK (DUMM,IM,DUMM,DUMM,IP)
      DO 100 I=1,4
      IF(ABS(A(IP,I)).LE.1.) GO TO 100
      A(IP,I) = SIGN(1.,A(IP,I))
      IAOVER = IAOVER+1
      IF(IAOVER.LT.50) GO TO 100
      WRITE(NWRIT,1000)
      STOP
  100 CONTINUE
      K(IPL) = K(IPL)+A(IPL,2)
      K(IPU) = K(IPU)-A(IPU,1)
      RETURN
 1000 FORMAT (1H1,5X,81HPROGRAM HAS BEEN STOPPED BECAUSE 50 OF THE  A  C
     1OEFFICIENTS ARE GREATER THAN 1.0./6X,62HLOCAL SUPERSONIC FLOW MAY
     2BE THE CAUSE.  TRY A SMALLER REDFAC.)
      END
      SUBROUTINE HRBAK (I,IM,SURF,IT,IP)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1  HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
     2  WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3  THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4  BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5  BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6  BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7  BEH(100,2),PLOSMH(100,2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      LEVEL 2,  A,U,K,RHO
      COMMON /HRBAAK/H(4),R(4),B(4),KAK(4),KA(4),RZ,BZ,IH(4)
      COMMON /VARCOM/ RHOHB(100,2),RHOVB(100,2),WMB(100,2),WTB(100,2),
     1  WHCRM(100,2),LABEL(1,100)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1  UPPER,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
C
C  HRB CALCULATES MESH SPACING, H, DENSITIES, RZ AND R, AT GIVEN AND
C  ADJACENT POINTS, AND STREAM SHEET THICKNESSES, BZ AND B, AT GIVEN
C  AND ADJACENT POINTS
C
      ENTRY HRB
      H(1) = HT*RM(IM)
      H(2) = HT*RM(IM)
      H(3) = MV(IM) - MV(IM-1)
      H(4) = MV(IM+1)-MV(IM)
      RZ = RHO(IP)
      IP3 = IPF(IM-1,IT)
      IP4 = IPF(IM+1,IT)
      R(1) = RHO(IP-1)
      R(2) = RHO(IP+1)
      R(3) = RHO(IP3)
      R(4) = RHO(IP4)
      BZ = BE(IM)
      B(3) = BE(IM-1)
      B(4) = BE(IM+1)
      RETURN
C
C  AAK CALCULATES FINITE DIFFERENCE COEFFICIENTS, A, AND CONSTANT, K,
C  AT A SINGLE MESH POINT
C
      ENTRY AAK
      A12 = 2./H(1)/H(2)
      A34 = 2./H(3)/H(4)
      AZ = A12+A34
      B12 = (R(2)-R(1))/RZ/(H(1)+H(2))
      B34 = (B(4)*R(4)-B(3)*R(3))/BZ/RZ/(H(3)+H(4))-SAL(IM)/RM(IM)
      A(IP,1) = (2./H(1)+B12)/AZ/(H(1)+H(2))
      A(IP,2) = A12/AZ-A(IP,1)
      A(IP,3) = (2./H(3)+B34)/AZ/(H(3)+H(4))
      A(IP,4) = A34/AZ-A(IP,3)
      K(IP) = -TWW*BZ*RZ*SAL(IM)/AZ
      RETURN
C
C  BDRY12 CORRECTS VALUES COMPUTED BY HRB WHEN A VERTICAL MESH LINE
C  INTERSECTS A BLADE
C
      ENTRY BDRY12
      H(I) = ABS(FLOAT(IT)*HT-TV(IM,I))*RM(IM)
      R(I) = RHOVB(IM,I)
      KAK(I) = BV(I)
      KA(I) = 1
      RETURN
C
C  BDRY34 CORRECTS VALUES COMPUTED BY HRB WHEN A HORIZONTAL MESH LINE
C  INTERSECTS A BLADE
C
      ENTRY BDRY34
      IH(SURF) = IH(SURF)+1
      IHS = IH(SURF)
      H(I) = ABS(MV(IM)-MH(IHS,SURF))
      R(I) = RHOHB(IHS,SURF)
      B(I) = BEH(IHS,SURF)
      KAK(I) = BV(SURF)
      KA(I) = 1
      RETURN
      END
      SUBROUTINE SOR
C
C  SOR SOLVES THE SET OF SIMULTANEOUS EQUATIONS FOR THE STREAM FUNCTION
C  USING THE METHOD OF SUCCESSIVE OVER-RELAXATION
C
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1  LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
     2  NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
     3  INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
     4  NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
     5  PLOSS(50),MSP(50,2),THSP(50,2)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1  HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
     2  WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3  THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4  BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
```

```
     5  BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6  BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7  BEH(100,2),PLOSMH(100,2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      LEVEL 2,  A,U,K,RHO
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1  UPPER,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      AATEMP = AANDK
      IF (ORF.GE.2.) ORF=0.
      ICOUNT = 0
      IF (ORF.GT.1.) GO TO 50
      ORF = 1.
      ORFOPT = 2.
   40 ORFTEM = ORFOPT
      LMAX = 0.
   50 IF (AATEMP.GT.0) WRITE(NWRIT,1010)
      ERROR = 0.
      ICOUNT = ICOUNT+1
C
C  SOLVE MATRIX EQUATION BY SOR, OR CALCULATE OPTIMUM OVERRELAXATION
C  FACTOR
C
      IP = 0
      DO 120 IM=1,MM
      IPU = IV(IM)
      IPL = IV(IM+1)-1
      IT = ITV(IM,1)
      IF (AATEMP.GT.0) WRITE(NWRIT,1020) IM,IT
      DO 120 IP=IPU,IPL
      IP1 = IP-1
      IP2 = IP+1
C  CORRECT IP1 AND IP2 ALONG PERIODIC BOUNDARIES
      IF (IM.GE.MBI.AND.IM.LE.MBO) GO TO 60
      IF (IT.EQ.ITV(IM,1)) IP1=IP1+NBBI
      IF (IT.EQ.ITV(IM,2)) IP2=IP2-NBBI
   60 IT3 = IT
      IT4 = IT
  100 IP3 = IPF(IM-1,IT3)
      IP4 = IPF(IM+1,IT4)
      IF (ORF.GT.1.) GO TO 110
C  CALCULATE NEW ESTIMATE FOR LMAX
      UNEW = A(IP,1)*U(IP1)+A(IP,2)*U(IP2)+A(IP,3)*U(IP3)+A(IP,4)*U(IP4)
      IF (UNEW.LT.1.E-25) U(IP)=0.
      IF (U(IP).EQ.0.) GO TO 115
      RATIO = UNEW/U(IP)
      LMAX = AMAX1(RATIO,LMAX)
      U(IP) = UNEW
      GO TO 115
C  CALCULATE NEW ESTIMATE FOR STREAM FUNCTION BY SOR
  110 CHANGE = ORF*(K(IP)-U(IP)+A(IP,1)*U(IP1)+A(IP,2)*U(IP2)+A(IP,3)*
     1  U(IP3)+A(IP,4)*U(IP4))
      ERROR = AMAX1(ERROR,ABS(CHANGE))
      U(IP) = U(IP)+CHANGE
  115 IF (AATEMP.LE.0) GO TO 120
      WRITE(NWRIT,1030) IT,IP,IP1,IP2,IP3,IP4,(A(IP,I),I=1,4),K(IP)
  120 IT = IT+1
      AATEMP = 0
      IF (ORF.GT.1.) GO TO 130
      ORFOPT = 2./(1.+SQRT(ABS(1.-LMAX)))
      IF (ICOUNT.EQ.(ICOUNT/1000)*1000) WRITE(NWRIT,1000) ORFOPT
      IF (ORFTEM-ORFOPT.GT..00001.OR.ORFOPT.GT.1.99) GO TO 40
      WRITE(NWRIT,1000) ORFOPT
      WRITE(NWRIT,1070) IT
      ORF = ORFOPT
      ICOUNT = 0
      GO TO 50
  130 IF (ERSOR.GT.0) WRITE(NWRIT,1040) ERROR
      IF (ICOUNT.EQ.(ICOUNT/1000)*1000) WRITE(NWRIT,1080) ERROR,ICOUNT
      IF (ERROR.GT..00003) GO TO 50
      IF (STRFN.LE.0) RETURN
C
C  PRINT STREAM FUNCTION VALUES FOR THIS ITERATION
C
      IF (REDFAC.LT.1.) WRITE(NWRIT,1050)
      IF (REDFAC.EQ.1.) WRITE(NWRIT,1055)
      DO 140 IM=1,MM
      IPU = IV(IM)
      IPL = IV(IM+1)-1
      ITVU = ITV(IM,1)
      WRITE(NWRIT,1020) IM,ITVU
  140 WRITE(NWRIT,1060) (U(IP),IP=IPU,IPL)
      RETURN
C
C  FORMAT STATEMENTS
C
 1000 FORMAT (24H ESTIMATED OPTIMUM ORF =,F9.6)
 1010 FORMAT (82H1  IT    IP     IP1    IP2    IP3    IP4      A(1)        A(2)
     1      A(3)        A(4)        K)
 1020 FORMAT (5HKIM =,I4,6X,6HIT1 = ,I4)
 1030 FORMAT (1X,I4,5I6,5F10.5)
 1040 FORMAT (8H ERROR =,F11.8)
 1050 FORMAT (1H1,10X,44HSTREAM FUNCTION VALUES FOR REDUCED MASS FLOW)
 1055 FORMAT (1H1,10X,41HSTREAM FUNCTION VALUES FOR FULL MASS FLOW)
 1060 FORMAT (2X,10F13.8)
 1070 FORMAT(1H ,*  IT = *,I6)
 1080 FORMAT (8H ERROR =,G14.4,7H AFTER ,I6,11H ITERATIONS)
      END
      SUBROUTINE VELMER
C
C  VELMER CALLS VELPM AND VELBM TO CALCULATE RHO*W-SUB-M THROUGHOUT THE
C  REGION AND ON THE BLADE SURFACES
C
      COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1  LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
     2  NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
     3  INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
     4  NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
     5  PLOSS(50),MSP(50,2),THSP(50,2)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1  HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
     2  WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3  THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4  BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5  BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6  BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7  BEH(100,2),PLOSMH(100,2)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1  UPPER,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
```

```
C
C  CALL VELPM AND VELBM THROUGHOUT THE REGION
C
      ITVU = ITV(1,1)
      ITVL = ITV(1,2)
      DO 10 IM=1,MBIM1
   10 CALL VELPM(IM,ITVU,ITVL)
      DO 20 IM=MBI,MBO
      I = 0
   20 CALL VELBM(IM)
      ITVU = ITV(MBOP1,1)
      ITVL = ITV(MBOP1,2)
      DO 30 IM=MBOP1,MM
   30 CALL VELPM(IM,ITVU,ITVL)
      RETURN
      END
      SUBROUTINE VELSUB (IM,ITVU,ITVL)
C
C  VELSUB CALCULATES RHO*W-SUB-M THROUGHOUT THE REGION AND ON THE BLADE
C  SURFACES, AND CALCULATES THE STREAMLINE LOCATIONS
C
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
     2   NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
     3   INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
     4   NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
     5   PLOSS(50),MSP(50,2),THSP(50,2)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
     2   WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3   THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4   BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5   BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6   BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7   BEH(100,2),PLOSMH(100,2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      LEVEL 2,  A,U,K,RHO
      COMMON /VARCOM/ RHOHB(100,2),RHOVB(100,2),WMB(100,2),WTB(100,2),
     1   WWCRM(100,2),LABEL(1,100)
      DIMENSION W(2500),BETA(2500),DUDT(2500),DUDTT(2500),AAP(2500),
     1   BBP(2500)
      LEVEL 2,  W,BETA,DUDT,DUDTT,AAP,BBP
      EQUIVALENCE (A,W),(A(1,2),BETA),(A(1,3),DUDT),(A(1,4),DUDTT),
     1   (K,AAP),(RHO,BBP)
      COMMON /SLCOM/ USL(100,11),TSL(100,11)
      LEVEL 2,  USL,TSL
      COMMON /PLTCOM/ TSLPT(1100),XDOWN(400),YACROS(400)
      LEVEL 2,  TSLPT,XDOWN,YACROS
      DIMENSION TSP(51),USP(51),DDT(51),UINT(11),TINT(11)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
C
C  VELPM CALCULATES ALONG VERTICAL MESH LINES WHICH DO NOT
C  INTERSECT BLADES
C
      ENTRY VELPM
    5 NSL = 5
      NSLM1 = NSL-1
      LOC = 0
      NSP = ITVL-ITVU+2
      IP = IV(IM)-1
      DO 10 IT=1,NSP
      IP = IP+1
      TSP(IT) = FLOAT(IT+ITVU-1)*HT
   10 USP(IT) = U(IP)
      USP(NSP) = USP(1)+1.
      IP = IV(IM)
      INTU = INT(U(IP)*FLOAT(NSLM1))
      IF (U(IP).GT.0.) INTU=INTU+1
      DO 20 J=1,NSLM1
      UINT(J) = FLOAT(INTU)/FLOAT(NSLM1)
   20 INTU = INTU+1
      UINT(NSL) = UINT(1)
      CALL SPLIPR(HT,USP,NBBI,DDT,AAA)
      GO TO 100
C
C  VELBM CALCULATES ALONG VERTICAL MESHLINES WHICH INTERSECT BLADES
C
      ENTRY VELBM
      LOC = 1
      ITVUP1 = ITV(IM,1)
      ITVLM1 = ITV(IM,2)
      ITVU = ITVUP1-1
      ITVL = ITVLM1+1
      NSP = ITVL-ITVU+1
      TSP(1) = TV(IM,1)
      TSP(NSP) = TV(IM,2)
      USP(1) = BV(1)
      USP(NSP) = BV(2)
      IP = IV(IM)-1
      NSPM1 = NSP-1
      IF (2.GT.NSPM1) GO TO 70
      DO 60 IT=2,NSPM1
      IP = IP+1
      TSP(IT) = FLOAT(IT+ITVU-1)*HT
   60 USP(IT) = U(IP)
   70 DO 80 I=1,NSL
   80 UINT(I) = FLOAT(I-1)/FLOAT(NSLM1)
      CALL SPLINE(TSP,USP,NSP,DDT,AAA)
C
C  FOR VELPM AND VELBM, CALCULATE RHO*W-SUB-M IN THE REGION
C  FOR VELBM, CALCULATE RHO*W AT VERTICAL MESH LINE INTERSECTIONS WITH
C  BLADE SURFACES
C
  100 CONTINUE
      IT = LOC
      IPU = IV(IM)
      IPL = IV(IM+1)-1
      DO 110 IP=IPU,IPL
      IT = IT+1
      DUDT(IP) = DDT(IT)
  110 DUDTT(IP) = AAA(IT)
  120 IF (LOC.EQ.0) GO TO 130
      WMB(IM,1) = DDT(1)*WTFL/BE(IM)/RM(IM)
      WMB(IM,2) = DDT(NSP)*WTFL/BE(IM)/RM(IM)
      RMDTU2 = (RM(IM)*DTDMV(IM,1))**2
      RMDTL2 = (RM(IM)*DTDMV(IM,2))**2
      IF (RMDTU2.GT.10000.) WMB(IM,1) = 0.
```

```
      IF (RMDTL2.GT.10000.) WMB(IM,2) = 0.
      WMB(IM,1) = WMB(IM,1)*SQRT(1.+RMDTU2)
      WMB(IM,2) = WMB(IM,2)*SQRT(1.+RMDTL2)
  130 IF (SLCRD.LE.0) RETURN
      CALL SPLINT(USP,TSP,NSP,UINT,NSL,TINT,AAA,BBB)
      DO 140 J=1,NSL
      USL(IM,J) = UINT(J)
      TSL(IM,J) = TINT(J)
      L = (J-1)*MM+IM
  140 TSLPT(L) = TINT(J)
      RETURN
      END
      SUBROUTINE VELTAN
C
C  VELTAN CALCULATES RHO*W-SUB-THETA AND THEN RHO*W THROUGHOUT THE
C  REGION AND ON THE BLADE SURFACES,  AND CALCULATES BETA
C  THROUGHOUT THE REGION
C
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
     2   NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
     3   INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
     4   NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
     5   PLOSS(50),MSP(50,2),THSP(50,2)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
     2   WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3   THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4   BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5   BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6   BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7   BEH(100,2),PLOSMH(100,2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      LEVEL 2,  A,U,K,RHO
      COMMON /VARCOM/ RHOHB(100,2),RHOVB(100,2),WMB(100,2),WTB(100,2),
     1   WWCRM(100,2),LABEL(1,100)
      DIMENSION W(2500),BETA(2500),DUDT(2500),DUDTT(2500),AAP(2500),
     1   BBP(2500)
      LEVEL 2,  W,BETA,DUDT,DUDTT,AAP,BBP
      EQUIVALENCE (A,W),(A(1,2),BETA),(A(1,3),DUDT),(A(1,4),DUDTT),
     1   (K,AAP),(RHO,BBP)
      DIMENSION SPM(100),USP(100),DDT(100),DUDM(100),DUDMM(100),
     1   DUDTM(100),DWDM(100),WIP(100)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      EXTERNAL BL1,BL2
C
C  PERFORM CALCULATIONS ALONG ONE HORIZONTAL LINE AT A TIME
C
      IT = ITMIN
   10 IF (IT.GT.ITMAX) RETURN
      S1 = 0
C
C  ON THE GIVEN HORIZONTAL MESH LINE, FIND A FIRST POINT IN THE REGION
C
      IF (IT.GE.0.AND.IT.LT.NBBI) GO TO 60
      IM = MBIM1
   20 IM = IM+1
      IF (IM.GT.MBOP1) GO TO 200
      SURF = 1
      IF (IT.GE.ITV(IM,1).AND.IT.LT.ITV(IM-1,1)) GO TO 70
      IF (IM.EQ.MBOP1.AND.IT.EQ.ITV(MBO,1)-1.AND.ITV(MBO,1)-ITV(MBO,2)
     1   +NBBI.EQ.2) GO TO 70
      SURF = 2
      IF (IT.LE.ITV(IM,2).AND.IT.GT.ITV(IM-1,2)) GO TO 70
      GO TO 20
C
C  FIRST POINT IS ON BOUNDARY A-H
C
   60 IM1 = 1
      IM = 1
      SPM(1) = MV(1)
      USP (1) = U(IT+1)
      GO TO 90
C
C  FIRST POINT IS ON A BLADE SURFACE
C
   70 S1 = SURF
      IM1 = IM 1
      IM2 = IM
      TH = FLOAT(IT)*HT
      MVIM1 = MV(IM1)
      IF (IM.EQ.MBIP1) MVIM1 = MVIM1+(MV(IM2)-MVIM1)/1000.
      LER(2) = 5
C     BLCD (VIA ROOT) CALL NO. 5
      IF (S1.EQ.1.AND.IM1.NE.MBO) CALL ROOT(MVIM1,MV(IM2),TH,BL1,DTLR,
     1   ANS,AAA)
      LER(2) = 6
C     BLCD (VIA ROOT) CALL NO. 6
      IF (S1.EQ.2) CALL ROOT(MVIM1,MV(IM2),TH,BL2,DTLR,ANS,AAA)
      IF (S1.EQ.1.AND.IM1.EQ.MBO) ANS = MV(MBO)
      SPM(IM1) = ANS
      USP(IM1) = BV(S1)
C
C  MOVE ALONG HORIZONTAL MESH LINE UNTIL MESH LINE INTERSECTS BOUNDARY
C
   90 IF (IM.LT.MBI.OR.IM.GT.MBO) GO TO 120
      SURF = 1
      IF (IT.LT.ITV(IM,SURF).AND.IT.GE.ITV(IM-1,SURF)) GO TO 140
      SURF = 2
      IF (IT.GT.ITV(IM,SURF).AND.IT.LE.ITV(IM-1,SURF)) GO TO 140
  120 SPM(IM) = MV(IM)
      IP = IPF(IM,IT)
      USP(IM) = U(IP)
      IF (IM.EQ.MM) GO TO 130
      IM = IM+1
      GO TO 90
C
C   FINAL POINT IS ON BOUNDARY D-E
C
  130 IMT = MM
      GO TO 150
C
C  FINAL POINT IS ON A BLADE SURFACE
C
  140 ST = SURF
      IMT = IM
      IMTM1 = IMT-1
```

```
      TH = FLOAT(IT)*HT
      MVIM1 = MV(IMTM1)
      IF (IM.EQ.MBIP1) MVIM1 = MVIM1+(MV(IMT)-MVIM1)/1000.
      LER(2) = 7
C     BLCD (VIA ROOT) CALL NO. 7
      IF (ST.EQ.1.AND.IMT.NE.MBI) CALL ROOT(MVIM1,MV(IMT),TH,BL1,
     1  DTLR,ANS,AAA)
      LER(2) = 8
C     BLCD (VIA ROOT) CALL NO. 8
      IF (ST.EQ.2) CALL ROOT(MVIM1,MV(IMT),TH,BL2,DTLR,ANS,AAA)
      IF (ST.EQ.1.AND.IMT.EQ.MBI) ANS=MV(MBI)
      SPM(IMT) = ANS
      USP(IMT) = BV(ST)
  150 NSP = IMT-IM1+1
      CALL SPLINE(SPM(IM1),USP(IM1),NSP,DUDM(IM1),DUDMM(IM1))
C
C  CALCULATE RHO*W ON THE BLADE SURFACES
C
      FIRST = 1
      LAST = MM
      IF (IM1.EQ.1) GO TO 160
      FIRST = IM2
      CALL SEARCH (SPM(IM1),S1,IHS)
      ANS = DUDM(IM1)*WTFL/BEH(IHS,S1)
      IF (S1.EQ.2) ANS=-ANS
      WTB(IHS,S1) = ANS*SQRT(1.+1./(RMH(IHS,S1)*DTDMH(IHS,S1))**2)
      DDT(IM1) = -DUDM(IM1)/DTDMH(IHS,S1)
      WIP(IM1) = WTB(IHS,S1)/RHOHB(IHS,S1)
  160 IF (IMT.EQ.MM) GO TO 170
      LAST = IMTM1
      CALL SEARCH (SPM(IMT),ST,IHS)
      ANS = DUDM(IMT)*WTFL/BEH(IHS,ST)
      IF (ST.EQ.1) ANS=-ANS
      WTB(IHS,ST) = ANS*SQRT(1.+1./(RMH(IHS,ST)*DTDMH(IHS,ST))**2)
      DDT(IMT) = -DUDM(IMT)/DTDMH(IHS,ST)
      WIP(IMT) = WTB(IHS,ST)/RHOHB(IHS,ST)
C
C  CALCULATE RHO*W-SUB-THETA AND THEN RHO*W AND BETA IN THE REGION
C
  170 IF (FIRST.GT.LAST) GO TO 190
      DO 180 I=FIRST,LAST
      IP = IPF(I,IT)
      DDT(I) = DUDT(IP)
      RWM = DDT(I)/RM(I)
      RWT = -DUDM(I)
      W(IP) = SQRT(RWT**2+RWM**2)/BE(I)*WTFL
      TWLMR = 2.*OMEGA*LAMBDA-(OMEGA*RM(I))**2
      RHOIPL = RHOIP*(1.-PLOSIM(I))
      LER(1) = 6
C     DENSTY CALL NO. 6
      RHOTEM=RHO(IP)
      WTEM=W(IP)
      CALL DENSTY(WTEM,RHOTEM,ANS,TWLMR,CPTIP,EXPON,RHOIPL,GAM,AR,TIP,
     1    JZ)
      RHO(IP)=RHOTEM
      W(IP)=WTEM
      W(IP) = ANS
      WIP(I) = W(IP)
      BETA(IP) = ATAN2(RWT,RWM)*57.295779
  180 CONTINUE
      IF (IEND.LT.0) GO TO 190
      CALL SPLINE (SPM(IM1),DDT(IM1),NSP,DUDTM(IM1),AAA(IM1))
      CALL SPLINE (SPM(IM1),WIP(IM1),NSP,DWDM(IM1),AAA(IM1))
      DO 185 I=FIRST,LAST
      IP = IPF(I,IT)
      SBETA = SIN(BETA(IP)/57.295779)
      CBETA = SQRT(1.-SBETA**2)
      AAP(IP) = SBETA**2*(2.*DUDTM(I)/DUDM(I)-DUDT(IP)/DUDM(I)**2*
     1  DUDMM(I)-DUDTT(IP)/DUDT(IP))+SAL(I)*SBETA/CBETA*(1.+CBETA**2)
      BBP(IP) = RM(I)/CBETA*(2.*ACTOMG*SAL(I)+SBETA*DWDM(I)/REDFAC)
  185 CONTINUE
  190 CONTINUE
      IF (IMT.NE.MM) GO TO 20
  200 IT = IT+1
      GO TO 10
      END
      SUBROUTINE SEARCH (DIST,SURF,IS)
C
C  SEARCH LOCATES THE POSITION OF A GIVEN VALUE OF M IN THE MH ARRAY
C
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1  MM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
     2  WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3  THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4  BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5  BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6  BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7  BEH(100,2),PLOSMH(100,2)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1  UPPER,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      DO 10 I=1,100
      IF (ABS(MH(I,SURF)-DIST).GT.DMLR) GO TO 10
      IS = I
      RETURN
   10 CONTINUE
      WRITE(NWRIT,1000) DIST,SURF
      STOP
 1000 FORMAT (36HL SEARCH CANNOT FIND M IN THE MH ARRAY/7H DIST =,G14.6,
     110X,6HSURF =,G14.6)
      END
      SUBROUTINE STRLIN
C
C  STRLIN CALCULATES, PRINTS, AND PLOTS THE STREAMLINE OUTPUT DATA
C
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1  LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
     2  NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
     3  INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
     4  NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
     5  PLOSS(50),MSP(50,2),THSP(50,2)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1  MM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
     2  WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3  THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4  BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5  BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6  BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
```

```
     7   BEH(100,2),PLOSMH(100,2)
      COMMON /SLCOM/ USL(100,11),TSL(100,11)
      LEVEL 2,  USL,TSL
      COMMON /PLTCOM/ TSLPT(1100),XDOWN(400),YACROS(400)
      LEVEL 2,  TSLPT,XDOWN,YACROS
      COMMON /DUM1/ MSL(100)
      LEVEL 2,  MSL
      DIMENSION DTDM(100),D2TDM2(100),KKK(24),P(11)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      DATA KKK(4)/1H*/,KKK(6)/1H*/,KKK(8)/1H*/,KKK(10)/1H*/,KKK(12)/1H*/
     1   ,KKK(14)/1H*/,KKK(16)/1H*/,KKK(18)/1H*/,KKK(20)/1H*/,
     2   KKK(22)/1H*/,KKK(24)/1H*/
C
C  CALCULATE AND PRINT STREAMLINE OUTPUT DATA
C
      IF (SLCRD.LE.0) RETURN
      NSL = 5
      MMBL = MBO-MBI+1
      RADDEG = 180./3.1415927
      IF (REDFAC.LT.1.) WRITE(NWRIT,1000)
      IF (REDFAC.EQ.1.) WRITE(NWRIT,1010)
      DO 70 J=1,NSL
      TSLTEM=TSL(MBI,J)
      CALL SPLINE(MV(MBI),TSLTEM,MMBL,DTDM(MBI),D2TDM2(MBI))
      TSL(MBI,J)=TSLTEM
      WRITE(NWRIT,1020) J
      DO 40 IM=1,MBIM1
   40 WRITE(NWRIT,1030) IM,MV(IM),RM(IM),USL(IM,J),TSL(IM,J)
      DO 50 IM=MBI,MBO
      BETASL = ATAN(RM(IM)*DTDM(IM))*RADDEG
      CURVSL = (RM(IM)*D2TDM2(IM)+SAL(IM)*DTDM(IM))/(1.+(RM(IM)*
     1   DTDM(IM))**2)**1.5
      IF (DTDM(IM).NE.0.) DTDMN=-1./RM(IM)**2/DTDM(IM)
      IF (DTDM(IM).EQ.0.) DTDMN=1.E10
   50 WRITE(NWRIT,1040) IM,MV(IM),RM(IM),USL(IM,J),TSL(IM,J),DTDM(IM),
     1   DTDMN,BETASL,CURVSL
      DO 60 IM=MBOP1,MM
   60 WRITE(NWRIT,1030) IM,MV(IM),RM(IM),USL(IM,J),TSL(IM,J)
   70 CONTINUE
C
C  PLOT STREAMLINES
C
      KKK(1) = 7
      KKK(2) = NSL
      KKK(3) = MM
      P(1) = 1.
      P(3) = 0.
      P(4) = 0.
      DO 80 IM=1,MM
   80 MSL(IM) = MV(IM)
      IF (REDFAC.LT.1.) WRITE(NWRIT,1050)
      IF (REDFAC.EQ.1.) WRITE(NWRIT,1060)
      CALL PLOTMY(MSL,TSLPT,KKK,P)
      WRITE(NWRIT,1070)
      RETURN
C
C  FORMAT STATEMENTS
C
 1000 FORMAT (1H1,2X,44HSTREAMLINE COORDINATES FOR REDUCED MASS FLOW)
 1010 FORMAT (1H1,2X,41HSTREAMLINE COORDINATES FOR FULL MASS FLOW)
 1020 FORMAT (////4X,14HSTREAMLINE NO.,I3,23H  - WITHIN BLADE REGION//
     1   8X,2HIM,8X,1HM,14X,1HR,13X,3HUSL,12X,3HTSL,12X,4HDTDM,10X,
     2   5HDTDMN,10X,6HBETASL,9X,6HCURVSL)
 1030 FORMAT (7X,I3,2G15.5,3X,F7.3,5X,G15.5)
 1040 FORMAT (7X,I3,2G15.5,3X,F7.3,5X,3G15.5,3X,F7.2,5X,G15.5)
 1050 FORMAT (1H1,50X,38HSTREAMLINE PLOTS FOR REDUCED MASS FLOW)
 1060 FORMAT (1H1,50X,35HSTREAMLINE PLOTS FOR FULL MASS FLOW)
 1070 FORMAT (     /40X,70HSTREAMLINES ARE PLOTTED WITH THETA ACROSS THE
     1PAGE AND M DOWN THE PAGE)
      END
```

```
      SUBROUTINE OUTPUT
C
C  OUTPUT CALLS SUBROUTINES TO CALCULATE DENSITIES AND VELOCITIES
C  THROUGHOUT THE REGION AND ON THE BLADE SURFACES, AND IT PLOTS
C  THE SURFACE VELOCITIES
C
      DIMENSION PRES(100,2),PRATIO(100,2),XVDIM(100),CPT(100,2)
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
     2   NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
     3   INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
     4   NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
     5   PLOSS(50),MSP(50,2),THSP(50,2)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,THL,
     2   WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3   THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4   BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5   BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6   BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7   BEH(100,2),PLOSMH(100,2)
      COMMON /VARCOM/ RHOHB(100,2),RHOVB(100,2),WMB(100,2),WTB(100,2),
     1   WHCRM(100,2),LABEL(1,100)
      COMMON /PLTCOM/ TSLPT(1100),XDOWN(400),YACROS(400)
      LEVEL 2,  TSLPT,XDOWN,YACROS
      DIMENSION KKK(14)
      DIMENSION P(11)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      DATA KKK(4)/1H*/,KKK(6)/1HO/,KKK(8)/1H+/,KKK(10)/1HX/
C
C  CALL VELP, VELB, AND VELSUR THROUGHOUT THE REGION
C
      IF (INTVL.GT.0) CALL VELP(1,MBIM1)
      CALL VELB(MBI,MBO)
   20 IF (INTVL.GT.0) CALL VELP(MBOP1,MM)
      CALL VELSUR
C
C  PREPARE INPUT ARRAYS FOR PLOT OF VELOCITIES
C
      IF (SURVL.LE.0) RETURN
      NP2 = 0
C
C   TANGENTIAL COMPONENTS
      DO 50 SURF=1,2
      NP1 = NP2
```

```
      IMSS = IMS(SURF)
      IF (IMSS.LT.1) GO TO 40
      DO 30 IHS=1,IMSS
      IF (ABS(DTDMH(IHS,SURF)*RMH(IHS,SURF)).LT..57735) GO TO 30
      NP1 = NP1+1
      YACROS(NP1) = WTB(IHS,SURF)
      XDOWN(NP1) = MH(IHS,SURF)
   30 CONTINUE
   40 KKK(2*SURF+1) = NP1-NP2
   50 NP2 = NP1
C
C  MERIDIONAL COMPONENTS
      DO 80 SURF=1,2
      NP1 = NP2
      DO 60 IM=MBIP1,MBOM1
      IF (ABS(DTDMV(IM,SURF)*RM(IM)).GT.1.7321) GO TO 60
      NP1 = NP1+1
      YACROS(NP1) = WMB(IM,SURF)
      XDOWN(NP1) = MV(IM)
   60 CONTINUE
   70 KKK(2*SURF+5) = NP1-NP2
   80 NP2 = NP1
C
C PLOT VELOCITIES
C
      KK1 = KKK(3)
      KK2 = KKK(5)
      KK3 = KKK(7)
      KK4 = KKK(9)
      IF (BLDAT.GE.2) CALL VEPLOT(KK1,KK2,KK3,KK4)
      KKK(1) = 1
      KKK(2) = 4
      P(1)   = 5.
      IF (REDFAC.LT.1.) WRITE(NWRIT,1000)
      IF (REDFAC.EQ.1.) WRITE(NWRIT,1020)
      CALL PLOTMY(XDOWN,YACROS,KKK,P)
      WRITE(NWRIT,1010)
      PTOTAL = RHOIP*AR*TIP
      DO 90 IS= 1,2
      PRESIN = PTOTAL
      DO 90 IM = MBI,MBO
      XVDIM(IM)=MV(IM)/MV(MBO)
      TWLMI = 2.*OMEGA*LAMBDA - (OMEGA*RM(IM))**2
      PRES(IM,IS) = PTOTAL*(1.-PLOSIM(IM))*(1.-(WMB(IM,IS)**2+TWLMI)/
     1          CPTIP)**(GAM/(GAM-1.))
      PRATIO(IM,IS)=PRES(IM,IS)/PRESIN
      CPT(IM,IS)=PRATIO(IM,IS)-1.
   90 CONTINUE
      IF(REDFAC.LT.1.) WRITE (NWRIT,998)
      IF(REDFAC.EQ.1.) WRITE (NWRIT,999)
      WRITE(6,1100)
      DO 91 IM = MBI,MBO
      WRITE (6,1001)IM,MV(IM),XVDIM(IM),PRATIO(IM,1),PRATIO(IM,2),CPT(
     1IM,1),CPT(IM,2)
   91 CONTINUE
      MB2=MBI+1
      IND=MBI+MBO
      IF (IEND.LE.0) RETURN
      WRITE (9,1003) TITLEI
      WRITE (9,1002) (XVDIM(IND-IM),IM=MBI,MBO),(XVDIM(IM),IM=MB2,MBO)
      WRITE (9,1002) (   CPT(IND-IM,2),IM=MBI,MBO),
     1               (   CPT(IM,1),IM=MB2,MBO)
  998 FORMAT (1H1,50X,47HBLADE SURFACE PRESSURES FOR REDUCED WEIGHT FLOW
     1 /)
  999 FORMAT (1H1,50X,44HBLADE SURFACE PRESSURES FOR FULL WEIGHT FLOW /)
 1100 FORMAT (1H ,7X,3HIM ,8X,7H  M     ,8X,7H  M/MC  ,8X,7HP(1)/PT,
     18X,7HP(2)/PT,8X,7HCPT(1) ,8X,7HCPT(2) ,/)
 1001 FORMAT(5X,I5,6(6X,G10.4))
 1002 FORMAT (8F10.6)
 1003 FORMAT (20A4)
      RETURN
C
C  FORMAT STATEMENTS
C
 1000 FORMAT(1H1,50X48HBLADE SURFACE VELOCITIES FOR REDUCED WEIGHT FLOW)
 1010 FORMAT ( 39X    ,63HVELOCITY(W) VS. MERIDIONAL STREAMLINE DISTANCE
     1(M) DOWN THE PAGE  //
     2    52X,50H+ - BLADE SURFACE 1, BASED ON MERIDIONAL COMPONENT/
     3    52X,50H* - BLADE SURFACE 1, BASED ON TANGENTIAL COMPONENT/
     4    52X,50HX - BLADE SURFACE 2, BASED ON MERIDIONAL COMPONENT/
     5    52X,50HO - BLADE SURFACE 2, BASED ON TANGENTIAL COMPONENT)
 1020 FORMAT(1H1,50X,45HBLADE SURFACE VELOCITIES FOR FULL WEIGHT FLOW)
      END
      SUBROUTINE VEL (FIRST,LAST)
C
C VEL CALCULATES DENSITIES AND VELOCITIES FROM THE PRODUCT OF
C DENSITY TIMES VELOCITY
C
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
     2   NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
     3   INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
     4   NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
     5   PLOSS(50),MSP(50,2),THSP(50,2)
      COMMON /CALCON/ ACTHT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,THL,
     2   WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3   TLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4   BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5   BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6   BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7   BEH(100,2),PLOSMH(100,2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      LEVEL 2,  A,U,K,RHO
      COMMON /VARCOM/ RHOHB(100,2),RHOVB(100,2),WMB(100,2),WTB(100,2),
     1   WWCRM(100,2),LABEL(1,100)
      DIMENSION W(2500),BETA(2500),DUDT(2500),DUDTT(2500),AAP(2500),
     1   BBP(2500)
      LEVEL 2,  W,BETA,DUDT,DUDTT,AAP,BBP
      EQUIVALENCE (A,W),(A(1,2),BETA),(A(1,3),DUDT),(A(1,4),DUDTT),
     1   (K,AAP),(RHO,BBP)
      DIMENSION WWCRT(100,2),SURFL(100,2)
C
C VELP WRITES OUTPUT ALONG VERTICAL MESH LINES WHICH DO NOT
C INTERSECT BLADES
C
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
```

```
      DATA REDFUL/7HREDUCED/,FULL/6H  FULL/
      ENTRY VELP
      IF (REDFAC.EQ.1) REDFUL=FULL
      IF (FIRST.GT.LAST) RETURN
      IF (FIRST.EQ.1) WRITE(NWRIT,1000) REDFUL
      DO 20 IM=FIRST,LAST
      IPU = IV(IM)
      IPL = IPU+NBBI-1
      WRITE(NWRIT,1010) IM,(W(IP),BETA(IP),IP=IPU,IPL)
   20 CONTINUE
      RETURN
C
C  VELB CALCULATES ALONG VERTICAL MESH LINES WHICH INTERSECT BLADES
C
      ENTRY VELB
      IF (FIRST.GT.LAST) RETURN
      IF (FIRST.NE.MBI) GO TO 30
      RELER =  0.
      RELERA = 0.
      ZMREL = 0.
      IMREL = 0
      ITREL = 0
      ISREL = 0
      ICOUNT = 0
      SURFL(MBI,1) = 0.
      SURFL(MBI,2) = 0.
   30 DO 75 IM=FIRST,LAST
      ITVU = ITV(IM,1)
      ITVL = ITV(IM,2)
      IPUP1 = IPF(IM,ITVU)
      IPLM1 = IPF(IM,ITVL)
      TWLMR = 2.*OMEGA*LAMBDA-(OMEGA*RM(IM))**2
      WCR = SQRT(TGROG*TIP*(1.-TWLMR/CPTIP))
      IF (ITVL.LT.ITVU) GO TO 50
C  ALONG THE LINE BETWEEN BLADES
      IF (INTVL.LE.0) GO TO 50
      WRITE(NWRIT,1010) IM,(W(IP),BETA(IP),IP=IPUP1,IPLM1)
C  ON THE UPPER SURFACE
   50 RHOB = RHOVB(IM,1)
      RHOIPL = RHOIP*(1.-PLOSIM(IM))
      LER(1) = 7
C     DENSTY CALL NO. 7
      CALL DENSTY(WMB(IM,1),RHOVB(IM,1),ANS,TWLMR,CPTIP,EXPON,RHOIPL,
     1   GAM,AR,TIP,JZ)
      WMB(IM,1) = ANS
      WWCRM(IM,1) = WMB(IM,1)/WCR
      IF (IM.EQ.MBI) GO TO 60
      DELTV = TV(IM-1,1)-TV(IM,1)
      SURFL(IM,1) = SURFL(IM-1,1)+SQRT((MV(IM)-MV(IM-1))**2+
     1   (DELTV*(RM(IM)+RM(IM-1))/2.)**2)
   60 ERR = ABS((RHOB-RHOVB(IM,1))/RHOVB(IM,1))
      RELER = AMAX1(RELER,ERR)
      RELERA = RELERA+ERR
      IF (RELER.NE.ERR) GO TO 65
      IMREL = IM
      ISREL = 1
      ZMREL = MV(IM)
   65 IF (ERR.GE.DENTOL) ICOUNT=ICOUNT+1
C  ON THE LOWER SURFACE
      RHOB = RHOVB(IM,2)
      RHOIPL = RHOIP*(1.-PLOSIM(IM))
      LER(1) = 8
C     DENSTY CALL NO. 8
      CALL DENSTY(WMB(IM,2),RHOVB(IM,2),ANS,TWLMR,CPTIP,EXPON,RHOIPL,
     1   GAM,AR,TIP,JZ)
      WMB(IM,2) = ANS
      WWCRM(IM,2) = WMB(IM,2)/WCR
      IF (IM.EQ.MBI) GO TO 70
      DELTV = TV(IM-1,2)-TV(IM,2)
      SURFL(IM,2) = SURFL(IM-1,2)+SQRT((MV(IM)-MV(IM-1))**2+
     1   (DELTV*(RM(IM)+RM(IM-1))/2.)**2)
   70 ERR = ABS((RHOB-RHOVB(IM,2))/RHOVB(IM,2))
      RELER = AMAX1(RELER,ERR)
      RELERA = RELERA+ERR
      IF (RELER.NE.ERR) GO TO 75
      IMREL = IM
      ISREL = 2
      ZMREL = MV(IM)
   75 IF (ERR.GE.DENTOL) ICOUNT=ICOUNT+1
      RETURN
C
C  VELSUR CALCULATES WHERE HORIZONTAL MESH LINES INTERSECT THE BLADES
C
      ENTRY VELSUR
      ITERMX = 10
      DO 90 SURF=1,2
      IMSS = IMS(SURF)
      IF (IMSS.EQ.0) GO TO 90
      DO 80 IHS=1,IMSS
      TWLMR = 2.*OMEGA*LAMBDA-(OMEGA*RMH(IHS,SURF))**2
      WCR = SQRT(TGROG*TIP*(1.-TWLMR/CPTIP))
      RHOB = RHOHB(IHS,SURF)
      RHOIPL = RHOIP*(1.-PLOSMH(IHS,SURF))
      LER(1) = 9
C     DENSTY CALL NO. 9
      CALL DENSTY(WTB(IHS,SURF),RHOHB(IHS,SURF),ANS,TWLMR,CPTIP,
     1   EXPON,RHOIPL,GAM,AR,TIP,JZ)
      WTB(IHS,SURF) = ANS
      WWCRT(IHS,SURF) = WTB(IHS,SURF)/WCR
      ERR = ABS((RHOB-RHOHB(IHS,SURF))/RHOHB(IHS,SURF))
      RELER = AMAX1(RELER,ERR)
      RELERA = RELERA+ERR
      IF (RELER.NE.ERR) GO TO 80
      IMREL = 0
      IF (SURF.EQ.1) CALL BL1(MH(IHS,SURF),THET,DTDM,INF)
      IF (SURF.EQ.2) CALL BL2(MH(IHS,SURF),THET,DTDM,INF)
      ITREL = THET/HT+SIGN(.1,THET)
      ISREL = SURF
      ZMREL = MH(IHS,SURF)
   80 IF (ERR.GE.DENTOL) ICOUNT=ICOUNT+1
   90 CONTINUE
      IF (RELER.LT.DENTOL.OR.ITER.GE.ITERMX) IEND=IEND+1
      RELERA = RELERA/FLOAT(2*(MBO-MBI+1)+IMS(1)+IMS(2))
      WRITE(NWRIT,1080) ITER,RELER,IMREL,ITREL,ISREL,ZMREL,RELERA,ICOUNT
C
C  WRITE ALL BLADE SURFACE VELOCITIES
C
      IF (SURVL.LE.0) RETURN
      WRITE (NWRIT,1020) REDFUL
      WRITE(NWRIT,1040) (MV(IM),WMB(IM,1),BETAV(IM,1),SURFL(IM,1),
```

```
     1  WHCRM(IM,1),WMB(IM,2),BETAV(IM,2),SURFL(IM,2),WHCRM(IM,2),
     2  IM=MBI,MBO)
      WRITE (NWRIT,1050) REDFUL
      DO 100 SURF=1,2
      IMSS = IMS(SURF)
      IF (IMSS.LT.1) GO TO 100
      WRITE(NWRIT,1060) SURF
      WRITE(NWRIT,1070) (MH(IHS,SURF),WTB(IHS,SURF),BETAH(IHS,SURF),
     1  WHCRT(IHS,SURF), IHS=1,IMSS)
  100 CONTINUE
      RETURN
C
C  FORMAT STATEMENTS
C
 1000 FORMAT(1H1/40X,34HVELOCITIES AT INTERIOR MESH POINTS/45X,
     1  4HFOR ,A8,11HWEIGHT FLOW)
 1010 FORMAT(1HL,3HIM=,I3,5(24H   VELOCITY   ANGLE(DEG))/
     1(5X,5(G15.4,F9.2)))
 1020 FORMAT(1H1/16X,1H*,18X,52HSURFACE VELOCITIES BASED ON MERIDIONAL C
     1OMPONENTS - ,A8,11HWEIGHT FLOW,18X,1H*/16X,1H*,53X,1H*,53X,1H*/
     2  16X,1H*,19X,15HBLADE SURFACE 1,19X,1H*,20X,15HBLADE SURFACE 2,
     3  18X,1H*/7X,1HM,8X,1H*,2(3X,8HVELOCITY,3X,23HANGLE(DEG) SURF. LE
     4NGTH,5X,5HW/WCR,6X,1H*))
 1040 FORMAT((1H ,G13.4,3H  *,2(G12.4,F9.2,2G15.4,3H  *)))
 1050 FORMAT(1H1/3X,49HSURFACE VELOCITIES BASED ON TANGENTIAL COMPONENTS
     1  /18X,A8,11HWEIGHT FLOW)
 1060 FORMAT(//22X,15HBLADE SURFACE  ,I1/7X,1HM,10X,8HVELOCITY,3X,10HANG
     1LE(DEG),3X,5HW/WCR)
 1070 FORMAT(1H ,2G13.4,F9.2,G15.4)
 1080 FORMAT (///5X,14HITERATION  NO.,I4/5X,60HMAXIMUM RELATIVE CHANGE I
     1N DENSITY AT BLADE SURFACE POINTS =,G11.4,10H   AT IM =,I3,
     28H,   IT =,I3,10H,   SURF =,I2,7H,   M =,G11.4/5X,60HAVERAGE RELAT
     3IVE CHANGE IN DENSITY AT BLADE SURFACE POINTS =,G11.4/5X,49HNUMBER
     4 OF UNCONVERGED BLADE SURFACE MESH POINTS =,I4)
      END
      SUBROUTINE TVELCY
C
C  TVELCY SOLVES THE FULL MASS FLOW PROBLEM BY OBTAINING A
C  VELOCITY GRADIENT SOLUTION ALONG EACH VERTICAL MESH LINE
C
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1  LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
     2  NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
     3  INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
     4  NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
     5  PLOSS(50),MSP(50,2),THSP(50,2)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1  HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
     2  WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3  THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4  BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5  BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6  BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7  BEH(100,2),PLOSMH(100,2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      LEVEL 2,  A,U,K,RHO
      COMMON /VARCOM/ RHOHB(100,2),RHOVB(100,2),WMB(100,2),WTB(100,2),
     1  WHCRM(100,2),LABEL(1,100)
      COMMON /PLTCOM/ TSLPT(1100),XDOWN(400),YACROS(400)
      LEVEL 2,  TSLPT,XDOWN,YACROS
      DIMENSION W(2500),BETA(2500),DUDT(2500),DUDTT(2500),AAP(2500),
     1  BBP(2500)
      LEVEL 2,  W,BETA,DUDT,DUDTT,AAP,BBP
      EQUIVALENCE (A,W),(A(1,2),BETA),(A(1,3),DUDT),(A(1,4),DUDTT),
     1  (K,AAP),(RHO,BBP)
      DIMENSION KKK(14)
      DIMENSION P(11)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,SURFBV,
     1FIRST,UPPER,UPPRBV,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
C
C  CALL VELGRP AND VELGRB THROUGHOUT THE SOLUTION REGION
C
      LAMBDA = ACTLAM
      IF (INTVL.GT.0) WRITE(NWRIT,1000)
      IF (1.GT.MBIM1) GO TO 20
      DO 10 IM=1,MBIM1
   10 CALL VELGRP(IM)
   20 DO 30 IM=MBIP1,MBOM1
   30 CALL VELGRB(IM)
      IF (MBOP1.GT.MM) GO TO 50
      DO 40 IM=MBOP1,MM
   40 CALL VELGRP(IM)
C
C  FIX VELOCITIES ON LEADING AND TRAILING EDGE LINES
C
   50 FIRST = IV(MBI)
      LAST = IV(MBIP1)-1
      DO 54 I=FIRST,LAST
   54 W(I) = W(I)/REDFAC
      FIRST = IV(MBO)
      LAST = IV(MBOP1)-1
      DO 56 I=FIRST,LAST
   56 W(I) = W(I)/REDFAC
C
C  WRITE SURFACE VELOCITIES
C
      IF (SURVL.LE.0) RETURN
      WRITE(NWRIT,1010)
      WRITE(NWRIT,1020)(MV(IM),WMB(IM,1),WHCRM(IM,1),LABEL(1,IM),
     1  WMB(IM,2),WHCRM(IM,2),LABEL(1,IM),IM=MBIP1,MBOM1)
C
C   PREPARE ARRAYS FOR PLOT OF SURFACE VELOCITIES
C
      DO 60 IM=MBIP1,MBOM1
      I = IM-MBI
      I2 = I+MBOM1-MBI
      XDOWN(I) = MV(IM)
      YACROS(I) = WMB(IM,1)
   60 YACROS(I2) = WMB(IM,2)
      KKK(1) = 0
      KKK(2) = 2
      KKK(3) = MBOM1-MBI
      P(1)   = 1.
C
C   PLOT SURFACE VELOCITIES
C
      IF (BLDAT.GE.2) CALL TVPLOT
      WRITE(NWRIT,1030)
```

```
      CALL PLOTMY (XDOWN,YACROS,KKK,P)
      WRITE(NWRIT,1040)
      RETURN
C
C  FORMAT STATEMENTS
C
 1000 FORMAT(1H1/40X,34HVELOCITIES AT INTERIOR MESH POINTS/44X,
     1   27H(BASED ON FULL WEIGHT FLOW))
 1010 FORMAT(1H1/16X,1H*,13X,68HSURFACE VELOCITIES BASED ON MERIDIONAL C
     1OMPONENTS - FULL WEIGHT FLOW,30X,1H*/16X,1H*,55X,1H*,55X,1H*/16X,
     21H*,20X,15HBLADE SURFACE 1,20X,1H*,20X,15HBLADE SURFACE 2,20X,1H*/
     37X,1HM,8X,1H*,2(3X,8HVELOCITY,6X,5HW/WCR,33X,1H*))
 1020 FORMAT((1X,G13.4,3H  *,2(2G13.4,A8,21X,1H*)))
 1030 FORMAT(1H1,51X,24HBLADE SURFACE VELOCITIES/51X,27H(BASED ON FULL W
     1EIGHT FLOW))
 1040 FORMAT(39X,63HVELOCITY(W) VS. MERIDIONAL STREAMLINE DISTANCE(M) DO
     1WN THE PAGE//52X,19H* - BLADE SURFACE 1/52X,19H+ - BLADE SURFACE 2
     2)
      END
      SUBROUTINE VELGRA (IM)
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1   LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
     2   NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
     3   INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
     4   NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
     5   PLOSS(50),MSP(50,2),THSP(50,2)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
     2   WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3   THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4   BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5   BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6   BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7   BEH(100,2),PLOSMH(100,2)
      COMMON /AUKRHO/ A(2500,4),U(2500),K(2500),RHO(2500)
      LEVEL 2,  A,U,K,RHO
      COMMON /VARCOM/ RHOHB(100,2),RHOVB(100,2),WMB(100,2),WTB(100,2),
     1   WWCRM(100,2),LABEL(1,100)
      COMMON /BCDCOM/ INIT(2),EM(50,2),D2TDM2(100,2)
      DIMENSION W(2500),BETA(2500),DUDT(2500),DUDTT(2500),AAP(2500),
     1   BBP(2500)
      EQUIVALENCE (A,W),(A(1,2),BETA),(A(1,3),DUDT),(A(1,4),DUDTT),
     1   (K,AAP),(RHO,BBP)
      DIMENSION WGRAD(51),THETA(51),RWCB(51),CBETA(51),A2(51),B2(51)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,SURFBV,
     1FIRST,UPPER,UPPRBV,S1,ST,CHOKED,BLANK
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      DATA CHOKED/6HCHOKED/,BLANK/1H /
C
C  VELGRP SOLVES IN THE PERIODIC REGION
C
      ENTRY VELGRP
      AORB = 1.
      IP = IV(IM)
      WGRAD(1) = W(IP)/REDFAC
      IT1 = ITV(IM,1)
      IT = IT1
      NSP = NBBI+1
      GO TO 10
C
C  VELGRB SOLVES IN THE BLADE REGION
C
      ENTRY VELGRB
      IP = IV(IM)-1
      WGRAD(1) = WMB(IM,1)/REDFAC
      NSP = IV(IM+1)-IV(IM)+2
      AORB = 2.
      IT1 = ITV(IM,1)
      IT = IT1-1
   10 NSPM1 = NSP-1
      LABEL(1,IM) = BLANK
      TORSAL = 2.*ACTOMG*RM(IM)*SAL(IM)
      JZ = 1
      IF (IM.GE.IMS1.AND.IM.LE.IMS2) JZ=2
      TWLMR = 2.*ACTOMG*LAMBDA-(ACTOMG*RM(IM))**2
      WCR = SQRT(TGROG*TIP*(1.-TWLMR/CPTIP))
      DELMAX = WCR/ 10.
      TOLERC = WCR/ 100.
      DO 20 I=1,NSPM1
      CBETA(I) = COS(BETA(IP)/57.295779)
      THETA(I) = HT*FLOAT(IT)
      A2(I) = AAP(IP)
      B2(I) = BBP(IP)
      IT = IT+1
   20 IP = IP+1
      CBETA(NSP) = CBETA(1)
      A2(NSP) = A2(1)
      B2(NSP) = B2(1)
      THETA(NSP) = HT*FLOAT(IT)
      IF (AORB.LE.1.) GO TO 30
      CBETA(1) = 1./SQRT(1.+(RM(IM)*DTDMV(IM,1))**2)
      SBETA1 = SQRT(1.-CBETA(1)**2)*SIGN(1.,DTDMV(IM,1))
      A2(1) = (RM(IM)*CBETA(1))**2*D2TDM2(IM,1)+SAL(IM)*SBETA1/
     1   CBETA(1)*(1.+CBETA(1)**2)
      B2(1) = B2(2)+TORSAL*(1./CBETA(1)-1./CBETA(2))
      THETA(1) = TV(IM,1)
      CBETA(NSP) = 1./SQRT(1.+(RM(IM)*DTDMV(IM,2))**2)
      SBETAN = SQRT(1.-CBETA(NSP)**2)*SIGN(1.,DTDMV(IM,2))
      A2(NSP) = (RM(IM)*CBETA(NSP))**2*D2TDM2(IM,2)+SAL(IM)*SBETAN/
     1   CBETA(NSP)*(1.+CBETA(NSP)**2)
      B2(NSP) = B2(NSPM1)+TORSAL*(1./CBETA(NSP)-1./CBETA(NSPM1))
      THETA(NSP) = TV(IM,2)
   30 IND = 1
   40 CONTINUE
      DO 50 I=2,NSP
      WAS = WGRAD(I-1)+(A2(I-1)*WGRAD(I-1)+B2(I-1))*(THETA(I)-
     1   THETA(I-1))
      WASS = WGRAD(I-1)+(A2(I)*WAS+B2(I))*(THETA(I)-THETA(I-1))
   50 WGRAD(I) = (WAS+WASS)/2.
      DO 60 I=1,NSP
      TTIP = 1.-(WGRAD(I)**2+TWLMR)/CPTIP
      IF (TTIP.GE..0) GO TO 55
      WRITE(NWRIT,1010) IM
      IF (AORB.GT.1.) WGRAD(1) = 0.
      IF (AORB.GT.1.) WGRAD(NSP) = 0.
      GO TO 70
   55 RHOT = RHOIP*(1.-PLOSIM(IM))*TTIP**EXPON
   60 RWCB(I) = RHOT*WGRAD(I)*CBETA(I)
      CALL INTGRL (THETA,RWCB,NSP,AAA)
```

```
      WTFLES = BEF(IM)*RM(IM)*AAA(NSP)
      IF (IND.GE.6.AND.ABS(ACTWT-WTFLES).LE.ACTWT/1.E5) GO TO 70
      CALL CONTIN (WGRAD(1),WTFLES,IND,JZ,ACTWT,DELMAX)
      IF (IND.LT.10) GO TO 40
      AA = WTFLES/ACTWT
      IF (IND.EQ.10) WRITE(NWRIT,1030) IM,AA,IM
      IF (IND.EQ.10) GO TO 65
      WRITE(NWRIT,1020) IM
      IF (AORB.GT.1.) WGRAD(1) = 0.
      IF (AORB.GT.1.) WGRAD(NSP) = 0.
      GO TO 70
   65 LABEL(1,IM) = CHOKED
   70 CONTINUE
      FIRST = 1
      IF (AORB.GT.1.) FIRST = 2
      LAST = NSPM1
      IF (INTVL.GT.0) WRITE(NWRIT,1000) IM,IT1,(WGRAD(I),I=FIRST,LAST)
      IP = IV(IM)-1
      DO 80 I=FIRST,LAST
      IP = IP+1
   80 W(IP) = WGRAD(I)
      IF (AORB.LE.1.) RETURN
      WMB(IM,1) = WGRAD(1)
      WMB(IM,2) = WGRAD(NSP)
      WWCRM(IM,1) = WMB(IM,1)/WCR
      WWCRM(IM,2) = WMB(IM,2)/WCR
      RETURN
C
C     FORMAT STATEMENTS
C
 1000 FORMAT(5HKIM =,I3,10X,5HIT1 =,I3/(2X,10G13.4))
 1010 FORMAT (73HK  A VELOCITY GRADIENT SOLUTION CANNOT BE OBTAINED FOR
     1VERTICAL LINE IM =,I3)
 1020 FORMAT(92HK A VELOCITY GRADIENT SOLUTION COULD NOT BE OBTAINED IN
     150 ITERATIONS FOR VERTICAL LINE IM =,I3)
 1030 FORMAT(43HLACTWT EXCEEDS CHOKING WEIGHT FLOW FOR IM =,I3/22HKCHOKI
     1NG WEIGHT FLOW =,F6.3,3X,32H OF ACTUAL WEIGHT FLOW FOR IM = ,I3)
      END
      SUBROUTINE BLCD (M,THETA,DTDM,INF)
C
C     BLCD CALCULATES BLADE THETA COORDINATE AS A FUNCTION OF M
C
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      COMMON /INPUTT/ GAM,AR,TIP,RHOIP,WTFL,OMEGA,ORF,BETAI,BETAO,
     1 LAMBDA,RVTHO,REDFAC,DENTOL,FSMI,FSMO,SSM1,SSM2,MBI,MBO,MM,
     2 NBBI,NBL,NRSP,MOPT,LOPT,LRVB,BLDAT,AANDK,ERSOR,STRFN,SLCRD,
     3 INTVL,SURVL,CHORD(2),STGR(2),RI(2),RO(2),BETI(2),BETO(2),
     4 NSPI(2),TITLEI(20),MR(50),RMSP(50),BESPF(50),WOWCR(50),
     5 PLOSS(50),MSP(50,2),THSP(50,2)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1 HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWW,CPTIP,TGROG,TBI,TBO,TWL,
     2 WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3 THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4 BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5 BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6 BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7 BEH(100,2),PLOSMH(100,2)
      COMMON /BCDCOM/ INIT(2),EM(50,2),D2TDM2(100,2)
      DIMENSION DTDMI(2),DTDMO(2)
C
C     BL1 SOLVES FOR BLADE SURFACE 1
C
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1 UPPER,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1
      REAL M,MMLE,MSPMM,MMMSP
      ENTRY BL1
      SURF = 1
      SIGN = 1.
      GO TO 10
C
C     BL2 SOLVES FOR BLADE SURFACE 2
C
      ENTRY BL2
      SURF= 2
      SIGN =-1.
   10 INF = 0
      IM = 1
      DO 15 I=MBI,MBO
   15 IF (ABS(MV(I)-M).LE.DMLR) IM=I
      NSP = NSPI(SURF)
      IF (INIT(SURF).EQ.1) GO TO 30
      INIT(SURF) = 1
C
C     INITIAL CALCULATION OF FIRST AND LAST SPLINE POINTS ON BLADE
C
      AA = BETI(SURF)/57.295779
      AA = SIN(AA)
      MSP(1,SURF) = RI(SURF)*(1.-SIGN*AA)
      BB = SQRT(1.-AA**2)
      THSP(1,SURF) = SIGN*BB*RI(SURF)/RMI(SURF)
      DTDMI(SURF) = AA/BB/RMI(SURF)
      AA = BETO(SURF)/57.295779
      AA = SIN(AA)
      MSP(NSP,SURF) = CHORD(SURF)-RO(SURF)*(1.+SIGN*AA)
      BB = SQRT(1.-AA**2)
      THSP(NSP,SURF) = STGR(SURF)+SIGN*BB*RO(SURF)/RMO(SURF)
      DTDMO(SURF) = AA/BB/RMO(SURF)
      DO 20 IA=1,NSP
      MSP(IA,SURF) = MSP(IA,SURF)+MLE(SURF)
   20 THSP(IA,SURF) = THSP(IA,SURF)+THLE(SURF)
      CALL SPLISL(MSP(1,SURF),THSP(1,SURF),NSP,DTDMI(SURF),DTDMO(SURF),
     1 AAA,EM(1,SURF))
      IF (BLDAT.LE.0) GO TO 30
      GO TO 5000
      IF (SURF.EQ.1) WRITE(NWRIT,1000)
      WRITE(NWRIT,1010) SURF
      WRITE(NWRIT,1020) (MSP(IA,SURF),THSP(IA,SURF),AAA(IA),EM(IA,SURF),
     1 IA=1,NSP)
 5000 CONTINUE
C
C     BLADE COORDINATE CALCULATION
C
   30 KK = 2
      IF (M.GT.MSP(1,SURF)) GO TO 50
C
C     AT LEADING EDGE RADIUS
C
      MMLE = M-MLE(SURF)
      IF (MMLE.LT.-DMLR) GO TO 90
```

```
      MMLE = AMAX1(0.,MMLE)
      THETA = SQRT(MMLE*(2.*RI(SURF)-MMLE))*SIGN
      IF (THETA.EQ.0.) GO TO 40
      RMM = RI(SURF)-MMLE
      DTDM = RMM/THETA/RMI(SURF)
      THETA = THETA/RMI(SURF)
      D2TDM2(IM,SURF) = (-THETA-RMM*DTDM)/(RMI(SURF)*THETA)**2
      THETA = THETA+THLE(SURF)
      RETURN
   40 INF = 1
      DTDM = 1.E10*SIGN
      THETA = THLE(SURF)
      D2TDM2(IM,SURF) = 0.
      RETURN
C
C  ALONG SPLINE CURVE
C
   50 IF (M.LE.MSP(KK,SURF)) GO TO 60
      IF (KK.GE.NSP) GO TO 70
      KK = KK+1
      GO TO 50
   60 S = MSP(KK,SURF)-MSP(KK-1,SURF)
      EMKM1 = EM(KK-1,SURF)
      EMK = EM(KK,SURF)
      MSPMM = MSP(KK,SURF)-M
      MMMSP = M-MSP(KK-1,SURF)
      THK = THSP(KK,SURF)/S
      THKM1 = THSP(KK-1,SURF)/S
      THETA = EMKM1*MSPMM**3/6./S+EMK*MMMSP**3/6./S+(THK-EMK*S/6.)*
     1   MMMSP + (THKM1-EMKM1*S/6.)*MSPMM
      DTDM = -EMKM1*MSPMM**2/2./S+EMK*MMMSP**2/2./S+THK-THKM1-(EMK-
     1   EMKM1)*S/6.
      D2TDM2(IM,SURF) = EMKM1*MSPMM/S+EMK*MMMSP/S
      RETURN
C
C  AT TRAILING EDGE RADIUS
C
   70 CMM = CHORD(SURF)+MLE(SURF)-M
      IF (CMM.LT.-DMLR) GO TO 90
      CMM = AMAX1(0.,CMM)
      THETA = SQRT(CMM*(2.*RO(SURF)-CMM))*SIGN
      IF (THETA.EQ.0.) GO TO 80
      RMM = RO(SURF)-CMM
      DTDM = -RMM/THETA/RMO(SURF)
      THETA = THETA/RMO(SURF)
      D2TDM2(IM,SURF) = (-THETA+RMM*DTDM)/(RMO(SURF)*THETA)**2
      THETA = THETA+STGR(SURF)+THLE(SURF)
      RETURN
   80 INF = 1
      DTDM = -1.E10*SIGN
      THETA = THLE(SURF)+STGR(SURF)
      D2TDM2(IM,SURF) = 0.
      RETURN
C
C  ERROR RETURN
C
   90 WRITE(NWRIT,1030) LER(2),M,SURF
      STOP
C
C  FORMAT STATEMENTS
C
 1000 FORMAT (1H1,13X,33HBLADE DATA AT INPUT SPLINE POINTS)
 1010 FORMAT(1HL,17X,16HBLADE     SURFACE,I4)
 1020 FORMAT (7X ,1HM,10X,5HTHETA,10X,10HDERIVATIVE,5X,10H2ND DERIV. /
     1  (4G15.5) )
 1030 FORMAT (14HLBLCD CALL NO.,I3/33H M COORDINATE IS NOT WITHIN BLADE/
     14H M =,G14.6,10X,6HSURF =,G14.6)
      END
      SUBROUTINE MHORIZ(MV,ITV,BL,MBI,MBO,ITO,HT,DTLR,KODE,J,MH,DTDMH,
     1MRTS)
C
C  MHORIZ CALCULATES M COORDINATES OF INTERSECTIONS OF ALL HORIZONTAL
C  MESH LINES WITH A BLADE SURFACE
C  KODE = 0 FOR UPPER BLADE SURFACE
C  KODE = 1 FOR LOWER BLADE SURFACE
C
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      DIMENSION MV(100),ITV(100),MH(100),DTDMH(100)
      INTEGER BLDAT,AANDK,ERSOR,STRFN,SLCRD,SURVL,AATEMP,SURF,FIRST,
     1   UPPER,S1,ST
      REAL K,KAK,LAMBDA,LMAX,MH,MLE,MR,MSL,MSP,MV,MVIM1,MVIM
      EXTERNAL BL
      IF (MBI.GE.MBO) RETURN
      IM = MBI
   10 ITIND = 0
   20 IF (ITV(IM+1)-ITV(IM)-ITIND) 30,40,50
   30 J = J+1
      TI = FLOAT(ITV(IM+1)-ITO-ITIND+KODE)*HT
      ITIND = ITIND-1
      MVIM = MV(IM)
      IF (MRTS.EQ.1) MVIM=MVIM+(MV(IM+1)-MVIM)/1000.
      CALL ROOT (MVIM,MV(IM+1),TI,BL,DTLR,MH(J),DTDMH(J))
      GO TO 20
   40 IM = IM+1
      MRTS = 0
      IF (IM.EQ.MBO) RETURN
      GO TO 10
   50 J = J+1
      TI = FLOAT(ITV(IM)-ITO+ITIND+KODE)*HT
      ITIND = ITIND+1
      MVIM = MV(IM)
      IF (MRTS.EQ.1) MVIM=MVIM+(MV(IM+1)-MVIM)/1000.
      CALL ROOT(MVIM,MV(IM+1),TI,BL,DTLR,MH(J),DTDMH(J))
      GO TO 20
      END
      SUBROUTINE ROOT(A,B,Y,FUNCT,TOLERY,X,DFX)
C
C  ROOT FINDS A ROOT FOR (FUNCT MINUS Y) IN THE INTERVAL (A,B)
C
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      IERR = 0
    5 IF (IERR.EQ.1) WRITE(NWRIT,1010) A,B,Y,TOLERY
      X1 = A
      CALL FUNCT(X1,FX1,DFX,INF)
      FX1S = FX1
      DFXS = DFX
      IF (IERR.EQ.1) WRITE(NWRIT,1020) X1,FX1,DFX,INF
      X2 = B
   10 DO 30 I=1,20
      X = (X1+X2)/2.
```

```
      CALL FUNCT(X,FX,DFX,INF)
      IF (IERR.EQ.1) WRITE(NWRIT,1020) X,FX,DFX,INF
      IF ((FX1-Y)*(FX-Y).GT.0.) GO TO 20
      X2 = X
      GO TO 30
   20 X1 = X
      FX1 = FX
   30 CONTINUE
      IF (ABS(Y-FX).LT.TOLERY) RETURN
      IF (ABS(Y-FX1S).GT.1.2*TOLERY) GO TO 40
      X = A
      DFX = DFXS
      RETURN
   40 X = B
      CALL FUNCT(X,FX,DFX,INF)
      IF (ABS(Y-FX).LE.1.2*TOLERY) RETURN
      IF (IERR.EQ.1) RETURN
      WRITE(NWRIT,1000)
      IERR = 1
      GO TO 5
C
C  FORMAT STATEMENTS
C
1000  FORMAT(72H1ROOT HAS FAILED TO LOCATE A ROOT IN THE INTERVAL (A,B)
     1IN 20 ITERATIONS)
1010  FORMAT(22H ROOT ARGUMENTS -- A =,G13.5,3X,3HB =,G13.5,3X,3HY =,
     1G13.5,3X,8HTOLERY =,G13.5/16X,1HX,17X,2HFX,15X,3HDFX,10X,3HINF)
1020  FORMAT(8X,G16.5,2G18.5,I6)
      END
      SUBROUTINE DENSTY(RHOW,RHO,VEL,TWLMR,CPTIP,EXPON,RHOIP,GAM,AR,TIP,
     1   JZ)
C
C  DENSTY CALCULATES DENSITY AND VELOCITY FROM THE WEIGHT FLOW PARAMETER
C  DENSITY TIMES VELOCITY
C
      COMMON NREAD,NWRIT,ITER,IEND,LER(2),NER(2)
      VEL = RHOW/RHO
      IF (VEL.NE.0.) GO TO 10
      RHO = RHOIP*(1.-TWLMR/CPTIP)**EXPON
   10 TTIP = 1.-(VEL**2+TWLMR)/CPTIP
      IF (TTIP.LT.0.) GO TO 30
      IF (LER(1).GT.5) GO TO 25
      TEMP = TTIP**(EXPON-1.)
      RHOT = RHOIP*TEMP*TTIP
      RHOWP = -VEL**2/GAM*RHOIP/AR*TEMP/TIP+RHOT
      IF (JZ.EQ.1.AND.RHOWP.LE.0.) GO TO 30
      IF (JZ.EQ.2.AND.RHOWP.GE.0.) GO TO 30
      VELNEW = VEL+(RHOW-RHOT*VEL)/RHOWP
      VELNEW = ABS(VELNEW)
      IF (ABS(VELNEW-VEL)/VELNEW.LT..0001) GO TO 20
      VEL = VELNEW
      GO TO 10
   20 VEL = VELNEW
      RHO = RHOW/VEL
      RETURN
   25 RHO = RHOIP*TTIP**EXPON
      RETURN
   30 TGROG = 2.*GAM*AR/(GAM+1.)
      VEL = SQRT(TGROG*TIP*(1.-TWLMR/CPTIP))
      RHO = RHOIP*(1.-(VEL**2+TWLMR)/CPTIP)**EXPON
      RWMORW = RHOW/RHO/VEL
      IF (RWMORW.LT.1.) GO TO 40
      NER(1) = NER(1)+1
      WRITE(NWRIT,1000) LER(1),NER(1),RWMORW
      IF (NER(1).EQ.50) STOP
      RETURN
   40 IF (JZ.EQ.1) VEL=0.
      IF (JZ.EQ.2) VEL=VEL*1.1
      GO TO 10
 1000 FORMAT(16HLDENSTY CALL NO.,I3/9H NER(1) =,I3/10H RHO*W IS ,F7.4,
     134H TIMES THE MAXIMUM VALUE FOR RHO*W)
      END
      FUNCTION IPF(IM,IT)
      COMMON /CALCON/ ACTWT,ACTOMG,ACTLAM,MBIM1,MBIP1,MBOM1,MBOP1,MMM1,
     1   HM1,HT,DTLR,DMLR,PITCH,CP,EXPON,TWN,CPTIP,TGROG,TBI,TBO,TWL,
     2   WI,WMI,WCRI,ITMIN,ITMAX,NIP,IMS1,IMS2,IMS(2),BV(2),MLE(2),
     3   THLE(2),RMI(2),RMO(2),BESP(50),MV(100),RM(100),BE(100),
     4   BEF(100),DBDM(100),DBFDM(100),SAL(100),PLOSIM(100),AAA(100),
     5   BBB(100),IV(101),ITV(100,2),TV(100,2),DTDMV(100,2),
     6   BETAV(100,2),MH(100,2),DTDMH(100,2),BETAH(100,2),RMH(100,2),
     7   BEH(100,2),PLOSMH(100,2)
      IPF = IV(IM)+IT-ITV(IM,1)
      RETURN
      END
      SUBROUTINE CONTIN(XEST,YCALC,IND,JZ,YGIV,XDEL)
C
C--CONTIN CALCULATES AN ESTIMATE OF THE RELATIVE FLOW VELOCITY
C--FOR USE IN THE VELOCITY GRADIENT EQUATION
C
      DIMENSION X(3),Y(3)
      NCALL = NCALL+1
      IF (IND.NE.1.AND.NCALL.GT.100) GO TO 160
      GO TO (10,30,40,50,60,110,150),IND
C--FIRST CALL
   10 NCALL = 1
      XORIG = XEST
      IF (YCALC.GT.YGIV.AND.JZ.EQ.1) GO TO 20
      IND = 2
      Y(1) = YCALC
      X(1) = 0.
      XEST = XEST+XDEL
      RETURN
   20 IND = 3
      Y(3) = YCALC
      X(3) = 0.
      XEST = XEST-XDEL
      RETURN
C--SECOND CALL
   30 IND = 4
      Y(2) = YCALC
      X(2) = XEST-XORIG
      XEST = XEST+XDEL
      RETURN
   40 IND = 5
      Y(2) = YCALC
      X(2) = XEST-XORIG
      XEST = XEST-XDEL
      RETURN
C -- THIRD OR LATER CALL - FIND SUBSONIC OR SUPERSONIC SOLUTION
   50 Y(3) = YCALC
```

```
      X(3) = XEST-XORIG
      GO TO 70
   60 Y(1) = YCALC
      X(1) = XEST-XORIG
   70 IF (YGIV.LT.AMIN1(Y(1),Y(2),Y(3))) GO TO (120,130),JZ
   80 IND = 6
      CALL PABC(X,Y,APA,BPB,CPC)
      DISCR = BPB**2-4.*APA*(CPC-YGIV)
      IF (DISCR.LT.0.) GO TO 140
      IF (ABS(400.*APA*(CPC-YGIV)).LE.BPB**2) GO TO 90
      XEST = -BPB-SIGN(SQRT(DISCR),APA)
      IF (JZ.EQ.1.AND.APA.GT.0..AND.Y(3).GT.Y(1)) XEST = -BPB+
     1SQRT(DISCR)
      IF (JZ.EQ.2.AND.APA.LT.0.) XEST = -BPB-SQRT(DISCR)
      XEST = XEST/2./APA
      GO TO 100
   90 IF (JZ.EQ.2.AND.BPB.GT.0.) GO TO 130
      ACB2 = APA/BPB*(CPC-YGIV)/BPB
      IF (ABS(ACB2).LE.1.E-8) ACB2=0.
      XEST = -(CPC-YGIV)/BPB*(1.+ACB2+2.*ACB2**2)
  100 IF (XEST.GT.X(3)) GO TO 130
      IF (XEST.LT.X(1)) GO TO 120
      XEST = XEST+XORIG
      RETURN
C--FOURTH OR LATER CALL - NOT CHOKED
  110 IF(XEST-XORIG.GT.X(3)) GO TO 130
      IF(XEST-XORIG.LT.X(1)) GO TO 120
      Y(2) = YCALC
      X(2) = XEST-XORIG
      GO TO 70
C--THIRD OR LATER CALL - SOLUTION EXISTS,
C--BUT RIGHT OR LEFT SHIFT REQUIRED
  120 IND = 5
C--LEFT SHIFT
      XEST = X(1)-XDEL+XORIG
      XOSHFT = XEST-XORIG
      XORIG = XEST
      Y(3) = Y(2)
      X(3) = X(2)-XOSHFT
      Y(2) = Y(1)
      X(2) = X(1)-XOSHFT
      RETURN
  130 IND = 4
C--RIGHT SHIFT
      XEST = X(3)+XDEL+XORIG
      XOSHFT = XEST-XORIG
      XORIG = XEST
      Y(1) = Y(2)
      X(1) = X(2)-XOSHFT
      Y(2) = Y(3)
      X(2) = X(3)-XOSHFT
      RETURN
C--THIRD OR LATER CALL - APPEARS TO BE CHOKED
  140 XEST = -BPB/2./APA
      IND = 7
      IF (XEST.LT.X(1)) GO TO 120
      IF(XEST.GT.X(3)) GO TO 130
      XEST = XEST+XORIG
      RETURN
C--FOURTH OR LATER CALL - PROBABLY CHOKED
  150 IF (YCALC.GE.YGIV) GO TO 110
      IND = 10
      RETURN
C--NO SOLUTION FOUND IN 100 ITERATIONS
  160 IND = 11
      RETURN
      END
      SUBROUTINE PABC(X,Y,A,B,C)
C
C--PABC CALCULATES COEFFICIENTS A,B,C OF THE PARABOLA
C--Y=A*X**2+B*X+C, PASSING THROUGH THE GIVEN X,Y POINTS
C
      DIMENSION X(3),Y(3)
      C1 = X(3)-X(1)
      C2 = (Y(2)-Y(1))/(X(2)-X(1))
      A = (C1*C2-Y(3)+Y(1))/C1/(X(2)-X(3))
      B = C2-(X(1)+X(2))*A
      C = Y(1)-X(1)*B-X(1)**2*A
      RETURN
      END
      SUBROUTINE SPLINE (X,Y,N,SLOPE,EM)
C
C--SPLINE CALCULATES FIRST AND SECOND DERIVATIVES AT SPLINE POINTS
C--END CONDITION - SECOND DERIVATIVES AT END POINTS ARE
C--SDR1 AND SDRN TIMES SECOND DERIVATIVES AT ADJACENT POINTS
C
      COMMON NREAD,NWRIT
      DIMENSION X(N),Y(N),SLOPE(N),EM(N)
      DIMENSION G(101),SB(101)
      IERR = 0
      SDR1 = .5
      SDRN = .5
      C = X(2)-X(1)
      IF (C.EQ.0.) GO TO 50
      SB(1) = -SDR1
      G(1) = 0.
      NO = N-1
      IF (NO.LE.0) GO TO 60
      IF (NO.EQ.1) GO TO 20
      DO 10 I=2,NO
      A = C
      C = X(I+1)-X(I)
      IF (A*C.EQ.0.) GO TO 50
      IF (A*C.LT.0.) IERR = 1
      W = 2.*(A+C)-A*SB(I-1)
      SB(I) = C/W
      F = (Y(I+1)-Y(I))/C-(Y(I)-Y(I-1))/A
   10 G(I) = (6.*F-A*G(I-1))/W
   20 EM(N) = SDRN*G(N-1)/(1.+SDRN*SB(N-1))
      DO 30 I=2,N
      K = N+1-I
   30 EM(K) = G(K)-SB(K)*EM(K+1)
      SLOPE(1) = (X(1)-X(2))/6.*(2.*EM(1)+EM(2))+(Y(2)-Y(1))/(X(2)-X(1))
      DO 40 I=2,N
   40 SLOPE(I) = (X(I)-X(I-1))/6.*(2.*EM(I)+EM(I-1))+(Y(I)-Y(I-1))/
     1(X(I)-X(I-1))
      IF (IERR.EQ.0) RETURN
   50 WRITE(NWRIT,1000)
      WRITE(NWRIT,1020) N,(X(I),Y(I),I=1,N)
      IF (IERR.EQ.0) STOP
```

```
      WRITE(NWRIT,1030)
      RETURN
   60 WRITE(NWRIT,1010)
      WRITE(NWRIT,1020) N,(X(I),Y(I),I=1,N)
      STOP
 1000 FORMAT (1H1,10X,44HSPLINE ERROR -- ONE OF THREE POSSIBLE CAUSES/
     117X,51H1.  ADJACENT X POINTS ARE DUPLICATES OF EACH OTHER./
     217X,38H2.  SOME X POINTS ARE OUT OF SEQUENCE./
     317X,32H3.  SOME X POINTS ARE UNDEFINED.)
 1010 FORMAT (1H1,10X,62HSPLINE ERROR -- NUMBER OF SPLINE POINTS GIVEN I
     1S LESS THAN TWO)
 1020 FORMAT (//17X,18HNUMBER OF POINTS =,I4//17X,8HX  ARRAY,6X,8HY  ARR
     1AY/(17X,2G13.5))
 1030 FORMAT (1H1)
      END
      SUBROUTINE SPLINT (X,Y,N,Z,MAX,YINT,DYDX,D2YDX2)
C
C--SPLINT CALCULATES INTERPOLATED POINTS AND DERIVATIVES
C--FOR A SPLINE CURVE
C--END CONDITION - SECOND DERIVATIVES AT END POINTS ARE
C--SDR1 AND SDRN TIMES SECOND DERIVATIVES AT ADJACENT POINTS
C
      COMMON NREAD,NWRIT
      DIMENSION X(N),Y(N),Z(MAX),YINT(MAX),DYDX(MAX),D2YDX2(MAX)
      DIMENSION G(101),SB(101),EM(101)
      IERR = 0
      SDR1 = .5
      SDRN = .5
      TOLER= ABS(X(N)-X(1))/FLOAT(N)*1.E-5
      C = X(2)-X(1)
      IF (C.EQ.0.) GO TO 130
      SB(1) = -SDR1
      G(1) = 0.
      NO = N-1
      IF (NO.LE.0) GO TO 140
      IF (NO.EQ.1) GO TO 20
      DO 10 I=2,NO
      A = C
      C = X(I+1)-X(I)
      IF (A*C.EQ.0.) GO TO 130
      IF (A*C.LT.0.) IERR = 1
      W = 2.*(A+C)-A*SB(I-1)
      SB(I) = C/W
      F = (Y(I+1)-Y(I))/C-(Y(I)-Y(I-1))/A
   10 G(I) = (6.*F-A*G(I-1))/W
   20 EM(N) = SDRN*G(N-1)/(1.+SDRN*SB(N-1))
      DO 30 I=2,N
      K = N+1-I
   30 EM(K) = G(K)-SB(K)*EM(K+1)
      IF (MAX.LE.0) RETURN
C
      ENTRY SPLENT
      DO 120 I=1,MAX
      K=2
      IF (ABS(Z(I)-X(1)).LT.TOLER) GO TO 40
      IF (Z(I).GT.2.0*X(1)-X(2)) GO TO 50
      GO TO 80
   40 YINT(I) = Y(1)
      SK = X(K)-X(K-1)
      GO TO 110
   50 IF (ABS(Z(I)-X(K)).LT.TOLER) GO TO 60
      IF (Z(I).GT.X(K)) GO TO 70
      GO TO 100
   60 YINT(I) = Y(K)
      SK = X(K)-X(K-1)
      GO TO 110
   70 IF (K.GE.N) GO TO 90
      K = K+1
      GO TO 50
   80 S2 = X(2)-X(1)
      Y0 = EM(1)*S2**2+2.*Y(1)-Y(2)
      DYDX(I) = (Y(2)-Y(1))/S2-7.*EM(1)/6.*S2
      YINT(I) = Y0+DYDX(I)*(Z(I)-X(1)+S2)
      D2YDX2(I) = 0.
      GO TO 120
   90 IF (Z(I).LT.2.*X(N)-X(N-1)) GO TO 100
      SN = X(N)-X(N-1)
      YNP1 = EM(N)*SN**2+2.*Y(N)-Y(N-1)
      DYDX(I) = (Y(N)-Y(N-1))/SN+7.*EM(N)/6.*SN
      YINT(I) = YNP1+DYDX(I)*(Z(I)-X(N)-SN)
      D2YDX2(I) = 0.
      GO TO 120
  100 SK = X(K)-X(K-1)
      YINT(I) = EM(K-1)*(X(K)-Z(I))**3/6./SK  +EM(K)*(Z(I)-X(K-1))**3/6.
     1  /SK+(Y(K)/SK  -EM(K)*SK  /6.)*(Z(I)-X(K-1))+(Y(K-1)/SK  -EM(K-1)
     2  *SK/6.)*(X(K)-Z(I))
  110 DYDX(I)=-EM(K-1)*(X(K)-Z(I))**2/2.0/SK  +EM(K)*(X(K-1)-Z(I))**2/2.
     1   /SK+(Y(K)-Y(K-1))/SK  -(EM(K)-EM(K-1))*SK/6.
      D2YDX2(I) = EM(K)-(X(K)-Z(I))/SK*(EM(K)-EM(K-1))
  120 CONTINUE
      IF (IERR.EQ.0) RETURN
  130 WRITE(NWRIT,1000)
      WRITE(NWRIT,1020) N,(X(I),Y(I),I=1,N)
      IF (IERR.EQ.0) STOP
      WRITE(NWRIT,1030)
      RETURN
  140 WRITE(NWRIT,1010)
      WRITE(NWRIT,1020) N,(X(I),Y(I),I=1,N)
      STOP
 1000 FORMAT (1H1,10X,44HSPLINT ERROR -- ONE OF THREE POSSIBLE CAUSES/
     117X,51H1.  ADJACENT X POINTS ARE DUPLICATES OF EACH OTHER./
     217X,38H2.  SOME X POINTS ARE OUT OF SEQUENCE./
     317X,32H3.  SOME X POINTS ARE UNDEFINED.)
 1010 FORMAT (1H1,10X,62HSPLINT ERROR -- NUMBER OF SPLINE POINTS GIVEN I
     1S LESS THAN TWO)
 1020 FORMAT (//17X,18HNUMBER OF POINTS =,I4//17X,8HX  ARRAY,6X,8HY  ARR
     1AY/(17X,2G13.5))
 1030 FORMAT (1H1)
      END
      SUBROUTINE SPLISL(X,Y,N,Y1P,YNP,SLOPE,EM)
C
C--SPLISL CALCULATES FIRST AND SECOND DERIVATIVES AT SPLINE POINTS
C--END CONDITION - FIRST DERIVATIVES SPECIFIED AT END POINTS
C
      COMMON NREAD,NWRIT
      DIMENSION X(N),Y(N),SLOPE(N),EM(N)
      DIMENSION G(101),SB(101)
      IERR = 0
      C = X(2)-X(1)
      IF (C.EQ.0.) GO TO 50
```

```
      SB(1) = .5                                                      SPLISL13
      F = (Y(2)-Y(1))/C-Y1P                                           SPLISL14
      G(1) = 3.*F/C                                                   SPLISL15
      NO = N-1                                                        SPLISL16
      IF (NO.LE.0) GO TO 60                                           SPLISL17
      IF (NO.EQ.1) GO TO 20                                           SPLISL18
      DO 10 I=2,NO                                                    SPLISL19
      A = C                                                           SPLISL20
      C = X(I+1)-X(I)                                                 SPLISL21
      IF (A*C.EQ.0.) GO TO 50                                         SPLISL22
      IF (A*C.LT.0.) IERR = 1                                         SPLISL23
      W = 2.*(A+C)-A*SB(I-1)                                          SPLISL24
      SB(I) = C/W                                                     SPLISL25
      F = (Y(I+1)-Y(I))/C-(Y(I)-Y(I-1))/A                             SPLISL26
   10 G(I) = (6.*F-A*G(I-1))/W                                        SPLISL27
   20 W = C*(2.-SB(N-1))                                              SPLISL28
      F = YNP-(Y(N)-Y(N-1))/C                                         SPLISL29
      EM(N) = (6.*F-C*G(N-1))/W                                       SPLISL30
      DO 30 I=2,N                                                     SPLISL31
      K = N+1-I                                                       SPLISL32
   30 EM(K) = G(K)-SB(K)*EM(K+1)                                      SPLISL33
      SLOPE(1) = Y1P                                                  SPLISL34
      DO 40 I=2,NO                                                    SPLISL35
   40 SLOPE(I) = (X(I)-X(I-1))/6.*(2.*EM(I)+EM(I-1))+(Y(I)-Y(I-1))/   SPLISL36
     1(X(I)-X(I-1))                                                   SPLISL37
      SLOPE(N) = YNP                                                  SPLISL38
      IF (IERR.EQ.0) RETURN                                           SPLISL39
   50 WRITE(NWRIT,1000)                                               SPLISL40
      WRITE(NWRIT,1020) N,(X(I),Y(I),I=1,N)                           SPLISL41
      IF (IERR.EQ.0) STOP                                             SPLISL42
      WRITE(NWRIT,1030)                                               SPLISL43
      RETURN                                                          SPLISL44
   60 WRITE(NWRIT,1010)                                               SPLISL45
      WRITE(NWRIT,1020) N,(X(I),Y(I),I=1,N)                           SPLISL46
      STOP                                                            SPLISL47
 1000 FORMAT (1H1,10X,44HSPLISL ERROR -- ONE OF THREE POSSIBLE CAUSES/ SPLISL48
     117X,51H1.  ADJACENT X POINTS ARE DUPLICATES OF EACH OTHER./     SPLISL49
     217X,38H2.  SOME X POINTS ARE OUT OF SEQUENCE./                  SPLISL50
     317X,32H3.  SOME X POINTS ARE UNDEFINED.)                        SPLISL51
 1010 FORMAT (1H1,10X,62HSPLISL ERROR -- NUMBER OF SPLINE POINTS GIVEN ISPLISL52
     1S LESS THAN TWO)                                                SPLISL53
 1020 FORMAT (//17X,18HNUMBER OF POINTS =,I4//17X,8HX  ARRAY,6X,8HY  ARRSPLISL54
     1AY/(17X,2G13.5))                                                SPLISL55
 1030 FORMAT (1H1)                                                    SPLISL56
      END                                                             SPLISL57
      SUBROUTINE SPLIPR(DX,Y,N,SLOPE,EM)                              SPLIPR 2
C                                                                     SPLIPR 3
C--SPLIPR CALULATES FIRST AND SECOND DERIVATIVES AT EQUALLY SPACED    SPLIPR 4
C--SPLINE POINTS, USING SOR                                           SPLIPR 5
C--END CONDITIONS - FIRST AND SECOND DERIVATIVES ARE PERIODIC, AND    SPLIPR 6
C--Y INCREASES BY ONE IN THIS PERIOD                                  SPLIPR 7
C                                                                     SPLIPR 8
      COMMON NREAD,NWRIT                                              SPLIPR 9
      DIMENSION Y(N),SLOPE(N),EM(N)                                   SPLIPR10
      DIMENSION F(101)                                                SPLIPR11
      IF(N.LT.2) GO TO 50                                             SPLIPR12
      NO = N-1                                                        SPLIPR13
      DYP = 1.+Y(1)-Y(N)                                              SPLIPR14
      CONST = 3./2./DX**2                                             SPLIPR15
      DO 10 I=1,NO                                                    SPLIPR16
      EM(I) = 0.                                                      SPLIPR17
      DYM = DYP                                                       SPLIPR18
      DYP = Y(I+1)-Y(I)                                               SPLIPR19
   10 F(I) = CONST*(DYP-DYM)                                          SPLIPR20
      EM(N) = 0.                                                      SPLIPR21
      DYM = DYP                                                       SPLIPR22
      DYP = 1.+Y(1)-Y(N)                                              SPLIPR23
      F(N) = CONST*(DYP-DYM)                                          SPLIPR24
      ORF = 2./(1.+SQRT(.75))                                         SPLIPR25
      DO 30 K=1,10                                                    SPLIPR26
      EM(1) = EM(1)-ORF*((EM(N)+EM(2))/4.+EM(1)-F(1))                 SPLIPR27
      IF(NO.EQ.1) GO TO 30                                            SPLIPR28
      DO 20 I=2,NO                                                    SPLIPR29
   20 EM(I) = EM(I)-ORF*((EM(I-1)+EM(I+1))/4.+EM(I)-F(I))             SPLIPR30
   30 EM(N) = EM(N)-ORF*((EM(N-1)+EM(1))/4.+EM(N)-F(N))               SPLIPR31
      SLOPE(1) = -DX/6.*(2.*EM(1)+EM(2))+(Y(2)-Y(1))/DX               SPLIPR32
      DO 40 I=2,N                                                     SPLIPR33
   40 SLOPE(I) = DX/6.*(2.*EM(I)-EM(I-1))+(Y(I)-Y(I-1))/DX            SPLIPR34
      RETURN                                                          SPLIPR35
   50 WRITE(NWRIT,1000)                                               SPLIPR36
      STOP                                                            SPLIPR37
 1000 FORMAT (1H1,10X,62HSPLIPR ERROR -- NUMBER OF SPLINE POINTS GIVEN ISPLIPR38
     1S LESS THAN TWO)                                                SPLIPR39
      END                                                             SPLIPR40
      SUBROUTINE INTGRL(X,Y,N,SUM)                                    INTGRL 2
C                                                                     INTGRL 3
C--INTGRL CALCULATES THE INTEGRAL OF A SPLINE CURVE PASSING THROUGH   INTGRL 4
C--A GIVEN SET OF POINTS                                              INTGRL 5
C--END CONDITION - SECOND DERIVATIVES AT END POINTS ARE               INTGRL 6
C--SDR1 AND SDRN TIMES SECOND DERIVATIVES AT ADJACENT POINTS          INTGRL 7
C                                                                     INTGRL 8
      COMMON NREAD,NWRIT                                              INTGRL 9
      DIMENSION X(N),Y(N),SUM(N)                                      INTGRL10
      DIMENSION G(101),SB(101),EM(101)                                INTGRL11
      IERR = 0                                                        INTGRL12
      SDR1 = .5                                                       INTGRL13
      SDRN = .5                                                       INTGRL14
      C = X(2)-X(1)                                                   INTGRL15
      IF (C.EQ.0.) GO TO 50                                           INTGRL16
      SB(1) = -SDR1                                                   INTGRL17
      G(1) = 0.                                                       INTGRL18
      NO = N-1                                                        INTGRL19
      IF (NO.LE.0) GO TO 60                                           INTGRL20
      IF (NO.EQ.1) GO TO 20                                           INTGRL21
      DO 10 I=2,NO                                                    INTGRL22
      A = C                                                           INTGRL23
      C = X(I+1)-X(I)                                                 INTGRL24
      IF (A*C.EQ.0.) GO TO 50                                         INTGRL25
      IF (A*C.LT.0.) IERR = 1                                         INTGRL26
      W = 2.*(A+C)-A*SB(I-1)                                          INTGRL27
      SB(I) = C/W                                                     INTGRL28
      F = (Y(I+1)-Y(I))/C-(Y(I)-Y(I-1))/A                             INTGRL29
   10 G(I) = (6.*F-A*G(I-1))/W                                        INTGRL30
   20 EM(N) = SDRN*G(N-1)/(1.+SDRN*SB(N-1))                           INTGRL31
      DO 30 I=2,N                                                     INTGRL32
      K = N+1-I                                                       INTGRL33
   30 EM(K) = G(K)-SB(K)*EM(K+1)                                      INTGRL34
      SUM(1) = 0.                                                     INTGRL35
      DO 40 I=2,N                                                     INTGRL36
   40 SUM(I) = SUM(I-1)+(X(I)-X(I-1))*(Y(I)+Y(I-1))/2.-(X(I)-X(I-1))**3 INTGRL37
```

```
     1*(EM(I)+EM(I-1))/24.                                              INTGRL38
      IF (IERR.EQ.0) RETURN                                             INTGRL39
   50 WRITE(NWRIT,1000)                                                 INTGRL40
      WRITE(NWRIT,1020) N,(X(I),Y(I),I=1,N)                             INTGRL41
      IF (IERR.EQ.0) STOP                                               INTGRL42
      WRITE(NWRIT,1030)                                                 INTGRL43
      RETURN                                                            INTGRL44
   60 WRITE(NWRIT,1010)                                                 INTGRL45
      WRITE(NWRIT,1020) N,(X(I),Y(I),I=1,N)                             INTGRL46
      STOP                                                              INTGRL47
 1000 FORMAT (1H1,10X,44HINTGRL ERROR -- ONE OF THREE POSSIBLE CAUSES/  INTGRL48
     117X,51H1.  ADJACENT X POINTS ARE DUPLICATES OF EACH OTHER./       INTGRL49
     217X,38H2.  SOME X POINTS ARE OUT OF SEQUENCE./                    INTGRL50
     317X,32H3.  SOME X POINTS ARE UNDEFINED.)                          INTGRL51
 1010 FORMAT (1H1,10X,62HINTGRL ERROR -- NUMBER OF SPLINE POINTS GIVEN IINTGRL52
     1S LESS THAN TWO)                                                  INTGRL53
 1020 FORMAT (//17X,18HNUMBER OF POINTS =,I4//17X,8HX  ARRAY,6X,8HY  ARRINTGRL54
     1AY/(17X,2G13.5))                                                  INTGRL55
 1030 FORMAT (1H1)                                                      INTGRL56
      END                                                               INTGRL57
      SUBROUTINE PLOTMY(X,Y,K,P)                                        PLOTMY 2
C--PLOTMY PLOTS MULTIPLE CURVES                                         PLOTMY 3
      COMMON NREAD,NWRIT                                                PLOTMY 4
      LOGICAL FORY,STUG,TONLY                                           PLOTMY 5
      DIMENSION YLABEL(11),A(104),KPCSTD(6),XLAB(8),YLAB(6),FKFD(6)     PLOTMY 6
      DIMENSION X(1),Y(1),P(1),K(1)                                     PLOTMY 7
      LEVEL 2,  X,Y                                                     PLOTMY 8
      COMMON /JOLO/ F,DX,TLINX,N,LABOUT,KFD,FORY,STUG,TONLY             PLOTMY 9
      EQUIVALENCE (KPC,TPC)                                             PLOTMY10
      DATA BLANK,XGRID,YGRID,RMARK /1H ,1H-,1H1,1H=/                    PLOTMY11
      DATA KPCSTD /1H*,1H+,1HO,1HX,1H=,1HO  /                           PLOTMY12
      DATA XLAB/4H (1H,4H  ,5,4HX, F,4H10.0,4H,104,4HA1,F,4H10.0,4H)   /PLOTMY13
      DATA YLAB/4H (1H,4H  , ,4H8X, ,4H 11F,4H10.0,4H)    /             PLOTMY14
      DATA FKFD/4H10.1,4H10.2,4H10.3,4H10.4,4H10.5,4H10.6/              PLOTMY15
C                                                                       PLOTMY16
  100 WRITE(NWRIT,1000)                                                 PLOTMY17
      KODE = K(1)                                                       PLOTMY18
      KN = K(2)                                                         PLOTMY19
      NPTS = K(3)                                                       PLOTMY20
      LABOUT = 1                                                        PLOTMY21
      KTL = 1                                                           PLOTMY22
      FKFDO = XLAB(4)                                                   PLOTMY23
      IF(P(1).GT.2.) GO TO 110                                          PLOTMY24
C--P(1) = 1. (DUPX)                                                     PLOTMY25
      KTL = KN                                                          PLOTMY26
      GO TO 130                                                         PLOTMY27
  110 IF(P(1).GT.4.) GO TO 140                                          PLOTMY28
C--P(1) = 3. (DUPY)                                                     PLOTMY29
      KTIMES = KN-1                                                     PLOTMY30
      DO 120 I=1,KTIMES                                                 PLOTMY31
      MM = I*NPTS                                                       PLOTMY32
      K(2*I+3) = NPTS                                                   PLOTMY33
      DO 120 II=1,NPTS                                                  PLOTMY34
      L = MM+II                                                         PLOTMY35
  120 Y(L) = Y(II)                                                      PLOTMY36
C--P(1) = 1. OR 3. (DUPX OR DUPY)                                       PLOTMY37
  130 NPTST = KN*NPTS                                                   PLOTMY38
      TLINX = 55*(1+NPTS/35)                                            PLOTMY39
      GO TO 160                                                         PLOTMY40
C--P(1) = 5. (NO DUP)                                                   PLOTMY41
  140 NPTST = 0                                                         PLOTMY42
      DO 150 I=1,KN                                                     PLOTMY43
  150 NPTST = NPTST+K(2*I+1)                                            PLOTMY44
      TLINX = 55*(1+NPTST/(35*KN))                                      PLOTMY45
  160 IF(MOD(KODE,2).NE.0) GO TO 180                                    PLOTMY46
C--OPTION 1 NOT CHOSEN                                                  PLOTMY47
      DO 170 I=1,KN                                                     PLOTMY48
  170 K(2*I+2) = KPCSTD(I)                                              PLOTMY49
  180 NX = 10                                                           PLOTMY50
C--OPTION 2                                                             PLOTMY51
      IF(MOD(KODE/2,2).NE.0) NX = P(3)                                  PLOTMY52
      IF (NX.EQ.0) NX = 1000                                            PLOTMY53
      NY = 10                                                           PLOTMY54
C--OPTION 4                                                             PLOTMY55
      IF(MOD(KODE/4,2).NE.0) NY = P(4)                                  PLOTMY56
      IF(NY.EQ.0) NY = 100                                              PLOTMY57
C--OPTION 8 NOT AVAILABLE                                               PLOTMY58
      IF(MOD(KODE/8,2).NE.0) WRITE(NWRIT,1080)                          PLOTMY59
C--ALL OPTIONS                                                          PLOTMY60
      FORY = .TRUE.                                                     PLOTMY61
      STUG = .FALSE.                                                    PLOTMY62
      TONLY = .FALSE.                                                   PLOTMY63
      IF(MOD(KODE/32,2).EQ.0) GO TO 190                                 PLOTMY64
C--OPTION 32                                                            PLOTMY65
      STUG = .TRUE.                                                     PLOTMY66
      KSY = P(9)                                                        PLOTMY67
      PWR10Y = 10.** (KSY-6)                                            PLOTMY68
      FY = P(10)*PWR10Y                                                 PLOTMY69
      F = FY                                                            PLOTMY70
C--SPECIAL CASE OF OPTION 32                                            PLOTMY71
      IF(P(5).GE.2.) GO TO 190                                          PLOTMY72
      TONLY = .TRUE.                                                    PLOTMY73
      DY =  P(11)*PWR10Y                                                PLOTMY74
      DX =  DY                                                          PLOTMY75
C--ALL OPTIONS                                                          PLOTMY76
  190 N = NPTST                                                         PLOTMY77
      CALL PISTUG(Y)                                                    PLOTMY78
      IF(DX.EQ.0.) GO TO 510                                            PLOTMY79
      FY = F                                                            PLOTMY80
      DY = DX                                                           PLOTMY81
C--MODIFY Y LABEL FORMAT IF NECESSARY                                   PLOTMY82
      IF(KFD.GT.0.AND.KFD.LE.6) YLAB(5) = FKFD(KFD)                     PLOTMY83
C--SET LOGICAL VARIABLES                                                PLOTMY84
      FORY = .FALSE.                                                    PLOTMY85
      STUG = .FALSE.                                                    PLOTMY86
      TONLY = .FALSE.                                                   PLOTMY87
C--OPTION 16                                                            PLOTMY88
      IF(MOD(KODE/16,2).EQ.0) GO TO 200                                 PLOTMY89
      STUG = .TRUE.                                                     PLOTMY90
      KSX = P(6)                                                        PLOTMY91
      PWR10X = 10.**(KSX-6)                                             PLOTMY92
      FX = P(7)*PWR10X                                                  PLOTMY93
      F = FX                                                            PLOTMY94
C--SPECIAL CASE OF OPTION 16                                            PLOTMY95
      IF(MOD(IFIX(P(5)),2).EQ.1) GO TO 200                              PLOTMY96
      TONLY = .TRUE.                                                    PLOTMY97
      DX = P(8)*PWR10X                                                  PLOTMY98
  200 IF(P(1).LT.2.)N = NPTS                                            PLOTMY99
C--ALL OPTIONS                                                          PLOTM100
      ILIM = N                                                          PLOTM101
```

```
      CALL PISTUG(X)
      IF(DX.EQ.0.) GO TO 510
      FX = F
C--MODIFY X LABEL FORMAT IF NECESSARY
      IF(KFD.GT.0.AND.KFD.LE.6) XLAB(4) = FKFD(KFD)
      IF(KFD.GT.0.AND.KFD.LE.6) XLAB(7) = FKFD(KFD)
C--COMPUTE AND TEST Y LABELS
  210 TDY = DY*10.
      DO 220 I=1,11
      TEMP = FY+FLOAT(I-1)*TDY
      ATEMP = ABS(TEMP)
      IF (ATEMP.LT.1.E-7) TEMP = 0.
      IF (ATEMP.GE.1.E+7) LABOUT = 2
  220 YLABEL(I) = TEMP
      KSYLAB = 1
C--WRITE Y LABELS
  230 WRITE(NWRIT,YLAB) YLABEL
      IF(KSYLAB.EQ.2) GO TO 510
C--INITIALIZE VARIABLES FOR MAIN PLOT
      KSYLAB = 2
      LCTR = 0
      NCTR = 1
      KOUT = 1
      KQUIT = 1
      XMIN = FX-DX/2.
      XMAX = FX+DX/2.
C--FILL THE A ARRAY FOR CURRENT LINE OF PLOT
C----FILL THE A ARRAY WITH BLANKS OR X-GRID MARKS, AS APPROPRIATE
  240 AFILL = BLANK
      IF(MOD(LCTR,NX).EQ.0) AFILL = XGRID
C--FILL IN LINE AND INSERT Y GRID MARKS
  260 DO 270 I=2,104
  270 A(I) = AFILL
      DO 280  I=2,104,NY
  280 A(I) = YGRID
      A(1) = BLANK
      IF(KOUT.EQ.2) GO TO 410
C--FIND INDEX OF NEXT X ON THE CURRENT LINE
      IGON = 1
  290 CONTINUE
      IF(IGON.GT.ILIM) GO TO 420
      DO 300 I=IGON,ILIM
      IF(X(I).LE.XMIN.OR.X(I).GT.XMAX) GO TO 300
      IMIN = I
      GO TO 310
  300 CONTINUE
C--IF NO MORE POINTS ON THE CURRENT LINE, WRITE IT OUT
      GO TO 420
C--PLACE PLOTTING CHARACTER IN PROPER POSITION (KYL) IN THE A ARRAY
  310 DO 390 IM=1,KTL
      LL = IMIN+(IM-1)*NPTS
      KY = (Y(LL)-FY)/DY+.5
      IF(P(1).LT.2.) GO TO 340
C--DUPY OR NODUP
      IK = 0
      KLAST = 2*KN+1
      DO 320 IL=3,KLAST,2
      IK ■ IK+K(IL)
      IF(IK.GE.IMIN) GO TO 330
  320 CONTINUE
      LABOUT = 5
      GO TO 510
  330 KPC ■ K(IL+1)
      GO TO 350
C--DUPX
  340 KPC = K(2*IM+2)
C--DEFINE KYL
  350 KYL = KY+2
C--PLOT OUT RANGE Y AS ■
      IF(KY.LT.0) GO TO 360
      IF(KY.GT.101) GO TO 370
      GO TO 390
  360 KYL = 1
      GO TO 380
  370 KYL ■ 104
  380 TPC = RMARK
  390 A(KYL) = TPC
C--ARE ALL POINTS DONE_
      IF(NCTR.GE.ILIM) GO TO 400
      NCTR = NCTR+1
      IGON = IMIN+1
      GO TO 290
C--ALL POINTS PLOTTED FOR ENTIRE PLOT
  400 KOUT = 2
  410 IF(MOD(LCTR,10).GT.5) NX = 10
      IF(MOD(LCTR,NX).EQ.0) KQUIT = 2
C
C--WRITE CURRENT LINE OF PLOT
  420 IF(MOD(LCTR,10).EQ.0) GO TO 430
C--WRITE LINE WITHOUT XLABEL
      WRITE(NWRIT,1020) A
      GO TO 440
  430 XLABEL = FX+FLOAT(LCTR)*DX
      TEMP ■ ABS(XLABEL)
      IF(TEMP.GE.1.E+7) LABOUT = 2
      IF(TEMP.LT.1.E-7) XLABEL ■ 0.
C--WRITE LINE WITH XLABEL
      WRITE(NWRIT,XLAB) XLABEL,A,XLABEL
C--INCREMENT LINE COUNTER
  440 LCTR = LCTR+1
      XMIN = XMAX
      XMAX = DX*(FLOAT(LCTR)+.5)+FX
C--DO NEXT LINE
      GO TO (240,230),KQUIT
C
C--PLOT COMPLETED - WRITE FINAL LINE OR ERROR MESSAGE
  510 GO TO (560,520,540,540,550),LABOUT
  520 WRITE(NWRIT,1040)
      GO TO 570
  540 WRITE(NWRIT,1050) (X(I),Y(I),I=1,2),(K(J),J=1,3),P(1)
      GO TO 570
  550 WRITE(NWRIT,1060)
  560 WRITE(NWRIT,1000)
C--RESTORE INITIAL FORMATS BEFORE RETURN
  570 CONTINUE
      YLAB(5) = FKFD0
      XLAB(4) = FKFD0
      XLAB(7) = FKFD0
      RETURN
C--FORMAT STATEMENTS
```

```
 1000 FORMAT(1H  )                                                          PLOTM222
 1020 FORMAT(1H  ,15X,104A1)                                                PLOTM223
 1040 FORMAT(1H  ,3X,10HBAD LABELS)                                         PLOTM224
 1050 FORMAT(1H  ,5H N.G.,4G20.8,3I7,F8.2)                                  PLOTM225
 1060 FORMAT(1H  ,16HERROR IN K ARRAY)                                      PLOTM226
 1080 FORMAT(1H  ,98X,28HOPTION 8 NO LONGER AVAILABLE)                      PLOTM227
      END                                                                   PLOTM228
       SUBROUTINE PISTUG(ARRAY)                                             PISTUG 2
C--PISTUG CALCULATES DX, KFD, AND FX OR FY                                   PISTUG 3
      LOGICAL FORY,STUG,TONLY                                               PISTUG 4
       DIMENSION ARRAY(1)                                                   PISTUG 5
      LEVEL 2,  ARRAY                                                       PISTUG 6
      COMMON /JOLO/ F,DX,TLINX,N,LABOUT,KFD,FORY,STUG,TONLY                 PISTUG 7
      KHAR(XMAX) = INT(ALOG10(XMAX)+40.)-40                                 PISTUG 8
C--FIND X1, THE FIRST POINT TO BE PLOTTED                                   PISTUG 9
   10 X1 = ARRAY(1)                                                         PISTUG10
      IF(STUG) GO TO 110                                                    PISTUG11
      DO 100 J=2,N                                                          PISTUG12
  100 X1 = AMIN1(X1,ARRAY(J))                                               PISTUG13
  110 IF(STUG) X1 = F                                                       PISTUG14
C--FIND XN, THE LAST POINT TO BE PLOTTED                                    PISTUG15
      DIFMAX = 0.                                                           PISTUG16
      DO 120 J=1,N                                                          PISTUG17
      DIF = ABS(X1-ARRAY(J))                                                PISTUG18
      IF(DIF.LE.DIFMAX) GO TO 120                                           PISTUG19
      DIFMAX = DIF                                                          PISTUG20
      IHOLD = J                                                             PISTUG21
  120 CONTINUE                                                              PISTUG22
      IF(DIFMAX.EQ.0.) GO TO 300                                            PISTUG23
      XN = ARRAY(IHOLD)                                                     PISTUG24
      IF(TONLY) GO TO 150                                                   PISTUG25
C--CALCULATE DX                                                             PISTUG26
      TLIN = 101.                                                           PISTUG27
      IF(.NOT.FORY) TLIN ■ TLINX                                            PISTUG28
      C5 ■ (XN-X1)/TLIN                                                     PISTUG29
      K7 = KHAR(ABS(C5))                                                    PISTUG30
      C9 = ABS(C5)/10.**K7                                                  PISTUG31
      D = 2.                                                                PISTUG32
      IF(C9.GT.2.) D ■ 2.5                                                  PISTUG33
      IF(C9.GT.2.5) D = 5.                                                  PISTUG34
      IF(C9.GT.5.) D ■ 10.                                                  PISTUG35
      DX ■ SIGN(D*10.**K7,C5)                                               PISTUG36
C--CALCULATE KFD                                                            PISTUG37
  150 K7 = KHAR(ABS(DX))                                                    PISTUG38
      IF(FLOAT(K7)+1.5.GT.5.6) LABOUT = 2                                   PISTUG39
      KFD = MAX0(0,MIN0(6,-K7))                                             PISTUG40
      IF(STUG) GO TO 200                                                    PISTUG41
C--CALCULATE F (FX OR FY)                                                   PISTUG42
      KC12 ■ INT(ABS(X1/DX))                                                PISTUG43
      KC15 = KC12-MOD(KC12,10)                                              PISTUG44
      IF(DX*X1.LT.0.) KC15 = KC15+10                                        PISTUG45
      IF(X1.LT.0.) KC15 = -KC15                                             PISTUG46
      F = ABS(DX)*FLOAT(KC15)                                               PISTUG47
      IF(.NOT.FORY) GO TO 200                                               PISTUG48
C--CALCULATE F DIFFERENTLY, IF NECESSARY, TO KEEP ALL POINTS ON PLOT        PISTUG49
      TEMP = F+100.*DX                                                      PISTUG50
      IF(TEMP.GE.SIGN(XN,DX)) GO TO 200                                     PISTUG51
      IF(DX*X1.LT.0.) KC12 = KC12+1                                         PISTUG52
      IF(X1.LT.0.) KC12 = -KC12                                             PISTUG53
      F = ABS(DX)*FLOAT(KC12)                                               PISTUG54
C--RETURN                                                                   PISTUG55
  200 RETURN                                                                PISTUG56
C--ERROR RETURN                                                             PISTUG57
  300 DX = 0.                                                               PISTUG58
      LABOUT = 3                                                            PISTUG59
      GO TO 200                                                             PISTUG60
      END                                                                   PISTUG61
      SUBROUTINE BLADE(IPLOTB)                                              BLADE  2
C ***             THE CENTRAL CONTROL ROUTINE                               BLADE  3
      REAL KIC, KICR, KIP, KIS, KM, KOC, KOCR, KOP, KOS, KTC, KTP, KTS      BLADE  4
      COMMON /SCALR/                                                        BLADE  5
     1 BLADES, CALP, CCC, CEPE, CKTC, CKTS, C1, C2, DRCE, DRCLEP, DRCLESBLADE  6
     2, DPCMST, DRCMT, DRCOI, DRCS, DRCT, DRCTEP, DRCTES, DRCTI, DRCTMP,BLADE  7
     3 DRCTMS, DRCTPI, DRCTSI, DSME, DSMT, DSOT, DSP, DSP1, DSP2, DSSE, BLADE  8
     4 DSS1, DSS2, DST, DSTI, EMT, IR, IW, KIC, KIP, KIS, KM, KOC, KOP, BLADE  9
     5 KOS, KTC, KTP, KTS, PI, RCI, RCMS, RCO, RCT, RCTP, RCTS, REE,    BLADE 10
     6 RELEP, PELES, REM, REMS, REMT, REOI, RES, RET, RETEP, RETES, RETIBLADE 11
     7, RETMP, RETMS, RETP, RETS, R1C, SGAM, SALP, SKTC, SKTS, TEPE,    BLADE 12
     8 THMAX, TKTN                                                      BLADE 13
      COMMON /INPUTB/                                                       BLADE 14
     1 ALP, KICR, KOCR, NB, P, R, SOLID, T, THLE, THTE, TMX, ZM         BLADE 15
      COMMON /INPUTT/RDUM1(17),IDUM1(16),XCHORD(2),XSTGR(2),XRI(2),     BLADE 16
     1 XRO(2),XBETI(2),XBETO(2),IDUM2(2),RDUM2(270),XMSP(50,2),         BLADE 17
     2 XTHSP(50,2)                                                      BLADE 18
      COMMON/BLANK/                                                         BLADE 19
     1 THSP1(13), THSP2(13), ZMSP1(13), ZMSP2(13), BETI1, BETI2, BETO1, BLADE 20
     2 BETO2, CHORD, RI, RO, STGR                                       BLADE 21
      COMMON/INPUTA/ XXMSP(50,2),XXTHSP(50,2)                               BLADE 22
      DIMENSION FSB(13)                                                     BLADE 23
      DATA FSB/.0,.05,.12,.2,.3,.4,.5,.6,.7,.8,.88,.95,1.0/                 BLADE 24
      IR = 5                                                                BLADE 25
      IW = 6                                                                BLADE 26
      PI = 3.1415927                                                        BLADE 27
      RADIAN = 57.29578                                                     BLADE 28
      IF(IPLOTB.NE.0)                                                       BLADE 29
     *WRITE (IW,2000)  NB, R, ALP, SOLID, KICR, KOCR, T, ZM, P, TMX,        BLADE 30
     X THLE, THTE                                                           BLADE 31
C *** INPUT OPTIONS ARE IP ■ 1 OR 3 FOR PUNCH AND IP.GT.1 FOR PLOT          BLADE 32
      THMAX ■ TMX/2.0                                                       BLADE 33
      KIC = KICR/RADIAN                                                     BLADE 34
      KOC = KOCR/RADIAN                                                     BLADE 35
      TALP = TAN(ALP/RADIAN)                                                BLADE 36
      SALP = TALP/SQRT(1.0 + TALP**2)                                       BLADE 37
      CALP = SQRT(1.0 - SALP**2)                                            BLADE 38
      BLADES ■ NB                                                           BLADE 39
      GBL = (KIC + KOC)/2.0                                                 BLADE 40
      SGAM = SIN(GBL)                                                       BLADE 41
      CGAM = SQRT(1.0 - SGAM**2)                                            BLADE 42
      CCC = 1.0 - THLE - THTE                                               BLADE 43
      C1 = T - THLE                                                         BLADE 44
      C2 = 1.0 - T - THTE                                                   BLADE 45
      THD = THLE - THTE                                                     BLADE 46
      TEPE = THD/CCC                                                        BLADE 47
      CEPE = 1.0/SQRT(1.0 + TEPE**2)                                        BLADE 48
      SEPE ■ TEPE*CEPE                                                      BLADE 49
      CALL CONIC(CHD)                                                       BLADE 50
C ***      BLADE ELEMENT SUCTION SURFACE Z AND THETA ARRAYS REFERENCED      BLADE 51
C *** TO THE LOWEST Z POINT OF THE LEADING EDGE CIRCLE.                     BLADE 52
      ZTRS = (DRCTSI + THLE)*CHD                                            BLADE 53
      RTC = R1C + (DRCTSI + THLE)*SALP                                      BLADE 54
```

```
      TTRS = RETS/RTC                                                   BLADE 55
      FST = DSS1/(DSS1 + DSS2)                                          BLADE 56
      DO 50 K=1,13                                                      BLADE 57
      FS = FSB(K) - FST                                                 BLADE 58
      IF (FS.GT.0.0)  GO TO 20                                          BLADE 59
      DSS = DSS1*FS/FST                                                 BLADE 60
      DK = (KTS - KIS)*DSS/DSS1                                         BLADE 61
      GO TO 30                                                          BLADE 62
   20 DSS = DSS2*FS/(1.0 - FST)                                         BLADE 63
      DK = (KOS - KTS)*DSS/DSS2                                         BLADE 64
   30 CALL EPSLON(KTS,DK,RCTS,DSS,DRCTS,RES)                            BLADE 65
      ZMSP1(K) = ZTRS + DRCTS*CHD                                       BLADE 66
   50 THSP1(K) = TTRS + RES/(RTC + DRCTS*SALP)                          BLADE 67
C ***      BLADE ELEMENT PRESSURE SURFACE Z AND THETA ARRAYS REFERENCED BLADE 68
C *** TO THE LOWEST Z POINT OF THE LEADING EDGE CIRCLE.                 BLADE 69
      ZTRP = (DRCTPI + THLE)*CHD                                        BLADE 70
      RTC = R1C + (DRCTPI + THLE)*SALP                                  BLADE 71
      TTRP = RETP/RTC                                                   BLADE 72
      FST = DSP1/DSP                                                    BLADE 73
      DO 100 K=1,13                                                     BLADE 74
      FS = FSB(K) - FST                                                 BLADE 75
      IF (FS.GT.0.0)  GO TO 70                                          BLADE 76
      DSS = DSP1*FS/FST                                                 BLADE 77
      DK = (KTP - KIP)*DSS/DSP1                                         BLADE 78
      GO TO 80                                                          BLADE 79
   70 DSS = DSP2*FS/(1.0 - FST)                                         BLADE 80
      DK = (KOP - KTP)*DSS/DSP2                                         BLADE 81
   80 CALL EPSLON(KTP,DK,RCTP,DSS,DRCTS,RES)                            BLADE 82
      ZMSP2(K) = ZTRP + DRCTS*CHD                                       BLADE 83
  100 THSP2(K) = TTRP + RES/(RTC + DRCTS*SALP)                          BLADE 84
      CHORD = (DRCOI + THLE + THTE)*CHD                                 BLADE 85
      RI = THLE*CHD                                                     BLADE 86
      RO = THTE*CHD                                                     BLADE 87
      BETI1 = KIS*RADIAN                                                BLADE 88
      BETO1 = KOS*RADIAN                                                BLADE 89
      BETI2 = KIP*RADIAN                                                BLADE 90
      BETO2 = KOP*RADIAN                                                BLADE 91
      GBL=RADIAN*ASIN(SGAM)                                             BLADE 92
      IF(IPLOTB.NE.0)                                                   BLADE 93
     *WRITE (IW,2100) CHD, GBL                                          BLADE 94
      RTE = R1C + DRCOI*SALP                                            BLADE 95
      RM = R1C + (DRCTI + DRCMT)*SALP                                   BLADE 96
      REOI = REOI/RTE                                                   BLADE 97
      STGR = REOI                                                       BLADE 98
      RETI = RETI/(R1C + DRCTI*SALP)                                    BLADE 99
      REMT = RETI + REMT/RM                                             BLADE100
      RELES = RELES/(R1C + DRCLES*SALP)                                 BLADE101
      RETES = REOI + RETES/(RTE + DRCTES*SALP)                          BLADE102
      RELEP = RELEP/(R1C + DRCLEP*SALP)                                 BLADE103
      RETEP = REOI + RETEP/(RTE + DRCTEP*SALP)                          BLADE104
      REMS = REMT + REMS/(R1C + (DRCTSI - DRCTMS)*SALP)                 BLADE105
      REM = REMT + REM/(R1C + (DRCTPI - DRCTMP)*SALP)                   BLADE106
      RETMS = REMS + RETMS/(R1C + DRCTSI*SALP)                          BLADE107
      RETMP = REM + RETMP/(R1C + DRCTPI*SALP)                           BLADE108
      DRCMT = (DRCTI + DRCMT)*CHD                                       BLADE109
      DRCTMS = (DRCTSI - DRCTMS)*CHD                                    BLADE110
      DRCTMP = (DRCTPI - DRCTMP)*CHD                                    BLADE111
      DRCTSI = DRCTSI*CHD                                               BLADE112
      DRCTPI = DRCTPI*CHD                                               BLADE113
      DRCTI = DRCTI*CHD                                                 BLADE114
      DRCTES = (DRCTES + DRCOI)*CHD                                     BLADE115
      DRCTEP = (DRCTEP + DRCOI)*CHD                                     BLADE116
      DRCOI = DRCOI*CHD                                                 BLADE117
      DRCLES = DRCLES*CHD                                               BLADE118
      DRCLEP = DRCLEP*CHD                                               BLADE119
      REF = 0.0                                                         BLADE120
      KTC = KTC*RADIAN                                                  BLADE121
      KTS = KTS*RADIAN                                                  BLADE122
      KTP = KTP*RADIAN                                                  BLADE123
      KM = KM*RADIAN                                                    BLADE124
      DSTI = DSTI*CHD                                                   BLADE125
      DSOT = DSOT*CHD                                                   BLADE126
      DSS1 = DSS1*CHD                                                   BLADE127
      DSS2 = DSS2*CHD                                                   BLADE128
      DSP1 = DSP1*CHD                                                   BLADE129
      DSP2 = DSP2*CHD                                                   BLADE130
      IF(IPLOTB.EQ.0) GO TO 110                                         BLADE131
      WRITE (IW,2120)  DRCLES, DRCTES, DRCTSI, DRCTMS, RELES, RETES,    BLADE132
     X RETMS, REMS, BETI1, BETO1, KTS, KM, DSS1, DSS2                   BLADE133
      WRITE (IW,2110)  REF, DRCOI, DRCTI, DRCMT, REF, REOI, RETI, REMT, BLADE134
     X KICR, KOCR, KTC, KM, DSTI, DSOT                                  BLADE135
      WRITE (IW,2130)  DRCLEP, DRCTEP, DRCTPI, DRCTMP, RELEP, RETEP,    BLADE136
     X RETMP, REM, BETI2, BETO2, KTP, KM, DSP1, DSP2                    BLADE137
  110 CONTINUE                                                          BLADE138
      RSALP = RADIAN*SALP                                               BLADE139
      RELES = RSALP*RELES                                               BLADE140
      RETES = RSALP*RETES                                               BLADE141
      RETMS = RSALP*RETMS                                               BLADE142
      REMS = RSALP*REMS                                                 BLADE143
      REOI = RSALP*REOI                                                 BLADE144
      RETI = RSALP*RETI                                                 BLADE145
      REMT = RSALP*REMT                                                 BLADE146
      RELEP = RSALP*RELEP                                               BLADE147
      RETEP = RSALP*RETEP                                               BLADE148
      RETMP = RSALP*RETMP                                               BLADE149
      REM = RSALP*REM                                                   BLADE150
      BETI1 = BETI1 + RELES                                             BLADE151
      BETO1 = BETO1 + RETES                                             BLADE152
      KTS = KTS + RETMS                                                 BLADE153
      BMS = KM + REMS                                                   BLADE154
C     KOCR = KOCR + REOI                                                BLADE155
      KTC = KTC + RETI                                                  BLADE156
      KM = KM + REMT                                                    BLADE157
      BETI2 = BETI2 + RELEP                                             BLADE158
      BETO2 = BETO2 + RETEP                                             BLADE159
      KTP = KTP + RETMP                                                 BLADE160
      BMP = KM + REM                                                    BLADE161
      IF(IPLOTB.EQ.0) GO TO 125                                         BLADE162
      WRITE (IW,2135)                                                   BLADE163
      WRITE (IW,2136)  RELES, RETES, RETMS, REMS, BETI1, BETO1, KTS, BMSBLADE164
      WRITE (IW,2137)  REF, REOI, RETI, REMT, KICR, KOCR, KTC, KM       BLADE165
      WRITE (IW,2138)  RELEP, RETEP, RETMP, REM, BETI2, BETO2, KTP, BMP BLADE166
      WRITE (IW,2140)  CHORD, STGR, RI, RO, BETI1, BETO1, RI, RO, BETI2,BLADE167
     X BETO2                                                            BLADE168
      DO 120 K=1,13                                                     BLADE169
  120 WRITE (IW,2150)  ZMSP1(K), THSP1(K), ZMSP2(K), THSP2(K)           BLADE170
      WRITE (IW,2160)                                                   BLADE171
  125 CONTINUE                                                          BLADE172
C *** ASSIGN VALUES TO BE PASSED TO TSONIC AND OPTIMIZATION PROGRAM     BLADE173
      XCHORD(1)=CHORD                                                   BLADE174
```

```
      XSTGR(1)=STGR                                                     BLADE175
      XRI(1)=RI                                                         BLADE176
      XRO(1)=RO                                                         BLADE177
      XBETI(1)=BETI1                                                    BLADE178
      XBETO(1)=BETO1                                                    BLADE179
      DO 130 K=1,13                                                     BLADE180
      XMSP(K,1)=ZMSP1(K)                                                BLADE181
      XXMSP(K,1)=ZMSP1(K)                                               BLADE182
      XXTHSP(K,1)=THSP1(K)                                              BLADE183
  130 XTHSP(K,1)=THSP1(K)                                               BLADE184
      XRI(2)=RI                                                         BLADE185
      XRO(2)=RO                                                         BLADE186
      XBETI(2)=BETI2                                                    BLADE187
      XBETO(2)=BETO2                                                    BLADE188
      DO 140 K=1,13                                                     BLADE189
      XMSP(K,2)=ZMSP2(K)                                                BLADE190
      XXMSP(K,2)=ZMSP2(K)                                               BLADE191
      XXTHSP(K,2)=THSP2(K)                                              BLADE192
  140 XTHSP(K,2)=THSP2(K)                                               BLADE193
      IF (IPLOTB.EQ.1) CALL EPLOT                                       BLADE194
      RETURN                                                            BLADE195
 2000 FORMAT (1H1 // 46X,39H*** INPUT FOR BLADE ELEMENT PROGRAM ***   //BLADE196
     1/ 49X,5HINLET,4X,6HOUTLET,4X,6HTRANS.,4X,7HMAX.TH.,3X,6HIN/OUT,5X,BLADE197
     2 4HMAX.,6X,4HL.E.,6X,4HT.E. / 10X,3HNO.,6X,5HINLET,5X,4HCONE,4X,  BLADE198
     3 8HSOLIDITY,4X,5HBLADE,5X,5HBLADE,5X,4HLOC.,6X,4HLOC.,5X,7HTURNINGBLADE199
     4,3X,6HTHICK.,5X,4HRAD.,6X,4HRAD. / 8X,6HBLADES,4X,6HRADIUS,5X,    BLAQE200
     5 5HANGLE,15X,5HANGLE,5X,5HANGLE,4X,6H/CHORD,4X,6H/CHORD,5X,4HRATE,BLADE201
     6 5X,6H/CHORD,4X,6H/CHORD,4X,6H/CHORD / 20X,3H(L),6X,5H(DEG),15X,  BLADE202
     7 5H(DEG),5X,5H(DEG) // 9X,I3,3X,3F10.5,F9.3,F10.4,6F10.5)         BLADE203
 2100 FORMAT (//////// 45X,41H*** OUTPUT FROM BLADE ELEMENT PROGRAM *** BLADE204
     1 /// 69X,7HELEMENT / 69X,7HSETTING / 58X,5HCHORD,7X,5HANGLE / 59X,BLADE205
     2 3H(L),8X,5H(DEG) // 52X,F12.5,F12.4 /// 10X,32H* MERID. LOC. FROMBLADE206
     3 L.E. CENT. **,2X,32H** THETA LOC. FROM L.E. CENT. **,2X,9(1H*),  BLADE207
     4 14H BLADE ANGLES ,9(1H*),3X,15HSEGMENT LENGTHS / 10X,8(4H****),  BLADE208
     5 2X,8(4H****),2X,33H(WITH RESPECT TO LOCAL CONIC RAY),2X,5(3H***) BLADE209
     6 // 8X,3(4X,5HINLET,2X,6HOUTLET,2X,6HTRANS.,2X,7HMAX.TH.),4X,     BLADE210
     7 5HFIRST,2X,6HSECOND / 8X,4(5X,3H(L)),3X,4(3X,5H(RAD)),2X,4(3X,   BLADE211
     8 5H(DEG)),1X,2(5X,3H(L)) / )                                      BLADE212
 2110 FORMAT (3X,5HCENT.,1X,4F8.5,2X,4F8.5,2X,4F8.3,3X,2F8.5 / )        BLADE213
 2120 FORMAT (3X,5HSUCT.,1X,4F8.5,2X,4F8.5,2X,4F8.3,3X,2F8.5 / )        BLADE214
 2130 FORMAT (3X,5HPRES.,1X,4F8.5,2X,4F8.5,2X,4F8.3,3X,2F8.5 / )        BLADE215
 2135 FORMAT ( // 44X,32H**** CONIC ANGLE COORD. - C ****,2X,9(1H*),    BLADE216
     1 14H BLADE ANGLES ,9(1H*) / 1H+,69X,1H- /  44X,32H** (FROM LEADINGBLADE217
     2 EDGE CENT.) ***,2X,32H(WITH RESPECT TO L.E. CENT. RAY)// 42X,    BLADE218
     3 2(4X,5HINLET,2X,6HOUTLET,2X,6HTRANS.,2X,7HMAX.TH.) / 43X,        BLADE219
     4 4(3X,5H(DEG)),2X,4(3X,5H(DEG)) / )                               BLADE220
 2136 FORMAT (37X,5HSUCT.,1X,4F8.3,2X,4F8.3 / )                         BLADE221
 2137 FORMAT (37X,5HCENT.,1X,4F8.3,2X,4F8.3 / )                         BLADE222
 2138 FORMAT (37X,5HPRES.,1X,4F8.3,2X,4F8.3 / )                         BLADE223
 2140 FORMAT (1H1 ///// 41X,51H*** OUTPUT THAT CAN BE PUNCHED FOR TSONICBLADE224
     1 INPUT *** /////51X,10(3H***)/51X,1H*,28X,1H*/ 51X,1H*,6X,5HCHORD,BLADE225
     2 6X,4HSTGR,7X,1H* / 51X,1H*,7X,3H(L),7X,5H(RAD),6X,1H* / 51X,1H*, BLADE226
     3 28X,1H* / 51X,1H*,2X,F10.5,F11.6,5X,1H* / 51X,1H*,28X,1H* / 15X, BLADE227
     4 103(1H*) / 15X,1H*,2(50X,1H*) / 15X,1H*14X,23H*** SUCTION SURFACEBLADE228
     5 ***,13X,1H*,13X,24H*** PRESSURE SURFACE ***,13X,1H* / 15X,1H*,2( BLADE229
     6 50X,1H*) / 15X,1H*,8X,34(1H*),8X,1H*,8X,34(1H*),8X,1H* / 15X,1H*,BLADE230
     7 2(50X,1H*) / 15X,2(1H*,8X,2HRI,9X,2HRO,8X,4HBETI,7X,4HBETO,6X),  BLADE23
     8 1H* / 15X,2(1H*,7X,3H(L),8X,3H(L),7X,5H(DEG),6X,5H(DEG),6X),1H* /BLADE23
     9 15X,1H*,2(50X,1H*) / 15X,1H*,2(1X,2F11.5,2F11.4,5X,1H*) / 15X,1H*BLADE23
     .,2(50X,1H*) / 15X,1H*,2(50X,1H*) / 15X,1H*,2(15X,4HZMSP,12X,4HTHSPBLADE23
     1,15X,1H*) / 15X,1H*,2(16X,3H(L),12X,5H(RAD),14X,1H*) / 15X,1H*,   BLADE235
     2 2(50X,1H*) )                                                     BLADE236
 2150 FORMAT (15X,1H*,2(9X,F11.5,6X,F11.6,13X,1H*) )                    BLADE237
 2160 FORMAT (15X,1H*,2(50X,1H*)/ 15X,103(1H*) )                        BLADE238
      END                                                               BLADE239
      SUBROUTINE CONIC(CHORD)                                           CONIC  2
C ***      THIS IS THE MAIN BLADE ELEMENT LAYOUT ROUTINE.  BLADE ELEMENTSCONIC  3
C *** ARE LAID OUT ON A CONE SUCH THAT THE CIRCULAR ARC CHARACTERISTIC  CONIC  4
C *** OF CONSTANT RATE OF ANGLE CHANGE WITH PATH DISTANCE IS MAINTAINED.CONIC  5
      REAL KIC, KIP, KIS, KM, KOC, KOP, KOS, KTC, KTP, KTS              CONIC  6
      COMMON /SCALR/                                                    CONIC  7
     1 BLADES, CALP, CCC, CEPE, CKTC, CKTS, C1, C2, DRCE, DRCLEP, DRCLESCONIC  8
     2, DRCMST, DRCMT, DRCOI, DRCS, DRCT, DRCTEP, DRCTES, DRCTI, DRCTMP,CONIC  9
     3 DRCTMS, DRCTPI, DRCTSI, DSME, DSMT, DSOT, DSP, DSP1, DSP2, DSSE, CONIC 10
     4 DSS1, DSS2, DST, DSTI, EMT, IR, IW, KIC, KIP, KIS, KM, KOC, KOP, CONIC 11
     5 KOS, KTC, KTP, KTS, PI, RCI, RCMS, RCO, RCT, RCTP, RCTS, REE,    CONIC 12
     6 RELEP, RELES, REM, REMS, REMT, REOI, RES, RET, RETEP, RETES, RETICONIC 13
     7, RETMP, RETMS, RETP, RETS, R1C, SGAM, SALP, SKTC, SKTS, TEPE,    CONIC 14
     8 THMAX, TKTN                                                      CONIC 15
      COMMON /INPUTB/                                                   CONIC 16
     1 ALP, KICR, KOCR, NB, P, R, SOLID, T, THLE, THTE, TMX, ZM         CONIC 17
C ***       ESTABLISH BLADE ELEMENT CENTERLINE TO SATISFY CAMBER, CHORD CONIC 18
C *** AND TRANSITION POINT REQUIREMENTS.                                CONIC 19
      PI2 = PI/2.0                                                      CONIC 20
      CGAM= SQRT(1.0 - SGAM**2)                                         CONIC 21
      ICONV = 0                                                         CONIC 22
      DKAPPA = KIC - KOC                                                CONIC 23
      ICL = 1                                                           CONIC 24
      DK2 = DKAPPA/(1.0 + P*C1/C2)                                      CONIC 25
   10 CHORD = 2.0*PI*R*SOLID/(BLADES - PI*SOLID*CGAM*SALP)              CONIC 26
      R1C = R/CHORD                                                     CONIC 27
      CCHORD = CALP*CHORD                                               CONIC 28
      IF (ICL.GT.1)  GO TO 15                                           CONIC 29
      EPS= CCC*SGAM*SALP/(R1C + (THLE + CCC*CGAM)*SALP)                 CONIC 30
      DPHI = DKAPPA - EPS                                               CONIC 31
      DPHI4 = DPHI/4.0                                                  CONIC 32
      DPHIHS = DPHI*DPHI4                                               CONIC 33
      DSOI = CCC/(1.0 - DPHIHS/6.0*(1.0 - DPHIHS/20.0))                 CONIC 34
      DSTI= C1/CCC*DSOI                                                 CONIC 35
      DSOT= DSOI - DSTI                                                 CONIC 36
   15 IF (ABS(SALP/R1C).LT.1.0E-08)  GO TO 20                           CONIC 37
      RCI= R1C/SALP + THLE                                              CONIC 38
      GO TO 30                                                          CONIC 39
   20 RCI= 1.0E+08                                                      CONIC 40
   30 DK1 = DKAPPA - DK2                                                CONIC 41
      CALL EPSLON(KIC,-DK1,RCI,DSTI, DRCTI, RETI)                       CONIC 42
      KTC = KIC - DK1                                                   CONIC 43
      RCT= RCI + DRCTI                                                  CONIC 44
   35 CALL EPSLON(KTC,-DK2,RCT,DSOT,DRCOT,REOT)                         CONIC 45
      RCO= RCT + DRCOT                                                  CONIC 46
      REOI= RCO/RCT*RETI + REOT                                         CONIC 47
      DRCOI= DRCTI + DRCOT                                              CONIC 48
      CALL TANKAP(RCI,DRCOI,REOI,TANCCO)                                CONIC 49
      TGBL= (TANCCO + TEPE)/(1.0 - TANCCO*TEPE)                         CONIC 50
      CALL RPOINT(RCI,DRCTI,RETI,TGBL,DRCTP)                            CONIC 51
      SECGBL= SQRT(1.0 + TGBL**2)                                       CONIC 52
      DC1= DRCTP*SECGBL - C1                                            CONIC 53
      DC2 = DRCOI*SQRT(1.0 + TANCCO**2)*CEPE - CCC                      CONIC 54
      IF (ICL.GT.1.AND.ABS(TGBL - TGBLL).LT.1.0E-04)  GO TO 37          CONIC 55
      ICL = 2                                                           CONIC 56
```

```
      TGBLL = TGBL
      CGAM = 1.0/SQRT(1.0 + TGBL**2)
      GO TO 38
   37 ICONV = 1
   38 IF (ABS(DC1).LT.1.0E-04)  GO TO 40
      DS1= DSTI*DC1/(C1 + DC1)
      DSTI= DSTI - DS1
      DSOT= DSOT - DSOI*DC2/(CCC + DC2) + DS1
      DSOI = DSTI + DSOT
      DK2 = DKAPPA/(1.0 + P*DSTI/DSOT)
      IF (ICONV.LT.1)  GO TO 10
      GO TO 30
   40 IF (ABS(DC2).LT.1.0E-05)  GO TO 50
      DSOT= DSOT - DSOI*DC2/(CCC + DC2)
      DSOI = DSTI + DSOT
      IF (ICONV.LT.1)  GO TO 10
      GO TO 35
C ***     CONIC COORDINATES OF THE MAXIMUM THICKNESS POINT
   50 SGAM=TGBL*CGAM
      ZMT = ZM - T
      IF (ABS(ZMT).GT.1.0E-07)  GO TO 120
      DRCMT= 0.0
      REMT= 0.0
      DK= 0.0
      DSMT = 0.0
      DSME= DSTI
      DRCTMS = 0.0
      RETMS = 0.0
      DRCTMP = 0.0
      RETMP = 0.0
      GO TO 150
  120 HKTC= KTC/2.0
      SHKTC= HKTC*SRS(HKTC)
      SHKTCQ= SHKTC**2
      SKTC= 2.0*SHKTC*SQRT(1.0 - SHKTCQ)
      IF (ABS(SKTC).LT.1.0E-07) SKTC = 1.0E-07
      CKTC= 1.0 - 2.0*SHKTCQ
      TKTN= -CKTC/SKTC
      IF (ZMT.GT.0.0)  GO TO 130
      DSMT= DSTI*ZMT/C1
      DKDS= DK1/DSTI
      DSME= DSTI
      GO TO 140
  130 DSMT= DSOT*ZMT/C2
      DKDS = DK2/DSOT
      DSME= -DSOT
  140 DK= -DSMT*DKDS
      CALL EPSLON(KTC,DK,RCT,DSMT,DRCMT,REMT)
      CALL RPOINT(RCT,DRCMT,REMT,TGBL,DRCMP)
      ZMTCAL= DRCMP*SECGBL
      IF (ABS(ZMTCAL - ZMT).LT.1.5E-05)  GO TO 150
      DSMT = DSMT*ZMT/ZMTCAL
      GO TO 140
  150 RCM= RCT + DRCMT
      KM= KTC  + DK
      HKM= KM/2.0
      SHKM= HKM*SRS(HKM)
      SHKMQ= SHKM**2
      CHKM= SQRT(1.0 - SHKMQ)
      SKM= 2.0*SHKM*CHKM
      CKM= 1.0 - 2.0*SHKMQ
      DSME= DSME + DSMT
C ***     DEFINITION OF SUCTION SURFACE MAX. THICKNESS POINT
      CALL EPSLON(KM+PI2,0.0,RCM,THMAX,DRCM,REM)
      REMS = REM
      RCMS= RCM + DRCM
      IF (ZMT.GT.1.0E-07)  GO TO 180
      REMI = (1.0 + DRCMT/RCT)*RETI + REMT
C ***     DEFINITION OF SUCTION SURFACE CURVE FOR MAXIMUM THICKNESS
C *** POINT ON OR AHEAD OF THE TRANSITION POINT
      DK= 2.0*(THMAX - THLE)/DSME
      KIS = KIC + DK
      KIP = KIC - DK
      DRCIM= -DRCTI - DRCMT - DRCM
      EMSI= REMI/RCM + REM/RCMS
      CALL SURF(KIS,KM,SKM,CKM,RCI,DRCIM,THLE,EMSI,DSSE)
      DRCLES = DRCE
      RELES = REE
      IF (ABS(ZMT).LT.1.0E-07)  GO TO 160
      DRCMST= DRCMT + DRCM
      EMT= REM/RCMS + REMT/RCM
      CALL TRAN(KIS,THLE,THMAX,KTS,RCTS,RETS,DSS1)
      DRCTMS = DRCS
      RETMS = RES
      DHKT= (KTC - KTS)/2.0
      DK= 2.0*(DST - THTE - DSOT*DHKT)/(DSOT + (DST - THTE)*DHKT)
      DRCOTS= DRCOT - DRCT
      EMSO= RET/RCTS - REOT/RCO
      DRCTSI = DRCTI + DRCT
      GO TO 170
  160 RCTS= RCMS
      KTS= KM
      RETS = RCMS*EMSI
      DSS1 = -DSSE
      SKTS= SKM
      CKTS= CKM
      DK= 2.0*(THMAX - THTE)/DSOT
      DRCOTS= DRCOT - DRCM
      EMSO= REM/RCMS - REOT/RCO
      DRCTSI = DRCTI + DRCM
  170 KOS = KOC - DK
      KOP = KOC + DK
      CALL SURF(KOS,KTS,SKTS,CKTS,RCO,DRCOTS,THTE,EMSO,DSS2)
      DRCTES = DRCE
      RETES = REE
      GO TO 190
C ***     DEFINITION OF SUCTION SURFACE CURVE FOR MAXIMUM THICKNESS
C *** POINT BEHIND THE TRANSITION POINT
  180 DK= 2.0*(THMAX - THTE)/DSME
      KOS = KOC + DK
      KOP = KOC - DK
      DRCOM= DRCOT - DRCMT - DRCM
      EMSO= REMT/RCM - REOT/RCO + REM/RCMS
      CALL SURF(KOS,KM,SKM,CKM,RCO,DRCOM,THTE,EMSO,DSSE)
      DRCTES = DRCE
      RETES = REE
      DRCMST= DRCMT + DRCM
      EMT= REM/RCMS + REMT/RCM
      CALL TRAN(KOS,THTE,THMAX,KTS,RCTS,RETS,DSS2)
      DRCTMS = DRCS
```

```
      RETMS = RES
      DHKT= (KTS- KTC)/2.0
      DK= 2.0*(DST - THLE - DSTI*DHKT)/(DSTI + (DST - THLE)*DHKT)
      KIS = KIC + DK
      KIP = KIC - DK
      DPCTSI = DRCTI + DRCT
      EMSI= RETI/RCT + RET/RCTS
      CALL SURF(KIS,KTS,SKTS,CKTS,RCI,-DRCTSI,THLE,EMSI,DSSE)
      DRCLES = DRCE
      RELES = REE
      DSS1 = -DSSE
C ***      DEFINITION OF PRESSURE SURFACE MAXIMUM THICKNESS POINT
  190 CALL EPSLON(KM+PI2,0.0,RCM,-THMAX,DRCM,REM)
      RCMS= RCM + DRCM
      IF (ZMT.GT.1.0E-07)  GO TO 220
C ***      DEFINITION OF PRESSURE SURFACE CURVE FOR MAXIMUM THICKNESS
C *** POINT ON OR AHEAD OF THE TRANSTION POINT
      DRCIM= -DRCTI - DRCMT - DRCM
      EMSI= REMI/RCM + REM/RCMS
      CALL SURF(KIP,KM,SKM,CKM,RCI,DRCIM,-THLE,EMSI,DSSE)
      DRCLEP = DRCE
      RELEP = REE
      IF (ABS(ZMT).LT.1.0E-07)  GO TO 200
      DRCMST= DRCMT + DRCM
      EMT = REM/RCMS + REMT/RCM
      CALL TRAN(KIP,-THLE,-THMAX,KTP,RCTP,RETP,DSP1)
      DRCTMP = DRCS
      RETMP = RES
      DRCOTS= DRCOT - DRCT
      EMSO = RET/RCTP - REOT/RCO
      DRCTPI = DRCTI + DRCT
      GO TO 210
  200 RCTP = RCMS
      KTP= KM
      RETP= RCMS * EMSI
      DSP1 = -DSSE
      DRCOTS= DRCOT - DRCM
      EMSO = REM/RCMS - REOT/RCO
      DRCTPI = DRCTI + DRCM
  210 CALL SURF(KOP,KTP,SKTS,CKTS,RCO,DRCOTS,-THTE,EMSO,DSP2)
      DRCTEP = DRCE
      RETEP = REE
      GO TO 230
C ***      DEFINITION OF PRESSURE SURFACE CURVE FOR THE MAXIMUM
C *** THICKNESS POINT BEHIND THE TRANSITION POINT
  220 DRCOM= DRCOT - DRCMT - DRCM
      EMSO = REMT/RCM - REOT/RCO + REM/RCMS
      CALL SURF(KOP,KM,SKM,CKM,RCO,DRCOM,-THTE,EMSO,DSSE)
      DRCTEP = DRCE
      RETEP = REE
      DRCMST= DRCMT + DRCM
      EMT = REM/RCMS + REMT/RCM
      CALL TRAN(KOP,-THTE,-THMAX,KTP,RCTP,RETP,DSP2)
      DRCTMP = DRCS
      RETMP = RES
      DRCTPI = DRCTI + DRCT
      EMSI = RETI/RCT + RET/RCTP
      CALL SURF(KIP,KTP,SKTS,CKTS,RCI,-DRCTPI,-THLE,EMSI,DSSE)
      DRCLEP = DRCE
      RELEP = REE
      DSP1 = -DSSE
  230 DSS = DSS1 + DSS2
      DSP = DSP1 + DSP2
      DKE = KIS - KIC
      IF (DKE.GT.0.0)  GO TO 240
      WRITE (IW,2000)
  240 IF ((KOS - KOC).LE.0.0)  GO TO 250
      WRITE (IW,2010)
  250 RETURN
 2000 FORMAT (//// 30X,73H** NOTE THAT THE ELEMENT THICKNESS IS DECREASI
     1NG FROM THE LEADING EDGE ** )
 2010 FORMAT (//// 23X,86H** NOTE THAT THE ELEMENT THICKNESS IS INCREASI
     2NG AS THE TRAILING EDGE IS APPROACHED **)
      END
      SUBROUTINE EPLOT
      DIMENSION X(26),Y(26),STRING(132)
      COMMON/BLANK/
     1 THSP1(13), THSP2(13), ZMSP1(13), ZMSP2(13), BETI1, BETI2, BETO1,
     2 BETO2, CHORD, RI, RO, STGR
      EQUIVALENCE (X(1),ZMSP1(1)),(Y(1),THSP1(1))
      DATA STRING/132*1H /, BLANK/1H /, PLUS/1H+/
      WRITE(6,900)
C *** FIND X RANGE
      XR=0.0
      DO 10 I=1,26
   10 IF (X(I) .GT. XR) XR=X(I)
C *** ARRANGE POINTS IN ORDER OF DECREASING Y
   20 IEX=0
      DO 30 I=1,25
      IF (Y(I) .GE. Y(I+1)) GO TO 30
      TEMP=X(I)
      X(I)=X(I+1)
      X(I+1)=TEMP
      TEMP=Y(I)
      Y(I)=Y(I+1)
      Y(I+1)=TEMP
      IEX=1
   30 CONTINUE
      IF (IEX .EQ. 1) GO TO 20
C *** FIND Y RANGE, X AND Y INCREMENTS
      YR=Y(1)-Y(26)
C *** NOTE: 0.1IN/CHARACTER, 0.167IN/LINE, SO 79.2 = 0.6*132.0
      NY=0.5+79.2*YR/XR
      DY=YR/(FLOAT(NY)-1.0)
      DX=XR/131.0
C *** PLOT POINTS
      IYO=0
      DO 50 I=1,26
      IY=1.5+(Y(1)-Y(I))/DY
      NFEED=IY-IYO
      IYO=IY
      IF (NFEED .EQ. 0) GO TO 45
      DO 40 IFEED=1,NFEED
   40 WRITE(6,1000)
   45 IX=1.5+X(I)/DX
      STRING(IX)=PLUS
      WRITE(6,1010) STRING
      STRING(IX)=BLANK
   50 CONTINUE
      RETURN
```

```
  900 FORMAT (1H1//40X,                                                 EPLOT 48
     154H*** PLOT OF BLADE SURFACE IN THETA - M COORDINATES ***/////)   EPLOT 49
 1000 FORMAT (1H )                                                      EPLOT 50
 1010 FORMAT (1H+,132A1)                                                EPLOT 51
      END                                                               EPLOT 52
      SUBROUTINE EPSLON(KO,DK,RO,DS,DR,RE)                              EPSLON 2
C ***      CALCULATION OF CONIC RADIAL AND CIRCUMFERENTIAL COMPONENTS OFEPSLON 3
C *** A BLADE ELEMENT SEGMENT WITH GIVEN PATH DISTANCE AND END ANGLES   EPSLON 4
      REAL KO                                                           EPSLON 5
    • IF (ABS(DS).LT.1.0E-08)  GO TO 70                                 EPSLON 6
      HDK= DK/2.0                                                       EPSLON 7
      IF (ABS(HDK).GT.0.78539816)  GO TO 2                              EPSLON 8
      SR= SRS(HDK)                                                      EPSLON 9
      SHDK= HDK*SR                                                      EPSLON10
      SHDKQ= SHDK**2                                                    EPSLON11
      CHDK= SQRT(1.0 - SHDKQ)                                           EPSLON12
      GO TO 6                                                           EPSLON13
    2 IF (HDK.LT.0.0)  GO TO 3                                          EPSLON14
      HDK = 1.5707963 - HDK                                             EPSLON15
      CHDK = HDK*SRS(HDK)                                               EPSLON16
      SHDKQ = 1.0 - CHDK**2                                             EPSLON17
      SHDK = SQRT(SHDKQ)                                                EPSLON18
      GO TO 4                                                           EPSLON19
    3 HDK = 1.5707963 + HDK                                             EPSLON20
      CHDK = HDK*SRS(HDK)                                               EPSLON21
      SHDKQ = 1.0 - CHDK**2                                             EPSLON22
      SHDK = -SQRT(SHDKQ)                                               EPSLON23
    4 SR = ABS(SHDK)/HDK                                                EPSLON24
    6 HKO = KO/2.0                                                      EPSLON25
      IF (HKO.GT.0.78539816)  GO TO 7                                   EPSLON26
      SHKO= HKO*SRS(HKO)                                                EPSLON27
      SHKOQ= SHKO**2                                                    EPSLON28
      CHKO= SQRT(1.0 - SHKOQ)                                           EPSLON29
      SKO= 2.0*SHKO*CHKO                                                EPSLON30
      CKO= 1.0 - 2.0*SHKOQ                                              EPSLON31
      GO TO 8                                                           EPSLON32
    7 HKO = 0.78539816 - HKO                                            EPSLON33
      SHKO = HKO*SRS(HKO)                                               EPSLON34
      SHKOQ = SHKO**2                                                   EPSLON35
      CHKO = SQRT(1.0 - SHKOQ)                                          EPSLON36
      SKO = 1.0 - 2.0*SHKOQ                                             EPSLON37
      CKO = 2.0*SHKO*CHKO                                               EPSLON38
    8 SKA = SHDK*CKO + SKO*CHDK                                         EPSLON39
      CKA= CHDK*CKO - SKO*SHDK                                          EPSLON40
C ***      CONIC RADIAL COMPONENT OF THE PATH                           EPSLON41
      DR= DS*CKA*SR                                                     EPSLON42
      IF (ABS(DK).GT.0.00001) GO TO 10                                  EPSLON43
      DRR = DR/RO                                                       EPSLON44
      IF (ABS(DRR).LT.0.01)  GO TO 9                                    EPSLON45
C ***      CIRCUMFERENTIAL COMP. WHEN PATH ANGLE IS ESSENTIALLY CONSTANTEPSLON46
      RE = (RO + DR)*SKA/CKA*ALOG(1.0 + DRR)                            EPSLON47
      RETURN                                                            EPSLON48
    9 RE = (1.0 • DRR)*DS*SR*SKA*(1.0 - DRR*(0.5 - DRR*(0.33333333 - DRREPSLON49
     X /4.0)))                                                          EPSLON50
      RETURN                                                            EPSLON51
   10 RS= RO/DS                                                         EPSLON52
      IF (ABS(RS).GT.10000.0) GO TO 60                                  EPSLON53
      IF (RS**2/ABS(DK).GT.1.7E+09) GO TO 60                            EPSLON54
C ***      CONIC CIRCUMFERENTIAL COMPONENT OF PATH BY GENERAL EQUATION  EPSLON55
      RCK= RS*DK - SKO                                                  EPSLON56
      QDKS= HDK**2/4.0                                                  EPSLON57
      SES = 0.66666667*RCK*QDKS*(1.0 - 0.6*QDKS*(1.0 - 0.15873016*QDKS* EPSLON58
     X (1.0 - 0.077777778*QDKS*(1.0 - 0.046753247*QDKS*(1.0 - .031339031EPSLON59
     X *QDKS)))))                                                       EPSLON60
      DRR= DR/RO                                                        EPSLON61
      IF (ABS(DRR).GT.0.21)  GO TO 20                                   EPSLON62
      RRM = 0.5*DRR*(1.0 - 0.25*DRR*(1.0 - 0.5*DRR*(1.0 - 0.625*DRR*(1.0EPSLON63
     X - 0.7*DRR*(1.0 - 0.75*DRR*(1.0 - 0.78571429*DRR*(1.0 - 0.8125*DRREPSLON64
     X *(1.0 - 0.83333333*DRR))))))))                                   EPSLON65
      RRQ = RRM  + 1.0                                                  EPSLON66
      GO TO 30                                                          EPSLON67
   20 RRQ= SQRT(1.0 + DRR)                                              EPSLON68
      RRM= RRQ - 1.0                                                    EPSLON69
   30 RM= RRQ*RS                                                        EPSLON70
      D = RCK*CHDK + SKA + DK*RM                                        EPSLON71
      XS= SHDKQ*(1.0 -RCK**2)/D**2                                      EPSLON72
      XSN= 35.0*ABS(XS)                                                 EPSLON73
      NXS= XSN                                                          EPSLON74
      N= 5 + NXS                                                        EPSLON75
      SXS= 0.0                                                          EPSLON76
      XPS= 1.0                                                          EPSLON77
      DKN= 1.0                                                          EPSLON78
      DO 40  KN=1,N                                                     EPSLON79
      IF (ABS(XPS).LT.1.0E-12.AND.KN.NE.1) GO TO 50                     EPSLON80
      XPS= XPS*XS                                                       EPSLON81
      DKN= DKN + 2.0                                                    EPSLON82
   40 SXS= SXS + XPS/DKN                                                EPSLON83
   50 RE= (RO + DR)*(DK*(SKO + SKA + DK*RS*RRM - SES) - 4.0*RCK*SHDK*   EPSLON84
     X SXS)/D                                                           EPSLON85
      RETURN                                                            EPSLON86
C ***      CONIC CIRCUMFERENTIAL COMPONENT WHEN PATH DISTANCE IS A VERY EPSLON87
C *** SMALL FRACTION OF THE DISTANCE TO THE CONE VERTEX.                EPSLON88
   60 DRR= DR/RO                                                        EPSLON89
      RE = (1.0 + DRR)/(1.0 • 0.5*DRR*(1.0 - 0.25*DRR*(1.0 - 0.5*DRR*   EPSLON90
     X (1.0 - 0.625*DRR))))*DS*SKA*SR                                   EPSLON91
      RETURN                                                            EPSLON92
   70 DR = 0.0                                                          EPSLON93
      RE = 0.0                                                          EPSLON94
      RETURN                                                            EPSLON95
      END                                                               EPSLON96
      SUBROUTINE RPOINT(RO,DR,RE,TK,DRP)                                RPOINT 2
C ***      THIS SUBROUTINE CALCULATES THE CONIC RADIAL COORDINATE AT THERPOINT 3
C *** INTERSECTION OF PERPENDICULAR CONSTANT ANGLE LINES FROM TWO KNOWN RPOINT 4
C *** POINTS ON A CONE.  THE LINE THROUGH THE REFERENCE POINT HAS THE   RPOINT 5
C *** INPUT SLOPE TK.                                                   RPOINT 6
      R = DR/RO                                                         RPOINT 7
      CK = SQRT(1.0/(1.0 + TK**2))                                      RPOINT 8
      SK = TK*CK                                                        RPOINT 9
      IF (ABS(R).LT.0.01) GO TO 20                                      RPOINT10
      DRP = RO*(EXP((RE*SK/(RO + DR) + ALOG(1.0 + R)*CK)*CK) - 1.0)     RPOINT11
      RETURN                                                            RPOINT12
   20 C = (RE*SK/(RO + DR) + R*(1.0 - 0.5*R*(1.0 - 0.66666667*R*(1.0 -  RPOINT13
     X 0.75*R)))*CK)*CK                                                 RPOINT14
      DRP = C                                                           RPOINT15
   30 CS = DRP*(1.0 - 0.5*DRP*(1.0 - 0.66666667*DRP*(1.0 - 0.75*DRP)))  RPOINT16
      IF (ABS((CS - C)/C).LT.1.0E-06) GO TO 40                          RPOINT17
      DRP = DRP*C/CS                                                    RPOINT18
      GO TO 30                                                          RPOINT19
   40 DRP = DRP*RO                                                      RPOINT20
      RETURN                                                            RPOINT21
```

```
      END
      FUNCTION SRS(ANG)
C ***       SERIES FOR (SIN(ANG))/ANG WHEN THE MAGNITUDE OF ANG IS LESS
C *** THAN PI/4
      IF (ABS(ANG).LT.1.0E-05) GO TO 10
      AQ = ANG**2
      SRS = 1.0 - AQ/6.0*(1.0 - AQ/20.0*(1.0 - AQ/42.0*(1.0 - AQ/72.0)))
      RETURN
   10 SRS = 1.0
      RETURN
      END
      SUBROUTINE SURF(KE,KMM,SKM,CKM,RO,DRC,TE,EMS,DSS)
C ***       THIS SUBROUTINE CALCULATES THE BLADE ELEMENT SURFACE CURVE
C *** END POINT COORDINATES.  THE SURFACE CURVE IS NORMAL TO THE END
C *** POINT THICKNESS PATH AND TANGENT TO A SURFACE REFERENCE POINT
C *** WHICH IS EITHER THE TRANSITION OR MAXIMUM THICKNESS POINT.
      REAL KE, KE1, KIC, KIP, KIS, KM, KMM, KOC, KOP, KOS, KTC, KTP, KTS
      COMMON /SCALR/
     1 BLADES, CALP, CCC, CEPE, CKTC, CKTS, C1, C2, DRCE, DRCLEP, DRCLES
     2, DRCMST, DRCMT, DRCOI, DRCS, DRCT, DRCTEP, DRCTES, DRCTI, DRCTMP,
     3 DRCTMS, DRCTPI, DRCTSI, DSME, DSMT, DSOT, DSP, DSP1, DSP2, DSSE,
     4 DSS1, DSS2, DST, DSTI, EMT, IR, IW, KIC, KIP, KIS, KM, KOC, KOP,
     5 KOS, KTC, KTP, KTS, PI, RCI, RCMS, RCO, RCT, RCTP, RCTS, REE,
     6 RELEP, RELES, REM, REMS, REMT, REOI, RES, RET, RETEP, RETES, RETI
     7, RETMP, RETMS, RETP, RETS, R1C, SGAM, SALP, SKTC, SKTS, TEPE,
     8 THMAX, TKTN
      RMS = RO - DRC
      IT = 1
   10 CALL EPSLON(KE + 1.5707963,0.0,RO,TE,DRCE,REE)
      DRCS = DRC + DRCE
      DK = KE - KMM
      HDK = DK/2.0
      SR = SRS(HDK)
      SHDK = HDK*SR
      CHDK = SQRT(1.0 - SHDK**2)
      DSS = DRCS/(SR*(CHDK*CKM - SHDK*SKM))
      CALL EPSLON(KMM,DK,RMS,DSS,DRCS,RES)
      DRE = (RO + DRCE)*EMS + RES - REE
      IF (ABS(DRE).LT.1.0E-05) RETURN
      IF (IT.EQ.2) GO TO 20
      KE1 = KE
      DRE1 = DRE
      KE = KE - 2.0*DRE*(CKM*(1.0 -2.0*SHDK**2) -2.0*SKM*SHDK*CHDK)/DSS
      IT = 2
      GO TO 10
   20 KE = KE + (KE1 - KE)*DRE/(DRE - DRE1)
      GO TO 10
      END
      SUBROUTINE TANKAP(RO,DR,RE,TK)
C ***        CALCULATION OF THE SLOPE OF THE CONSTANT ANGLE PATH BETWEEN
C *** TWO POINTS IN CONIC RADIUS AND EPSILON COORDINATES
      R = DR/RO
      IF (ABS(R).LT.0.1) GO TO 20
      TK = RE/((RO + DR)*ALOG(1.0 + R))
      RETURN
   20 SUM = 1.0
      IF (ABS(R).GT.1.0E-08) GO TO 25
      IF (ABS(DR/RE).GT.1.0E-08) GO TO 35
      TK = 1.0E+08
      RETURN
   25 PROD = 1.0
      DN = 8.0/(-ALOG10(ABS(R)))
      NT = DN
      DO 30 I=1,NT
      N = I + 1
      DN = N
      PROD = -PROD*R
   30 SUM = SUM + PROD/DN
   35 TK = RE/((RO + DR)*R*SUM)
      RETURN
      END
      SUBROUTINE TRAN(KE,TE,TM,KT,RT,RE,DS)
C ***        THIS SUBROUTINE CALCULATES THE BLADE ELEMENT SURFACE CURVE
C *** TRANSITION POINT COORDINATES FROM THE INTERSECTION OF THE
C *** ESTABLISHED SURFACE CURVE OVER THE MAXIMUM THICKNESS POINT WITH A
C *** PATH PERPENDICULAR TO THE CENTERLINE AT THE TRANSITION POINT.
      REAL KE,KIC,KIP,KIS,KM,KOC,KOP,KOS,KT,KTC,KTP,KTS
      COMMON /SCALR/
     1 BLADES, CALP, CCC, CEPE, CKTC, CKTS, C1, C2, DRCE, DRCLEP, DRCLES
     2, DRCMST, DRCMT, DRCOI, DRCS, DRCT, DRCTEP, DRCTES, DRCTI, DRCTMP,
     3 DRCTMS, DRCTPI, DRCTSI, DSME, DSMT, DSOT, DSP, DSP1, DSP2, DSSE,
     4 DSS1, DSS2, DST, DSTI, EMT, IR, IW, KIC, KIP, KIS, KM, KOC, KOP,
     5 KOS, KTC, KTP, KTS, PI, RCI, RCMS, RCO, RCT, RCTP, RCTS, REE,
     6 RELEP, RELES, REM, REMS, REMT, REOI, RES, RET, RETEP, RETES, RETI
     7, RETMP, RETMS, RETP, RETS, R1C, SGAM, SALP, SKTC, SKTS, TEPE,
     8 THMAX, TKTN
      DST = TM - (TM - TE)*(DSMT/DSME)**2
      DSS = DST*(KM - KTC) - DSMT
      CS = (KE - KM)/DSSE
   10 DK = CS*DSS
      CALL EPSLON(KM,DK,RCMS,DSS,DRCS,RES)
      DRCT = DRCMST + DRCS
      RT = RCMS + DRCS
      RET = RES + RT*EMT
      CALL TANKAP(RCT,DRCT,RET,TK)
      TKD = (TK - TKTN)/(1.0 + TK*TKTN)
      IF (ABS(DST*TKD).LT.1.0E-05) GO TO 20
      DST = RET/(CKTC - SKTC*TKD)
      DSS = DSS + DST*TKD*SQRT(1.0 + TKD**2)/(1.0 -(DK +KM -KTC)**2/2.0)
      GO TO 10
   20 KT = KM + DK
      RE = RT*RETI/RCT + RET
      DS = DSS - DSSE
      IF (DSSE.GT.0.0) DS = -DS
      HKTS = KT/2.0
      SHKTS = HKTS*SRS(HKTS)
      SHKTSQ = SHKTS**2
      CHKTS = SQRT(1.0 - SHKTSQ)
      SKTS = 2.0*SHKTS*CHKTS
      CKTS = 1.0 - 2.0*SHKTSQ
      RETURN
      END
//
```

```
      FUNCTION YPC(DR,RE,EP)
C ***      Y PLOT COORDINATE FROM R AND EPSILON COORDINATES
      YPC = DR - RE*EP/2.0*(1.0 - EP**2/12.0*(1.0 - EP**2/30.0))
      RETURN
      END
      SUBROUTINE COPES(IFLAG)
C     ******************************************************************
C             COPES - CONTROL PROGRAM FOR ENGINEERING SYNTHESIS.
C     ******************************************************************
      COMMON /CNMN1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAX
     1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,IT
     2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER
      COMMON /COPES1/ TITLE(20)
      COMMON /COPES2/ RA(5000),IA(1000)
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,M
     12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR
     21,ISCR2,NXAPRX,NPS,NPFS,NPA,NF,INOM,IPAPRX,KMIN,KMAX,XFACT1,XFACT2
     3,NAN2,NAN3,NPMAX,NPTOT,JNOM,MAXTRM
      COMMON /GLOBCM/ ARRAY(1500)
C     BY G. N. VANDERPLAATS                      OCT., 1974.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C     NCALC OPTIONS:
C          0. READ ALL INPUT AND STOP.
C          1. SINGLE PASS ANALYSIS.
C          2. OPTIMIZATION.
C          3. SENSITIVITY - Z = F(X).
C          4. TWO VARIABLE FUNCTION SPACE - Z = F(X,Y).
C          5. OPTIMUM SENSITIVITY.
C          6. ANALYSIS/OPTIMIZATION USING APPROXIMATION TECHNIQUES.
C
C
C  NEAR,INC REVISION,          JULY, 1982
C     -  COPES CHANGED TO A SUBROUTINE CONTROLLED BY TSOPT DRIVER
C     -  CONTROL PARAMETER IFLAG ADDED
C          IFLAG = 1,  EXECUTE COPES FROM BEGINNING, READING ALL INPUT
C          IFLAG = 2,  SKIP INPUT AND INITIALIZATION
C
C
C     ------------------------------------------------------------------
C     **********                  INPUT                       **********
C     ------------------------------------------------------------------
      NAN2=0
      NAN3=0
      IF(IFLAG.EQ.2) GO TO 15
C     DIMENSIONS OF ARRAYS ARRAY, RA AND IA.
      NARRAY=1500
      NDRA=5000
      NDIA=1000
C     READ GENERAL SYNTHESIS CONTROL INPUT.
C     SCRATCH TAPE NUMBERS.
      ISCR1=20
      ISCR2=40
      CALL COPE01 (RA,IA,NDRA,NDIA)
      IF (NCALC.LT.0.OR.NCALC.GT.6) GO TO 340
C     CHECK TO INSURE STORAGE REQUIREMENTS DO NOT EXCEED
C     DIMENSIONED SIZES OF ARRAYS RA AND IA.
      NDRA1=LOCR(25)
      NDIA1=LOCI(25)
      IF (NDRA1.LE.NDRA.AND.NDIA1.LE.NDIA) GO TO 10
      WRITE (6,360) NDRA,NDRA1,NDIA,NDIA1
      GO TO 340
   10 CONTINUE
C     READ USER INPUT.
      ICALC=1
      CALL ANALIZ (ICALC)
      IF (NCALC.LT.1.OR.NCALC.GT.6) GO TO 340
C     ------------------------------------------------------------------
C     **********                EXECUTION                     **********
C     ------------------------------------------------------------------
   15 CONTINUE
      ICALC=2
      JCALC=3
      IF (NCALC.NE.2.AND.NCALC.LT.5) GO TO 60
C     ------------------------------------------------------------------
C     IF ABS(X(I)).GT.0 OVER-RIDE USER INPUT OF DECISION VARIABLES FOR
C     OPTIMIZATION.
C     ------------------------------------------------------------------
      DO 40 I=1,NDV
      XX=ABS(RA(I))
C     OVER-RIDE ANALIZ INPUT.
      N5=LOCR(5)
      M2=LOCI(2)
      DO 20 J=1,NDVTOT
      NN1=IA(M2)
      M2=M2+1
      IF (NN1.NE.I) GO TO 20
      NN1=IA(J)
      IF (XX.LT.1.0E-10) GO TO 30
      ARRAY(NN1)=RA(I)*RA(N5)
   20 N5=N5+1
      GO TO 40
   30 RA(I)=ARRAY(NN1)/RA(N5)
   40 CONTINUE
C     TRANSFER DESIGN VARIABLES TO ARRAY.
      M2=LOCI(2)
      N5=LOCR(5)
      DO 50 I=1,NDVTOT
      N=IA(M2)
      M=IA(I)
      ARRAY(M)=RA(N)*RA(N5)
      N5=N5+1
   50 M2=M2+1
   60 CONTINUE
      IF (NCALC.NE.3.AND.NCALC.NE.5) GO TO 80
C     ------------------------------------------------------------------
C     TRANSFER NOMINAL VALUES OF SENSITIVITY VARIABLES TO ARRAY.
C     ------------------------------------------------------------------
      M16=LOCI(16)
      M17=LOCI(17)
      N15=LOCR(15)
      DO 70 I=1,NSV
      NN=IA(M16)
      M16=M16+1
      ARRAY(NN)=RA(N15)
      N15=N15+IA(M17)
   70 M17=M17+1
   80 CONTINUE
      IF (NCALC.LT.6) GO TO 290
C     ------------------------------------------------------------------
C                INITIALIZATION FOR APPROXIMATE ANALYSIS/OPTIMIZATION
```

```
C     ------------------------------------------------------------------
      IF (NPA.EQ.0) GO TO 130
C     ANALIZ INPUT DEFINES AN X-VECTOR.
      M5=LOCI(5)
      N23=LOCR(23)
      DO 120 I=1,NXAPRX
      J=IA(M5)
C     IS THIS A DESIGN VARIABLE.
      DO 90 K=1,NDVTOT
      KK=K
      IF (IA(K).EQ.J) GO TO 100
90    CONTINUE
C     NO.
      AMULT=1.
      GO TO 110
C     YES.
100   K=LOCR(5)+KK-1
      AMULT=RA(K)
110   RA(N23)=ARRAY(J)/AMULT
      M5=M5+1
120   N23=N23+1
130   CONTINUE
      IF (NPS.GT.0.OR.NPFS.GT.0) GO TO 190
C     ONLY ONE DESIGN VECTOR IS AVAILABLE.  CREATE A SECOND X-VECTOR
C     SO OPTIMIZATION CAN PROCEED.
      N23=LOCR(23)
      N24=N23+NXAPRX
      M5=LOCI(5)
      DO 180 I=1,NXAPRX
C     GLOBAL LOCATION.
      IG=IA(M5)
      M5=M5+1
C     PROPOSED X-VALUE.
      XX=1.1*RA(N23)
      IF (ABS(XX).LT.1.0E-10) XX=.1
C     IS THIS A DESIGN VARIABLE.
      N5=LOCR(5)
      DO 140 J=1,NDVTOT
      JJ=J
      AMJ=RA(N5)
      IF (IA(J).EQ.IG) GO TO 150
C     NO.
      N5=N5+1
140   CONTINUE
      GO TO 170
150   CONTINUE
C     YES.  WHICH DESIGN VARIABLE IS IT.
      ID=LOCI(2)+JJ-1
      ID=IA(ID)
C     INSURE XX IS WITHIN BOUNDS.
      N2=LOCR(2)+ID-1
      N3=LOCR(3)+ID-1
      BL=RA(N2)*ABS(AMJ)
      BU=RA(N3)*ABS(AMJ)
      IF (BL.LE.BU) GO TO 160
      SAV=BL
      BL=BU
      BU=SAV
160   IF (XX.LT.BL) XX=BL
      IF (XX.GT.BU) XX=BU
      DX=RA(N23)-XX
      IF (ABS(DX).LT.1.0E-6) XX=1.001*RA(N23)
      IF (ABS(XX).LT.1.0E-6) XX=.001
170   RA(N24)=XX
      N23=N23+1
180   N24=N24+1
190   CONTINUE
      REWIND ISCR2
      NPSA=NPS+NPA
      IF (NPSA.EQ.0) GO TO 250
      IF (NPS.EQ.0) GO TO 210
C     READ X-VECTORS.
      NXI=LOCR(23)+NPA*NXAPRX
      DO 200 J=1,NPS
      NXIJ=NXI+NXAPRX-1
      READ (ISCR2) (RA(I),I=NXI,NXIJ)
200   NXI=NXI+NXAPRX
210   CONTINUE
      IF (NPFS.LE.0) NPSA=NPTOT
      NXI=LOCR(23)
      NY=NXI+NXAPRX*NPTOT
      DO 240 J=1,NPSA
C     TRANSFER X-VALUES.
      M5=LOCI(5)
      I1=NXI
      DO 220 I=1,NXAPRX
      II=IA(M5)
      ARRAY(II)=RA(I1)
      M5=M5+1
220   I1=I1+1
C     ANALIZE.
      NAN2=NAN2+1
      CALL ANALIZ (ICALC)
C     PUT FUNCTION VALUES IN Y-ARRAY.
      M6=LOCI(6)
      I1=NY
      DO 230 I=1,NF
      II=IA(M6)
      RA(I1)=ARRAY(II)
      I1=I1+1
230   M6=M6+1
      NXI=NXI+NXAPRX
      NY=NY+NF
240   CONTINUE
250   CONTINUE
      IF (NPFS.LE.0) GO TO 270
      NXI=LOCR(23)+NPSA*NXAPRX
      NY=LOCR(23)+NXAPRX*NPTOT+NF*NPSA
C     READ X AND Y VECTORS.
      DO 260 J=1,NPFS
      NXIJ=NXI+NXAPRX-1
      NYJ=NY+NF-1
C     X-VECTOR.
      READ (ISCR2) (RA(I),I=NXI,NXIJ)
C     Y-VECTOR.
      READ (ISCR2) (RA(I),I=NY,NYJ)
      NXI=NXI+NXAPRX
260   NY=NY+NF
270   CONTINUE
C     PUT X-F PAIRS BACK ON ISCR2.
```

```
      REWIND ISCR2                                                      COPES237
      NXI=LOCR(23)                                                      COPES238
      NY=NXI+NXAPRX*NPTOT                                               COPES239
      DO 280 I=1,NPTOT                                                  COPES240
      NXIJ=NXI+NXAPRX-1                                                 COPES241
      NYJ=NY+NF-1                                                       COPES242
C     X-VECTOR.                                                         COPES243
      WRITE (ISCR2) (RA(J),J=NXI,NXIJ)                                  COPES244
C     Y-VECTOR.                                                         COPES245
      WRITE (ISCR2) (RA(J),J=NY,NYJ)                                    COPES246
      NXI=NXIJ+1                                                        COPES247
280   NY=NYJ+1                                                          COPES248
290   CONTINUE                                                          COPES249
      GO TO (300,300,310,320,310,330),NCALC                             COPES250
C     ------------------------------------------------------------------COPES251
C                              ONE ANALYSIS                             COPES252
C     ------------------------------------------------------------------COPES253
300   NAN2=NAN2+1                                                       COPES254
      CALL ANALIZ (ICALC)                                               COPES255
      IF(NCALC.GT.1) GO TO 305                                          COPES256
      NAN3=NAN3+1                                                       COPES257
      CALL ANALIZ (JCALC)                                               COPES258
      IF (NCALC.EQ.1) GO TO 340                                         COPES259
C     ------------------------------------------------------------------COPES260
C                                OPTIMIZATION                           COPES261
C     ------------------------------------------------------------------COPES262
C     OPTIMIZATION.                                                     COPES263
  305 CONTINUE                                                          COPES264
      CALL COPE02 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)                        COPES265
C     OUTPUT RESULTS.                                                   COPES266
      CALL COPE18 (IOBJ,NDVTOT,NCONA,RA,IA,LOCR,LOCI,ARRAY)             COPES267
      NAN3=NAN3+1                                                       COPES268
      CALL ANALIZ (JCALC)                                               COPES269
      GO TO 340                                                         COPES270
C     ------------------------------------------------------------------COPES271
C                          SENSITIVITY ANALYSIS                         COPES272
C     ------------------------------------------------------------------COPES273
310   CONTINUE                                                          COPES274
C     ARRAY STARTING LOCATIONS.                                         COPES275
C     SENS.                                                             COPES276
      N1=LOCR(15)                                                       COPES277
      NN8=LOCR(16)-N1                                                   COPES278
C     NSENSZ.                                                           COPES279
      N2=LOCI(15)                                                       COPES280
      NN9=NSOBJ                                                         COPES281
C     ISENS.                                                            COPES282
      N3=LOCI(16)                                                       COPES283
      NN10=NSV                                                          COPES284
C     NSENS.                                                            COPES285
      N4=LOCI(17)                                                       COPES286
C     TEMP.                                                             COPES287
      N5=LOCR(23)                                                       COPES288
      CALL COPE04 (ARRAY,NARRAY,RA(N1),IA(N2),IA(N3),IA(N4),RA(N5),NN8,NCOPES289
     1N9,NN10,RA,IA,NDRA,NDIA)                                          COPES290
      CALL COPE04 (ARRAY,NARRAY,RA(N1),IA(N2),IA(N3),IA(N4),RA(N5),NN8,NCOPES289
     1N9,NN10,RA,IA,NDRA,NDIA)                                          COPES290
C     OUTPUT RESULTS.                                                   COPES291
      CALL COPE05 (RA,IA,NDRA,NDIA,ISCR1)                               COPES292
      GO TO 340                                                         COPES293
320   CONTINUE                                                          COPES294
C     ------------------------------------------------------------------COPES295
C                     TWO VARIABLE FUNCTION SPACE                       COPES296
C     ------------------------------------------------------------------COPES297
      CALL COPE06 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)                        COPES298
C     OUTPUT RESULTS.                                                   COPES299
      CALL COPE07 (RA,IA,NDRA,NDIA,ISCR1)                               COPES300
      GO TO 340                                                         COPES301
330   CONTINUE                                                          COPES302
C     ------------------------------------------------------------------COPES303
C                 APPROXIMATE ANALYSIS/OPTIMIZATION.                    COPES304
C     ------------------------------------------------------------------COPES305
      CALL COPE09                                                       COPES306
C     OUTPUT RESULTS.                                                   COPES307
      CALL COPE14 (NXAPRX,NF,NPTOT,RA,IA,LOCR,LOCI,TITLE,INOM,NDV,IPAPRXCOPES308
     1,ISCR2,MAXTRM)                                                    COPES309
      IF (KMAX.LT.0) GO TO 340                                          COPES310
      CALL COPE18 (IOBJ,NDVTOT,NCONA,RA,IA,LOCR,LOCI,ARRAY)             COPES311
      NAN3=NAN3+1                                                       COPES312
      CALL ANALIZ (JCALC)                                               COPES313
340   CONTINUE                                                          COPES314
      WRITE (6,350) NAN2,NAN3                                           COPES315
      REWIND ISCR1                                                      COPES316
      REWIND ISCR2                                                      COPES317
      RETURN                                                            COPES318
C     ------------------------------------------------------------------COPES319
C                                FORMATS                                COPES320
C     ------------------------------------------------------------------COPES321
350   FORMAT (1H1,4X,23HPROGRAM CALLS TO ANALIZ//8X,5HICALC,3X,5HCALLS/1COPES322
     10X,1H1,7X,1H1/10X,1H2,I8/10X,1H3,I8)                              COPES323
360   FORMAT (//5X,60HREQUIRED STORAGE FOR ARRAY RA OR IA EXCEEDS DIMENSCOPES324
     1IONED SIZE/5X,5HARRAY,2X,9HDIMENSION,2X,8HREQUIRED/7X,2HRA,I8,6X,ICOPES325
     25/7X,2HIA,I8,6X,I5//5X,22H* * PROGRAM TERMINATED)                 COPES326
      END                                                               COPES327
      SUBROUTINE COPE01 (RA,IA,NDRA,NDIA)                               COPES328
      COMMON /CNMN1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAXCOPES329
     1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,ITCOPES330
     2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER                               COPES331
      COMMON /COPES1/ ATITLE(20)                                        COPES332
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,MCOPES333
     12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCRCOPES334
     21,ISCR2,NXAPRX,NPS,NPFS,NPA,NF,INOM,IPAPRX,KMIN,KMAX,XFACT1,XFACT2COPES335
     3,NAN2,NAN3,NPMAX,NPTOT,JNOM,MAXTRM                                COPES336
      DIMENSION RA(NDRA), IA(NDIA), TITLE(20)                           COPES337
      DATA END1/1HE/,END2/1HN/,END3/1HD/                                COPES338
      DATA COM/1H$/,BLANK/1H /                                          COPES339
C     ******************************************************************COPES340
C     ROUTINE TO READ CONTROL INPUT FOR COPES.                          COPES341
C     ******************************************************************COPES342
C     BY G. N. VANDERPLAATS                          MAR., 1973.        COPES343
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.                  COPES344
C     ------------------------------------------------------------------COPES345
C     READ CARD IMAGES AND STORE ON UNIT ISCR2.  STORE ON UNIT ISCR1    COPES346
C     WITHOUT COMMENT CARDS                                             COPES347
C     ------------------------------------------------------------------COPES348
      REWIND ISCR1                                                      COPES349
      REWIND ISCR2                                                      COPES350
      NCARDS=0                                                          COPES351
      LOCI(25)=0                                                        COPES352
      NCOM=0                                                            COPES353
      ICARD=0                                                           COPES354
```

```
10    READ (5,580) (RA(I),I=1,80)                                       COPES355
      ICARD=ICARD+1                                                     COPES356
      NCARDS=NCARDS+1                                                   COPES357
      WRITE (ISCR2,590) NCARDS,(RA(I),I=1,80)                           COPES358
      IF (RA(1).EQ.COM) GO TO 10                                        COPES359
      IF (RA(1).EQ.END1.AND.(RA(2).EQ.END2.AND.RA(3).EQ.END3)) GO TO 20 COPES360
      IF (NCOM.NE.0) GO TO 30                                           COPES361
C     TITLE OR END CARD.                                                COPES362
20    WRITE (ISCR1,580) (RA(I),I=1,80)                                  COPES363
      IF (NCOM.GT.0) GO TO 70                                           COPES364
C     IT WAS THE TITLE CARD.                                            COPES365
      NCOM=1                                                            COPES366
      GO TO 10                                                          COPES367
30    CONTINUE                                                          COPES368
C     FORMAT DATA AS REQUIRED.                                          COPES369
      NA=1                                                              COPES370
      NB=81                                                             COPES371
      CALL COPE08 (RA(NA),RA(NB),IFORM,NFLD)                            COPES372
C     DETERMINE NUMBER OF CARDS OF DATA AND ADD BLANKS TO FILL.         COPES373
      NBC=(NFLD-1)/8+1                                                  COPES374
      NB=80*NBC+80                                                      COPES375
      NA=10*NFLD+81                                                     COPES376
      IF (NA.GE.NB) GO TO 50                                            COPES377
      DO 40 I=NA,NB                                                     COPES378
40    RA(I)=BLANK                                                       COPES379
50    CONTINUE                                                          COPES380
      WRITE (ISCR1,580) (RA(I),I=81,NB)                                 COPES381
      IF (IFORM.GT.0) GO TO 10                                          COPES382
C     DATA WAS NOT PREVIOUSLY FORMATTED.                                COPES383
      N1=1                                                              COPES384
      DO 60 II=1,NBC                                                    COPES385
      N1=N1+80                                                          COPES386
      N2=N1+79                                                          COPES387
      WRITE (ISCR2,590) NCARDS,(RA(I),I=N1,N2)                          COPES388
60    ICARD=ICARD+1                                                     COPES389
      GO TO 10                                                          COPES390
70    CONTINUE                                                          COPES391
      REWIND ISCR1                                                      COPES392
      REWIND ISCR2                                                      COPES393
C     ------------------------------------------------------------------COPES394
C                     GENERAL SYNTHESIS INFORMATION                     COPES395
C     ------------------------------------------------------------------COPES396
C     TITLE.                                                            COPES397
C --- DATA BLOCK A.                                                     COPES398
      READ (ISCR1,1190) (ATITLE(I),I=1,20)                              COPES399
C     CONTROL PARAMETERS.                                               COPES400
C --- DATA BLOCK B.                                                     COPES401
      READ (ISCR1,1200) NCALC,NDV,NSV,N2VAR,NXAPRX,IPNPUT,IPDBG         COPES402
      IF (NCALC.LT.0.OR.NCALC.GT.6) WRITE (6,1220) NCALC                COPES403
      IF (NCALC.LT.0.OR.NCALC.GT.6) RETURN                              COPES404
      IF (IPNPUT.GT.1) GO TO 100                                        COPES405
      WRITE (6,970)                                                     COPES406
      WRITE (6,980)                                                     COPES407
      WRITE (6,990) (ATITLE(I),I=1,20)                                  COPES408
C     ------------------------------------------------------------------COPES409
C                          CARD IMAGE PRINT                             COPES410
C     ------------------------------------------------------------------COPES411
      IF (IPNPUT.GT.0) GO TO 90                                         COPES412
      WRITE (6,870)                                                     COPES413
      WRITE (6,880)                                                     COPES414
      DO 80 I=1,ICARD                                                   COPES415
      READ (ISCR2,590) NCARDS,(RA(J),J=1,80)                            COPES416
80    WRITE (6,890) NCAPDS,(RA(J),J=1,80)                               COPES417
      REWIND ISCR2                                                      COPES418
90    CONTINUE                                                          COPES419
      WRITE (6,1000) (ATITLE(I),I=1,20)                                 COPES420
      WRITE (6,1010) NCALC,NDV,NSV,N2VAR,NXAPRX,IPNPUT,IPDBG            COPES421
      WRITE (6,910)                                                     COPES422
100   NACMX1=0                                                          COPES423
      NDVTOT=0                                                          COPES424
      NCONA=0                                                           COPES425
      NCON=0                                                            COPES426
      IF (NDV.LE.0) GO TO 270                                           COPES427
C     ------------------------------------------------------------------COPES428
C                       OPTIMIZATION INFORMATION                        COPES429
C     ------------------------------------------------------------------COPES430
C     OPTIMIZATION CONTROL VARIABLES. - CONMIN DEPENDENT.               COPES431
C --- DATA BLOCK C.                                                     COPES432
      READ (ISCR1,1200) IPRINT,ITMAX,ICNDIR,NSCAL,ITRM,LINOBJ,NACMX1,NFDCOPES433
     1G                                                                 COPES434
C --- DATA BLOCK D.                                                     COPES435
      READ (ISCR1,1210) FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,DELFUN,COPES436
     1DABFUN,ALPHAX,ABOBJ1                                              COPES437
C --- DATA BLOCK E.                                                     COPES438
C     TOTAL NO. OF D. V., OBJECTIVE GLOBAL NUMBER, SIGN                 COPES439
C     ON OPTIMIZATION OBJECTIVE.                                        COPES440
      READ (ISCR1,920) NDVTOT,IOBJ,SGNOPT                               COPES441
      IF (NDVTOT.LT.NDV) NDVTOT=NDV                                     COPES442
      IF (NCALC.EQ.6.AND.NACMX1.EQ.0) NACMX1=2*NDV+2                    COPES443
      IF (NACMX1.LE.0) NACMX1=NDV+2                                     COPES444
      IF (IPNFUT.GE.2) GO TO 110                                        COPES445
      IF (ABS(SGNOPT).LT.1.0E-10) SGNOPT=-1.                            COPES446
      WRITE (6,1070) IOBJ,SGNOPT                                        COPES447
      WRITE (6,760) IFPRINT,ITMAX,ICNDIR,NSCAL,ITRM,LINOBJ,NACMX1,NFDG  COPES448
      WRITE (6,770) FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,THETA,PHI,DELFUN,DABFCOPES449
     1UN,ALPHAX,ABOBJ1                                                  COPES450
110   N2=NDV+3                                                          COPES451
      N3=N2+NDV+2                                                       COPES452
      N4=N3+NDV+2                                                       COPES453
C --- DATA BLOCK F.                                                     COPES454
C     DESIGN VARIABLE INFORMATION, LB, UB, INITIAL VALUE, SCAL.         COPES455
      IF (IPNPUT.LT.2) WRITE (6,1080)                                   COPES456
      N5=N4+NDV+2                                                       COPES457
      IF (N5.LE.NDRA) GO TO 120                                         COPES458
      WRITE (6,780)                                                     COPES459
      WRITE (6,790)                                                     COPES460
      LOCR(25)=N5                                                       COPES461
      GO TO 550                                                         COPES462
120   CONTINUE                                                          COPES463
      NSIDE=0                                                           COPES464
      DO 130 I=1,NDV                                                    COPES465
      READ (ISCR1,1060) RA(N2),RA(N3),RA(I),RA(N4),(TITLE(J),J=1,5)     COPES466
      IF (RA(N2).GT.-1.0E+15.OR.PA(N3).LT.1.0E+15) NSIDE=1              COPES467
      IF (RA(N2).LE.-1.0E+15) RA(N2)=-1.1E+15                           COPES468
      IF (RA(N3).GE.1.0E+15) PA(N3)=1.1E+15                             COPES469
      IF (IPNFUT.LT.2) WRITE (6,1090) I,RA(N2),RA(N3),RA(I),RA(N4),(TITLCOPES470
     1E(J),J=1,5)                                                       COPES471
      N2=N2+1                                                           COPES472
      N3=N3+1                                                           COPES473
      N4=N4+1                                                           COPES474
```

```
  130 CONTINUE
C --- DATA BLOCK G.
C     D. V. NO., GLOBAL LOCATION, MULTIPLYING FACTOR.
      IF (IPNPUT.LT.2) WRITE (6,930)
      N5=4*NDV+9
      M2=NDVTOT+1
      N6=N5+NDVTOT
      M3=M2+NDVTOT
      IF (N6.LE.NDRA) GO TO 140
      WRITE (6,780)
      WRITE (6,800)
      LOCR(25)=N5
      GO TO 550
  140 CONTINUE
      IF (M3.LE.NDIA) GO TO 150
      WRITE (6,810)
      WRITE (6,800)
      LOCI(25)=M3
      GO TO 550
  150 CONTINUE
      DO 160 I=1,NDVTOT
      READ (ISCR1,920) IA(M2),IA(I),RA(N5)
      IF (ABS(RA(N5)).LT.1.0E-20) RA(N5)=1.0
      IF (IPNPUT.LT.2) WRITE (6,940) I,IA(M2),IA(I),RA(N5)
      M2=M2+1
      N5=N5+1
  160 CONTINUE
      NCON=0
C --- DATA BLOCK H.
C     NUMBER OF CONSTRAINT SETS.
      READ (ISCR1,920) NCONS
      IF (IPNPUT.LT.2) WRITE (6,1110)
      IF (IPNPUT.LT.2) WRITE (6,1120) NCONS
      IF (NCONS.EQ.0) GO TO 270
      IF (IPNPUT.LT.2) WRITE (6,1130)
      N6=4*NDV+NDVTOT+9
      M3=2*NDVTOT+1
      M4=2*NDVTOT+NCONS
      M4A=M4+1
      L=1
C --- DATA BLOCK I.
      DO 240 I=1,NCONS
      NNN=N6+3
      IF (NNN.GT.NDRA) GO TO 250
C     GLOBAL NO. 1, GLOBAL NO. 2, LINEAR CONSTRAINT ID.
      READ (ISCR1,1200) ICONI,JCONI,LCONI
C     LB, NORM, UB, NORM.
      READ (ISCR1,1210) (RA(J),J=N6,NNN)
      IF (RA(N6).LE.-1.0E+15) RA(N6)=-1.1E+15
      IF (RA(N6+2).GE.1.0E+15) RA(N6+2)=1.1E+15
      IF (RA(N6+1).LT.1.0E-20) RA(N6+1)=ABS(RA(N6))
      IF (RA(N6+1).LT.1.0E-20) RA(N6+1)=0.1
      IF (RA(N6+3).LT.1.0E-20) RA(N6+3)=ABS(RA(N6+2))
      IF (RA(N6+3).LT.1.0E-20) RA(N6+3)=0.1
C     NUMBER OF VARIABLES IN THIS SET.
      NVAR=JCONI-ICONI+1
      IF (NVAR.LT.1) NVAR=1
      NCONA=NCONA+NVAR
C     HOW MANY CONSTRAINTS?
      J1=0
      IF (RA(N6).GE.-1.0E+15) J1=1
      IF (RA(N6+2).LT.1.0E+15) J1=J1+1
      NCONI=J1*NVAR
      NCON=NCON+NCONI
      IF (J1.EQ.0) GO TO 180
C     ADD LINEAR CONSTRAINT IDENTIFIERS TO ISC.
      DO 170 J=1,NCONI
      M4=M4+1
      MMM=M4
      IF (MMM.GT.NDIA) GO TO 260
  170 IA(M4)=LCONI
  180 CONTINUE
C     ADD LB, UB AND SCAL TO BLU IF NVAR.GT.1.
      IF (NVAR.EQ.1) GO TO 200
      NVAR1=NVAR-1
      DO 190 J=1,NVAR1
      NNN=N6+7
      IF (NNN.GT.NDRA) GO TO 250
      RA(N6+4)=RA(N6)
      RA(N6+5)=RA(N6+1)
      RA(N6+6)=RA(N6+2)
      RA(N6+7)=RA(N6+3)
      N6=N6+4
  190 CONTINUE
  200 CONTINUE
C     ADD CONSTRAINED VARIABLE GLOBAL IDENTIFIERS TO ICON.
      ICON1=ICONI
      MMM=M4+NVAR-1
      IF (MMM.GT.NDIA) GO TO 260
      DO 230 J=1,NVAR
      IF (J.EQ.1) GO TO 220
C     SHIFT ISC VECTOR.
      L1=M4+1
      L2=M4
      DO 210 K=M4A,M4
      IA(L1)=IA(L2)
      L1=L1-1
  210 L2=L2-1
      M4=M4+1
      M4A=M4A+1
  220 IA(M3)=ICON1
      ICON1=ICON1+1
  230 M3=M3+1
      IF (IPNPUT.LT.2) WRITE (6,1100) L,ICONI,JCONI,LCONI,RA(N6),RA(N6+1
     1),RA(N6+2),RA(N6+3)
      N6=N6+4
      L=NCON+1
  240 CONTINUE
      IF (IPNPUT.LT.2) WRITE (6,900) NCONA
      GO TO 270
  250 WRITE (6,780)
      WRITE (6,820)
      LOCR(25)=NNN
      GO TO 550
  260 WRITE (6,810)
      WRITE (6,820)
      LOCI(25)=MMM
      GO TO 550
  270 CONTINUE
C     STARTING LOCATIONS FOR APPROXIMATION INFORMATION.
```

```
      NAPR=4*NDV+NDVTOT+4*NCONA+9                                       COPES595
      NAPI=2*(NDV+NCONA)+2*NDVTOT+NCONA+1                               COPES596
      NF=0                                                              COPES597
      KMAX=0                                                            COPES598
      NPTOT=0                                                           COPES599
      MAXTRM=0                                                          COPES600
      IF (NXAPRX.LE.0) GO TO 450                                        COPES601
C     ------------------------------------------------------------------COPES602
C                    APPROXIMATE ANALYSIS/DESIGN                        COPES603
C     ------------------------------------------------------------------COPES604
C --- DATA BLOCK J.                                                     COPES605
C     CONTROL PARAMETERS.                                               COPES606
      READ (ISCR1,1200) NF,NPS,NPFS,NPA,INOM,ISCRX,ISCRXF,IPAPRX        COPES607
      IF (NPA.NE.0) NPA=1                                               COPES608
      IF (NPS.EQ.0.AND.NPFS.EQ.0) NPA=1                                 COPES609
      IF (ISCRX.EQ.0) ISCRX=5                                           COPES610
      IF (ISCRXF.EQ.0) ISCRXF=5                                         COPES611
      IF (IPNPUT.LT.2) WRITE (6,600) NF,NPS,NPFS,NPA,INOM,ISCRX,ISCRXF,ICOPES612
     1PAPRX                                                             COPES613
      NPSFS=NPS+NPFS                                                    COPES614
      IF (NPSFS.LT.2) NPA=1                                             COPES615
      NPTOT=NPS+NPFS+NPA                                                COPES616
      IF (NPTOT.LT.2) NPTOT=2                                           COPES617
      READ (ISCR1,1200) KMIN,KMAX,NPMAX,JNOM,INXLOC,INFLOC,MAXTRM       COPES618
      IF (INXLOC.EQ.0) NXAPRX=NDVTOT                                    COPES619
      M=NXAPRX+(NXAPRX*(NXAPRX+1))/2                                    COPES620
      IF (NPMAX.LE.0) NPMAX=2*M                                         COPES621
      IF (KMAX.EQ.0) KMAX=3*M-NPTOT+1                                   COPES622
      IF (KMIN.EQ.0) KMIN=2*NDV-NPTOT+1                                 COPES623
      IF (KMIN.LT.0) KMIN=0                                             COPES624
      IF (KMAX.GT.0.AND.KMAX.LT.KMIN) KMAX=KMIN                         COPES625
      IF (JNOM.EQ.0) JNOM=2*M                                           COPES626
      IF (MAXTRM.LT.1) MAXTRM=3                                         COPES627
      IF (IPNPUT.LT.2) WRITE (6,610) KMIN,KMAX,NPMAX,JNOM,INXLOC,INFLOC,COPES628
     1MAXTRM                                                            COPES629
C --- DATA BLOCK K, PART 1.                                             COPES630
C     DELX BOUNDS ON APPROXIMATE OPTIMIZATION.                          COPES631
      IF (NDV.LE.0) GO TO 290                                           COPES632
      N7=NAPR                                                           COPES633
      NN7=N7+NDV-1                                                      COPES634
      IF (NN7.LE.NDRA) GO TO 280                                        COPES635
      WRITE (6,780)                                                     COPES636
      WRITE (6,560)                                                     COPES637
  280 CONTINUE                                                          COPES638
      READ (ISCR1,1210) (RA(I),I=N7,NN7)                                COPES639
      IF (IPNPUT.LT.2) WRITE (6,570)                                    COPES640
      IF (IPNPUT.LT.2) WRITE (6,1160) (RA(I),I=N7,NN7)                  COPES641
C --- DATA BLOCK K, PART 2.                                             COPES642
C     MULTIPLIERS ON DELX.                                              COPES643
      READ (ISCR1,1210) XFACT1,XFACT2                                   COPES644
      IF (XFACT1.LT.1.0E-10) XFACT1=1.5                                 COPES645
      IF (XFACT2.LT.1.0E-10) XFACT2=2.                                  COPES646
      IF (IPNPUT.LT.2) WRITE (6,620) XFACT1,XFACT2                      COPES647
  290 CONTINUE                                                          COPES648
C --- DATA BLOCK L.                                                     COPES649
C     GLOBAL LOCATIONS OF X-VARIABLES.                                  COPES650
      M5=NAPI                                                           COPES651
      MM5=M5+NXAPRX-1                                                   COPES652
      IF (MM5.LE.NDIA) GO TO 300                                        COPES653
      WRITE (6,780)                                                     COPES654
      WRITE (6,630)                                                     COPES655
      LOCI(25)=MM5                                                      COPES656
      GO TO 550                                                         COPES657
  300 CONTINUE                                                          COPES658
      IF (INXLOC.EQ.0) GO TO 310                                        COPES659
      READ (ISCR1,1200) (IA(I),I=M5,MM5)                                COPES660
      GO TO 330                                                         COPES661
  310 CONTINUE                                                          COPES662
C     X-LOCATIONS ARE DEFAULTED TO DESIGN VARIABLE LOCATIONS.           COPES663
      DO 320 I=1,NXAPRX                                                 COPES664
      IA(M5)=IA(I)                                                      COPES665
  320 M5=M5+1                                                           COPES666
      M5=NAPI                                                           COPES667
  330 CONTINUE                                                          COPES668
      IF (IPNPUT.LT.2) WRITE (6,640)                                    COPES669
      IF (IPNPUT.LT.2) WRITE (6,1180) (IA(I),I=M5,MM5)                  COPES670
C --- DATA BLOCK M.                                                     COPES671
C     GLOBAL LOCATIONS OF FUNCTIONS.                                    COPES672
      M6=NAPI+NXAPRX                                                    COPES673
      MM6=M6+NF-1                                                       COPES674
      IF (MM6.LE.NDIA) GO TO 340                                        COPES675
      WRITE (6,780)                                                     COPES676
      WRITE (6,650)                                                     COPES677
      LOCI(25)=MM6                                                      COPES678
      GO TO 550                                                         COPES679
  340 CONTINUE                                                          COPES680
      IF (INFLOC.EQ.0) GO TO 350                                        COPES681
      READ (ISCR1,1200) (IA(I),I=M6,MM6)                                COPES682
      GO TO 380                                                         COPES683
  350 CONTINUE                                                          COPES684
C     FUNCTION LOCATIONS ARE DEFAULTED TO OBJECTIVE AND CONSTRAINT      COPES685
C     LOCATIONS.                                                        COPES686
      NF1=1                                                             COPES687
      M3=2*NDVTOT+1                                                     COPES688
      IA(M6)=IOBJ                                                       COPES689
      IF (NCONA.EQ.0) GO TO 370                                         COPES690
      DO 360 I=1,NCONA                                                  COPES691
      IF (IA(M3).EQ.IOBJ) GO TO 360                                     COPES692
      NF1=NF1+1                                                         COPES693
      M6=M6+1                                                           COPES694
      IA(M6)=IA(M3)                                                     COPES695
  360 M3=M3+1                                                           COPES696
  370 NF=NF1                                                            COPES697
  380 M6=NAPI+NXAPRX                                                    COPES698
      MM6=M6+NF-1                                                       COPES699
      IF (IPNPUT.LT.2) WRITE (6,660)                                    COPES700
      IF (IPNPUT.LT.2) WRITE (6,1180) (IA(I),I=M6,MM6)                  COPES701
C --- DATA BLOCK N.                                                     COPES702
C     READ INPUT X-VECTORS AND STORE ON UNIT ISCR2.                     COPES703
      REWIND ISCR2                                                      COPES704
      IF (NPS.EQ.0) GO TO 410                                           COPES705
      N7=NAPR+NDV                                                       COPES706
      NN7=N7+NXAPRX-1                                                   COPES707
      IF (NN7.LE.NDRA) GO TO 390                                        COPES708
      WRITE (6,780)                                                     COPES709
      WRITE (6,670)                                                     COPES710
      LOCR(25)=NN7                                                      COPES711
      GO TO 550                                                         COPES712
  390 CONTINUE                                                          COPES713
      IF (IPNPUT.LT.2) WRITE (6,680) ISCRX                              COPES714
```

```
      DO 400 I=1,NPS
C     BINARY READ IF ISCRX.NE.5.
      IF (ISCRX.NE.5) READ (ISCRX) (RA(J),J=N7,NN7)
C     FORMATTED READ IF ISCRX.EQ.5.
      IF (ISCRX.EQ.5) READ (ISCR1,1210) (RA(J),J=N7,NN7)
      WRITE (ISCR2) (RA(J),J=N7,NN7)
      IF (IPNPUT.LT.2) WRITE (6,710) I,I
      IF (IPNPUT.LT.2) WRITE (6,1160) (RA(J),J=N7,NN7)
400   CONTINUE
410   CONTINUE
C --- DATA BLOCK O.
C     READ INPUT X-F PAIRS AND STORE ON UNIT ISCR2.
      IF (NPFS.EQ.0) GO TO 440
      N7=NAPR+NDV
      NN7=N7+NXAPRX-1
      NN8=N7+NF-1
      IF (NN7.GT.NN8) NN8=NN7
      IF (NN8.LE.NDRA) GO TO 420
      WRITE (6,780)
      WRITE (6,690)
      LOCR(25)=NN8
      GO TO 550
420   CONTINUE
      NN8=N7+NF-1
      IF (IPNPUT.LT.2) WRITE (6,700) ISCRXF
      DO 430 I=1,NPFS
C     X-VECTOR.
C     BINARY READ IF ISCRXF.NE.5.
      IF (ISCRXF.NE.5) READ (ISCRXF) (RA(J),J=N7,NN7)
C     FORMATTED READ IF ISCRXF.EQ.5.
      IF (ISCRXF.EQ.5) READ (ISCR1,1210) (RA(J),J=N7,NN7)
      II=I+NPS
      IF (IPNPUT.LT.2) WRITE (6,710) I,II
      IF (IPNPUT.LT.2) WRITE (6,720)
      IF (IPNPUT.LT.2) WRITE (6,1160) (RA(J),J=N7,NN7)
      WRITE (ISCR2) (RA(J),J=N7,NN7)
C     FUNCTION VALUES.
C     BINARY READ IF ISCRXF.NE.5.
      IF (ISCRXF.NE.5) READ (ISCRXF) (RA(J),J=N7,NN8)
C     FORMATTED READ IF ISCRXF.EQ.5.
      IF (ISCRXF.EQ.5) READ (ISCR1,1210) (RA(J),J=N7,NN8)
      IF (IPNPUT.LE.2) WRITE (6,730)
      IF (IPNPUT.LT.2) WRITE (6,1160) (RA(J),J=N7,NN8)
      WRITE (ISCR2) (RA(J),J=N7,NN8)
430   CONTINUE
440   CONTINUE
450   CONTINUE
      NSOBJ=0
      NSVTOT=0
C     STARTING LOCATIONS FOR SENSITIVITY INFORMATION.
      NSVR=NAPR+NDV
      NSVI=NAPI+NXAPRX+NF
      IF (NSV.LE.0) GO TO 500
C     -----------------------------------------------------------------
C                             SENSITIVITY INFORMATION
C     -----------------------------------------------------------------
      IF (IPNPUT.LT.2) WRITE (6,1020)
C --- DATA BLOCK P, PART 1.
C     NSOBJ, IPSENS
      READ (ISCR1,1200) NSOBJ,IPSENS
C --- DATA BLOCK P, PART 2.
C     NSENSZ.
      M15=NSVI
      MM15=M15+NSOBJ-1
      IF (MM15.LE.NDIA) GO TO 460
      WRITE (6,810)
      WRITE (6,830)
      LOCI(25)=MM15
      GO TO 550
460   CONTINUE
      READ (ISCR1,1200) (IA(I),I=M15,MM15)
      IF (IPNPUT.LT.2) WRITE (6,960) IPSENS,NSOBJ
      IF (IPNPUT.LT.2) WRITE (6,950) (IA(I),I=M15,MM15)
      IF (IPNPUT.LT.2) WRITE (6,1030)
      N15=NSVR
      M16=NSVI+NSOBJ
      M17=M16+NSV
      DO 490 I=1,NSV
C --- DATA BLOCK Q, PART 1.
C     ISENS, NSENS.
      READ (ISCR1,1200) IA(M16),NN1
      NN15=N15+NN1-1
      IF (NN15.LE.NDRA) GO TO 470
      WRITE (6,780)
      WRITE (6,840)
      LOCR(25)=NN15
      GO TO 550
470   CONTINUE
C --- DATA BLOCK Q, PART 2.
C     SENS.
      READ (ISCR1,1210) (RA(J),J=N15,NN15)
      IF (IPNPUT.GE.2) GO TO 480
      JJ=N15+5
      IF (JJ.GT.NN15) JJ=NN15
      WRITE (6,1040) I,IA(M16),(RA(J),J=N15,JJ)
      JJ=JJ+1
      IF (JJ.LE.NN15) WRITE (6,1050) (RA(J),J=JJ,NN15)
480   CONTINUE
      NSVTOT=NSVTOT+NN1
      IA(M17)=NN1
      N15=NN15+1
      M16=M16+1
      M17=M17+1
490   CONTINUE
500   CONTINUE
      M2VX=0
      M2VY=0
C     STARTING LOCATIONS FOR TWO-VARIABLE FUNCTION SPACE INFORMATION.
      N2VR=NSVR+NSVTOT
      N2VI=NSVI+NSOBJ+2*NSV
      IF (N2VAR.LE.0) GO TO 540
C     -----------------------------------------------------------------
C                   TWO-VARIABLE FUNCTION SPACE INFORMATION
C     -----------------------------------------------------------------
C --- DATA BLOCK R.
C     VARIABLE NUMBERS AND NUMBER OF VALUES OF X AND Y.
      READ (ISCR1,1200) N2VX,M2VX,N2VY,M2VY,IP2VAR
      N20=N2VR
      M20=N2VI
      MM20=M20+N2VAR-1
```

```
      IF (MM20.LE.NDIA) GO TO 510                                       COPES835
      WRITE (6,810)                                                     COPES836
      WRITE (6,850)                                                     COPES837
      LOCI(25)=MM20                                                     COPES838
      GO TO 550                                                         COPES839
  510 CONTINUE                                                          COPES840
C --- DATA BLOCK S.                                                     COPES841
C     GLOBAL VARIABLE NUMBERS CORRESPONDING TO FUNCTIONS OF X AND Y.    COPES842
      READ (ISCR1,1200) (IA(I),I=M20,MM20)                              COPES843
      IF (IPNPUT.LT.2) WRITE (6,1170) IP2VAR                            COPES844
      IF (IPNPUT.LT.2) WRITE (6,1180) (IA(I),I=M20,MM20)                COPES845
C --- DATA BLOCK T.                                                     COPES846
C     VALUES OF X COMPONENTS.                                           COPES847
      NN20=N20+M2VX-1                                                   COPES848
      IF (NN20.LE.NDRA) GO TO 520                                       COPES849
      WRITE (6,780)                                                     COPES850
      WRITE (6,740)                                                     COPES851
      LOCR(25)=NN20                                                     COPES852
      GO TO 550                                                         COPES853
  520 READ (ISCR1,1210) (RA(I),I=N20,NN20)                              COPES854
      IF (IPNPUT.LT.2) WRITE (6,1140) N2VX                              COPES855
      IF (IPNPUT.LT.2) WRITE (6,1160) (RA(I),I=N20,NN20)                COPES856
C --- DATA BLOCK U.                                                     COPES857
C     VALUES OF Y COMPONENTS.                                           COPES858
      N21=N20+M2VX                                                      COPES859
      NN21=N21+M2VY-1                                                   COPES860
      IF (NN21.LE.NDRA) GO TO 530                                       COPES861
      WRITE (6,780)                                                     COPES862
      WRITE (6,750)                                                     COPES863
      LOCR(25)=NN21                                                     COPES864
      GO TO 550                                                         COPES865
  530 CONTINUE                                                          COPES866
      NN20=NN21                                                         COPES867
      READ (ISCR1,1210) (RA(I),I=N21,NN21)                              COPES868
      IF (IPNPUT.LT.2) WRITE (6,1150) N2VY                              COPES869
      IF (IPNPUT.LT.2) WRITE (6,1160) (RA(I),I=N21,NN21)                COPES870
  540 CONTINUE                                                          COPES871
C     ------------------------------------------------------------------COPES872
C                    DYNAMIC STORAGE ALLOCATION                         COPES873
C     ------------------------------------------------------------------COPES874
      NDV2=NDV+2                                                        COPES875
C     REAL VARIABLES.                                                   COPES876
C     X.                                                                COPES877
      LOCR(1)=1                                                         COPES878
C     VLB.                                                              COPES879
      LOCR(2)=NDV+3                                                     COPES880
C     VUB.                                                              COPES881
      LOCR(3)=LOCR(2)+NDV2                                              COPES882
C     SCAL.                                                             COPES883
      LOCR(4)=LOCR(3)+NDV2                                              COPES884
C     AMULT.                                                            COPES885
      LOCR(5)=LOCR(4)+NDV2                                              COPES886
C     BLU.                                                              COPES887
      LOCR(6)=LOCR(5)+NDVTOT                                            COPES888
C     DELX.                                                             COPES889
      LOCR(7)=LOCR(6)+4*NCONA                                           COPES890
      LOCR(8)=LOCR(7)+NDV                                               COPES891
C     SENS.                                                             COPES892
      LOCR(15)=LOCR(8)                                                  COPES893
C     XM2V.                                                             COPES894
      LOCR(20)=LOCR(15)+NSVTOT                                          COPES895
C     YM2V.                                                             COPES896
      LOCR(21)=LOCR(20)+M2VX                                            COPES897
      LOCR(22)=LOCR(21)+M2VY                                            COPES898
C     START OF EXECUTION STORAGE.                                       COPES899
      LOCR(23)=LOCR(22)                                                 COPES900
C     INTEGER VARIABLES.                                                COPES901
C     IDSGN.                                                            COPES902
      LOCI(1)=1                                                         COPES903
C     NDSGN.                                                            COPES904
      LOCI(2)=NDVTOT+1                                                  COPES905
C     ICON.                                                             COPES906
      LOCI(3)=LOCI(2)+NDVTOT                                            COPES907
C     ISC.                                                              COPES908
      LOCI(4)=LOCI(3)+NCONA                                             COPES909
C     LOCX.                                                             COPES910
      LOCI(5)=LOCI(4)+2*(NDV+NCONA)                                     COPES911
C     LOCF.                                                             COPES912
      LOCI(6)=LOCI(5)+NXAPRX                                            COPES913
      LOCI(7)=LOCI(6)+NF                                                COPES914
C     NSENSZ.                                                           COPES915
      LOCI(15)=LOCI(7)                                                  COPES916
C     ISENS.                                                            COPES917
      LOCI(16)=LOCI(15)+NSOBJ                                           COPES918
C     NSENS.                                                            COPES919
      LOCI(17)=LOCI(16)+NSV                                             COPES920
      LOCI(18)=LOCI(17)+NSV                                             COPES921
C     N2VZ.                                                             COPES922
      LOCI(20)=LOCI(18)                                                 COPES923
      LOCI(21)=LOCI(20)+N2VAR                                           COPES924
C     START OF EXECUTION STORAGE.                                       COPES925
      LOCI(23)=LOCI(21)                                                 COPES926
C     EXECUTION STORAGE REQUIREMENTS.                                   COPES927
      NRI=NDV                                                           COPES928
      IF (NACMX1.GT.NRI) NRI=NACMX1                                     COPES929
      NR2=3*NCON+12*NDV+NACMX1*(NDV2+NACMX1)+3*NRI+12                   COPES930
      NI2=NACMX1+2*NRI+2*NDV+NCON                                       COPES931
      NR3=NSV                                                           COPES932
      IF (NSOBJ.GT.NR3) NR3=NSOBJ                                       COPES933
      NR4=N2VAR                                                         COPES934
      NR5=NR2+NR3                                                       COPES935
      NI5=NI2                                                           COPES936
      M=NXAPRX*(NXAPRX*(NXAPRX+1))/2                                    COPES937
      IF (MAXTRM.LT.3) M=MAXTRM*NXAPRX                                  COPES938
      NR6=3*NXAPRX+6*NDV+2*NF*M*NF                                      COPES939
      NI6=NCONA+NXAPRX                                                  COPES940
      NRI=(KMAX+1-NPTOT)*(NXAPRX+NF+1)                                  COPES941
      IF (NRI.LT.NR2) NRI=NR2                                           COPES942
      IF (KMAX.LT.0) NRI=NPTOT*(NXAPRX+NF+1)                            COPES943
      NR7=NR6+NRI                                                       COPES944
      NI7=NI6+NI2                                                       COPES945
C     START OF TEMPORARY STORAGE.                                       COPES946
      LOCR(24)=LOCR(23)                                                 COPES947
      LOCI(24)=LOCI(23)                                                 COPES948
      IF (NCALC.EQ.2) LOCR(24)=LOCR(23)+NR2                             COPES949
      IF (NCALC.EQ.3) LOCR(24)=LOCR(23)+NR3                             COPES950
      IF (NCALC.EQ.4) LOCR(24)=LOCR(23)+NR4                             COPES951
      IF (NCALC.EQ.5) LOCR(24)=LOCR(23)+NR5                             COPES952
      IF (NCALC.EQ.6.AND.KMAX.LT.0) LOCR(24)=LOCR(23)+NR6               COPES953
      IF (NCALC.EQ.6.AND.KMAX.GT.0) LOCR(24)=LOCR(23)+NR7               COPES954
```

```
      IF (NCALC.EQ.2) LOCI(24)=LOCI(23)+NI2                             COPES955
      IF (NCALC.EQ.5) LOCI(24)=LOCI(23)+NI5                             COPES956
      IF (NCALC.EQ.6.AND.KMAX.LT.0) LOCI(24)=LOCI(23)+NI6               COPES957
      IF (NCALC.EQ.6.AND.KMAX.GT.0) LOCI(24)=LOCI(23)+NI7               COPES958
C     TOTAL STORAGE REQUIREMENTS.                                       COPES959
      LOCR(25)=LOCR(24)                                                 COPES960
      LOCI(25)=LOCI(24)                                                 COPES961
      IF (NCALC.EQ.5) LOCR(25)=LOCR(25)+4*NDV+8                         COPES962
      IF (NCALC.EQ.5) LOCI(25)=LOCI(25)+2*NDVTOT                        COPES963
      IF (IPNPUT.LT.2) WRITE (6,860) LOCR(23),LOCR(25),NDRA,LOCI(23),LOCCOPES964
     1I(25),NDIA                                                        COPES965
550   CONTINUE                                                          COPES966
      RETURN                                                            COPES967
C     ------------------------------------------------------------------COPES968
C                                FORMATS                                COPES969
C     ------------------------------------------------------------------COPES970
560   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK K)                       COPES971
570   FORMAT (//5X,43HDELTA-X BOUNDS FOR APPROXIMATE OPTIMIZATION)      COPES972
580   FORMAT (80A1)                                                     COPES973
590   FORMAT (I5/80A1)                                                  COPES974
600   FORMAT (///5X,49H* • APPROXIMATE ANALYSIS/OPTIMIZATION INFORMATIONCOPES975
     1//5X,38HNUMBER OF FUNCTIONS APPROXIMATED, NF =,I5/5X,38HNUMBER OF COPES976
     2INPUT X-VECTORS,        NPS =,I5/5X,38HNUMBER OF INPUT X-F PAIRS, COPES977
     3   NPFS =,I5/5X,38HX-VECTOR FROM ANALIZ,             NPA =,I5/5X,3COPES978
     48HNOMINAL DESIGN,                 INOM =,I5/5X,38HREAD UNIT FOR X-COPES979
     5VECTORS,       ISCRX =,I5/5X,38HREAD UNIT FOR X-F PAIRS,      ISCRCOPES980
     6XF =,I5/5X,38HPRINT CONTROL,                   IPAPRX =,I5)       COPES981
610   FORMAT (/5X,38HMINIMUM APPROXIMATING CYCLES,   KMIN =,I5/5X,38HMAXCOPES982
     1IMUM APPROXIMATING CYCLES,   KMAX =,I5/5X,38HMAXIMUM DESIGNS USED COPES983
     2IN FIT,   NPMAX =,I5/5X,38HNOMINAL DESIGN PARAMETER,       JNOM =,COPES984
     3I5/5X,38HX-LOCATION INPUT PARAMETER,   INXLOC =,I5/5X,38HF-LOCATIOCOPES985
     4N INPUT PARAMETER,   INFLOC =,I5/5X,38HTAYLER SERIES I.D. CODE,   COPES986
     5   MAXTRM =,I5)                                                   COPES987
620   FORMAT (/5X,38HMULTIPLIER ON DELX,           XFACT1 =,E12.4/5X,38HCOPES988
     1MULTIPLIER ON DELX,           XFACT2 =,E12.4)                     COPES989
630   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK L)                       COPES990
640   FORMAT (//5X,31HGLOBAL LOCATIONS OF X-VARIABLES)                  COPES991
650   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK M)                       COPES992
660   FORMAT (//5X,29HGLOBAL LOCATIONS OF FUNCTIONS)                    COPES993
670   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK N)                       COPES994
680   FORMAT (//5X,25HX-VECTORS INPUT FROM UNIT,I5)                     COPES995
690   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK O)                       COPES996
700   FORMAT (////5X,25HX-F PAIRS INPUT FROM UNIT,I5)                   COPES997
710   FORMAT (//5X,6HNUMBER,I5,5X,6HDESIGN,I5)                          COPES998
720   FORMAT (/5X,8HX-VECTOR)                                           COPES999
730   FORMAT (/5X,15HFUNCTION VALUES)                                   COPE1000
740   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK T)                       COPE1001
750   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK U)                       COPE1002
760   FORMAT (/5X,58HCONMIN PARAMETERS (IF ZERO, CONMIN DEFAULT WILL OVECOPE1003
     1R-RIDE)//5X,6HIPRINT,2X,5HITMAX,3X,6HICNDIR,3X,5HNSCAL,3X,4HITRM,3COPE1004
     2X,6HLINOBJ,2X,6HNACMX1,3X,4HNFDG/8I8)                             COPE1005
770   FORMAT (/6X,4HFDCH,12X,5HFDCHM,11X,2HCT,14X,5HCTMIN/1X,4(2X,E14.5)COPE1006
     1//6X,3HCTL,13X,6HCTLMIN,10X,5HTHETA,11X,3HPHI/1X,4(2X,E14.5)//6X,6COPE1007
     2HDELFUN,10X,6HDABFUN,10X,6HALPHAX,10X,6HABOBJ1/1X,4(2X,E14.5))    COPE1008
780   FORMAT (//5X,54HREQUIRED STORAGE IN ARRAY RA EXCEEDS AVAILABLE STOCOPE1009
     1RAGE)                                                             COPE1010
790   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK F)                       COPE1011
800   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK G)                       COPE1012
810   FORMAT (//5X,54HREQUIRED STORAGE IN ARRAY IA EXCEEDS AVAILABLE STOCOPE1013
     1RAGE)                                                             COPE1014
820   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK I)                       COPE1015
830   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK P)                       COPE1016
840   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK Q)                       COPE1017
850   FORMAT (/5X,27HUNABLE TO READ DATA BLOCK R)                       COPE1018
860   FORMAT (////5X,39H* • ESTIMATED DATA STORAGE REQUIREMENTS//15X,4HRCOPE1019
     1EAL,26X,7HINTEGER/5X,27HINPUT  EXECUTION  AVAILABLE,5X,27HINPUT  ECOPE1020
     2XECUTION  AVAILABLE/I9,2I10,2X,3I10)                              COPE1021
870   FORMAT (1H1,4X,27HCARD IMAGES OF CONTROL DATA///5X,4HCARD,20X,5HIMCOPE1022
     1AGE)                                                              COPE1023
880   FORMAT (1H0)                                                      COPE1024
890   FORMAT (I8,1H),2X,80A1)                                           COPE1025
900   FORMAT (/5X,40HTOTAL NUMBER OF CONSTRAINED PARAMETERS =,I5)       COPE1026
910   FORMAT (//5X,26HCALCULATION CONTROL, NCALC/5X,5HVALUE,3X,7HMEANINGCOPE1027
     1/7X,1H1,5X,15HSINGLE ANALYSIS/7X,1H2,5X,12HOPTIMIZATION/7X,1H3,5X,COPE1028
     211HSENSITIVITY/7X,1H4,5X,27HTWO-VARIABLE FUNCTION SPACE/7X,1H5,5X,COPE1029
     319HOPTIMUM SENSITIVITY/7X,1H6,5X,24HAPPROXIMATE OPTIMIZATION)     COPE1030
920   FORMAT (2I10,F10.2)                                               COPE1031
930   FORMAT (//5X,16HDESIGN VARIABLES/11X,5HD. V.,5X,6HGLOBAL,4X,11HMULCOPE1032
     1TIPLYING/5X,2HID,5X,3HNO.,5X,8HVAR. NO.,5X,6HFACTOR)              COPE1033
940   FORMAT (2I7,5X,I5,6X,E12.5)                                       COPE1034
950   FORMAT (5X,16I5)                                                  COPE1035
960   FOPMAT (/5X,34HPRINT CONTROL,                  IPSENS =,I5/5X,34HNUMBER COPE1036
     1OF SENSITIVITY OBJECTIVES =,I5//5X,53HGLOBAL NUMBERS ASSOCIATED WICOPE1037
     2TH SENSITIVITY OBJECTIVES)                                        COPE1038
970   FORMAT (1H1,//////,5X,47HCCCCCCC   0000000   PPPPPPP   EEEEEEE   SCOPE1039
     1SSSSSS/5X,47HC          O     O   P     P   E          S     /5X,47COPE1040
     2HC        O     O   P     P   E          S     /5X,47HC        OCOPE1041
     3     O   PPPPPPP   EEEE      SSSSSSS/5X,47HC        O     O   P COPE1042
     4        E                S/5X,47HC        O     O   P          E   COPE1043
     5             S/5X,47HCCCCCCC   0000000   P          EEEEEEE   SSSSSSSCOPE1044
     6)                                                                 COPE1045
980   FORMAT (/////14X,29HC O N T R O L   P R O G R A M//26X,5HF O R//8XCOPE1046
     1,41HE N G I N E E R I N G   S Y N T H E S I S)                    COPE1047
990   FORMAT (/////24X,9HT I T L E//5X,20A4)                            COPE1048
1000  FORMAT (1H1,4X,6HTITLE:/5X,20A4)                                  COPE1049
1010  FORMAT (///5X,19HCONTROL PARAMETERS;/5X,42HCALCULATION CONTROL,   COPE1050
     1                   NCALC =,I5/5X,42HNUMBER OF GLOBAL DESIGN VARIABLES,   COPE1051
     2NDV =,I5/5X,42HNUMBER OF SENSITIVITY VARIABLES,      NSV =,I5/5X,42COPE1052
     3HNUMBER OF FUNCTIONS IN TWO-SPACE,  N2VAR =,I5/5X,42HNUMBER OF APPCOPE1053
     4ROXIMATING VAR.      NXAPRX =,I5/5X,42HINPUT INFORMATION PRINT COOCOPE1054
     5E,     IPNPUT =,I5/5X,42HDEBUG PRINT CODE,                   IPDBG COPE1055
     6=,I5)                                                             COPE1056
1020  FORMAT (////5X,27H* * SENSITIVITY INFORMATION)                    COPE1057
1030  FORMAT (/14X,6HGLOBAL,4X,7HNOMINAL/5X,6HNUMBER,2X,8HVARIABLE,4X,5HCOPE1058
     1VALUE,6X,18HOFF-NOMINAL VALUES)                                   COPE1059
1040  FORMAT (5X,I4,I8,5X,E12.5,1X,5E11.4)                              COPE1060
1050  FORMAT (35X,5E11.4)                                               COPE1061
1060  FORMAT (4F10.2,10A4)                                              COPE1062
1070  FORMAT (////5X,28H* * OPTIMIZATION INFORMATION//5X,35HGLOBAL VARIACOPE1063
     1BLE NUMBER OF OBJECTIVE,10X,1H=,I5/5X,46HMULTIPLIER (NEGATIVE INDICOPE1064
     2CATES MINIMIZATION) =,E12.4)                                      COPE1065
1080  FORMAT (/5X,27HDESIGN VARIABLE INFORMATION/5X,50HNON-ZERO INITIAL COPE1066
     1VALUE WILL OVER-RIDE MODULE INPUT/5X,5HD. V.,5X,5HLOWER,10X,5HUPPECOPE1067
     2R,9X,7HINITIAL/5X,3HNO.,7X,5HBOUND,10X,5HBOUND,10X,5HVALUE,10X,5HSCOPE1068
     3CALE)                                                             COPE1069
1090  FORMAT (I8,4X,E12.5,3X,E12.5,3X,E12.5,3X,E12.5,5A4)               COPE1070
1100  FORMAT (I8,I7,2I8,5X,E12.5,3X,E12.5,3X,E12.5,3X,E12.5)            COPE1071
1110  FORMAT (//5X,22HCONSTRAINT INFORMATION)                           COPE1072
1120  FORMAT (/5X,9HTHERE ARE,I3,16H CONSTRAINT SETS)                   COPE1073
1130  FORMAT (11X,6HGLOBAL,2X,6HGLOBAL,2X,6HLINEAR,6X,5HLOWER,6X,13HNORMCOPE1074
```

```
     1ALIZATION,6X,5HUPPER,6X,13HNORMALIZATION/6X,2HID,3X,6HVAR. 1,2X,6HCOPE1075
     2VAR. 2,4X,2HID,8X,5HBOUND,9X,6HFACTOR,10X,5HBOUND,9X,6HFACTOR)   COPE1076
 1140 FORMAT (//5X,49HGLOBAL VARIABLE NUMBER CORRESPONDING TO X, N2VX =,COPE1077
     1I5//5X,20HVALUES OF X-VARIABLE)                                  COPE1078
 1150 FORMAT (//5X,49HGLOBAL VARIABLE NUMBER CORRESPONDING TO Y, N2VY =,COPE1079
     1I5//5X,20HVALUES OF Y-VARIABLE)                                  COPE1080
 1160 FORMAT (3X,5E12.4)                                               COPE1081
 1170 FORMAT (////5X,51H* * TWO-VARIABLE FUNCTION SPACE MAPPING INFORMATCOPE1082
     1ION//5X,23HPRINT CONTROL, IP2VAR =,I5//5X,52HGLOBAL VARIABLE NUMBECOPE1083
     2RS ASSOCIATED WITH F(X,Y), N2VZ)                                 COPE1084
 1180 FORMAT (5X,10I5)                                                 COPE1085
 1190 FORMAT (20A4)                                                    COPE1086
 1200 FORMAT (8I10)                                                    COPE1087
 1210 FORMAT (8F10.2)                                                  COPE1088
 1220 FORMAT (///5X,26H* * INPUT ERROR, NCALC =,I2,2X,41HIS LT.0 OR GTCOPE1089
     1.6  PROGRAM TERMINATED * * *)                                    COPE1090
      END                                                              COPE1091
      SUBROUTINE COPE02 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)                 COPE1092
      COMMON /CNMN1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAXCOPE1093
     1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,ITCOPE1094
     2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER                              COPE1095
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,MCOPE1096
     12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCRCOPE1097
     21,ISCR2,NXAPRX,NPS,NPFS,NPA,NF,INOM,IPAPRX,KMIN,KMAX,XFACT1,XFACT2COPE1098
     3,NAN2,NAN3,NPMAX,NPTOT,JNOM,MAXTRM                               COPE1099
      DIMENSION ARRAY(NARRAY), RA(NDRA), IA(NDIA)                      COPE1100
C     *****************************************************************COPE1101
C                    ROUTINE TO CONTROL OPTIMIZATION.                  COPE1102
C     *****************************************************************COPE1103
C     BY G. N. VANDERPLAATS                        MAR., 1973.         COPE1104
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.                 COPE1105
C     -----------------------------------------------------------------COPE1106
C                          ARRAY DIMENSIONS                            COPE1107
C     -----------------------------------------------------------------COPE1108
      NN1=NDV+2                                                        COPE1109
      NN2=2*NDV+NCON                                                   COPE1110
      NN3=NACMX1                                                       COPE1111
      NN4=NN3                                                          COPE1112
      IF (NDV.GT.NN4) NN4=NDV                                          COPE1113
      NN5=2*NN4                                                        COPE1114
      NN6=NDVTOT                                                       COPE1115
      NN7=NCONA                                                        COPE1116
C     -----------------------------------------------------------------COPE1117
C                       ARRAY STARTING LOCATIONS                       COPE1118
C     -----------------------------------------------------------------COPE1119
C     X, VLB, VUB, DF, A, S, G1, G2, C, B, SCAL, ISC, IC, MS1          COPE1120
      NX=1                                                             COPE1121
      NVLB=LOCR(2)                                                     COPE1122
      NVUB=LOCR(3)                                                     COPE1123
      NNSCAL=LOCR(4)                                                   COPE1124
      NDF=LOCR(23)                                                     COPE1125
      NG=NDF+NN1                                                       COPE1126
      NA=NG+NN2                                                        COPE1127
      NS=NA+NN1*NN3                                                    COPE1128
      NG1=NS+NN1                                                       COPE1129
      NG2=NG1+NN2                                                      COPE1130
      NC=NG2+NN2                                                       COPE1131
      NB=NC+NN4                                                        COPE1132
      NISC=LOCI(4)                                                     COPE1133
      NIC=LOCI(23)                                                     COPE1134
      NMS1=NIC+NN3                                                     COPE1135
C     AMULT, BLU, IDSGN, NSDGN, ICON                                   COPE1136
      NAMULT=LOCR(5)                                                   COPE1137
      NBLU=LOCR(6)                                                     COPE1138
      NIDSGN=1                                                         COPE1139
      NNDSGN=LOCI(2)                                                   COPE1140
      NICON=LOCI(3)                                                    COPE1141
C     -----------------------------------------------------------------COPE1142
C                             OPTIMIZATION                             COPE1143
C     -----------------------------------------------------------------COPE1144
      IGOTO=0                                                          COPE1145
C     CALL CONMIN (X,VLB,VUB,G,SCAL,DF,A,S,G1,G2,B,C,ISC,IC,MS1,N1,N2,N3COPE1146
C    *,N4,N5)                                                          COPE1147
   10 CONTINUE                                                         COPE1148
      CALL CONMIN (RA(NX),RA(NVLB),RA(NVUB),RA(NG),RA(NNSCAL),RA(NDF),RACOPE1149
     1(NA),RA(NS),RA(NG1),RA(NG2),RA(NB),RA(NC),IA(NISC),IA(NIC),IA(NMS1COPE1150
     2),NN1,NN2,NN3,NN4,NN5)                                           COPE1151
C     ANALIZE.                                                         COPE1152
      CALL COPE03 (ARRAY,NARRAY,RA(NX),RA(NG),RA(NAMULT),RA(NBLU),IA(NIDCOPE1153
     1SGN),IA(NNDSGN),IA(NICON),NN1,NN2,NN6,NN7,ITER,OBJ)              COPE1154
      IF (IGOTO.GT.0) GO TO 10                                         COPE1155
      RETURN                                                           COPE1156
      END                                                              COPE1157
      SUBROUTINE COPE03 (ARRAY,NARRAY,X,G,AMULT,BLU,IDSGN,NDSGN,ICON,NN1COPE1158
     1,NN2,NN6,NN7,ITER,OBJ)                                           COPE1159
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,MCOPE1160
     12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCRCOPE1161
     21,ISCR2,NXAPRX,NPS,NPFS,NPA,NF,INOM,IPAPRX,KMIN,KMAX,XFACT1,XFACT2COPE1162
     3,NAN2,NAN3,NPMAX,NPTOT,JNOM,MAXTRM                               COPE1163
      DIMENSION ARRAY(NARRAY), X(NN1), G(NN2), AMULT(NN6), BLU(4,NN7)  COPE1164
      DIMENSION IDSGN(NN6), NDSGN(NN6), ICON(NN7)                      COPE1165
C     *****************************************************************COPE1166
C     BUFFER BETWEEN CONMIN AND COPES FUNCTION EVALUATION.             COPE1167
C     *****************************************************************COPE1168
C     BY G. N. VANDERPLAATS                        MAR., 1973.         COPE1169
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.                 COPE1170
C     INITIAL ANALYSIS HAS BEEN DONE.  IF ITER = 0, GO EVALUATE        COPE1171
C     OBJECTIVE AND CONSTRAINTS.                                       COPE1172
      IF (ITER.EQ.0) ITER1=0                                           COPE1173
      IF (ITER.LT.1) GO TO 40                                          COPE1174
C     -----------------------------------------------------------------COPE1175
C               PRINT OUTPUT IF DEBUG CONTROL IS TURNED ON             COPE1176
C     -----------------------------------------------------------------COPE1177
C     DEBUG OUTPUT AS REQUIRED.                                        COPE1178
      IF (IPDBG.LT.1) GO TO 20                                         COPE1179
      IF (ITER.EQ.ITER1.OR.ITER.LE.1) GO TO 20                         COPE1180
      XSAV2=X(1)                                                       COPE1181
      X(1)=XSAV1                                                       COPE1182
      N5=LOCR(5)                                                       COPE1183
      M2=LOCI(2)                                                       COPE1184
      DO 10 I=1,NDVTOT                                                 COPE1185
      N=NDSGN(I)                                                       COPE1186
      M=IDSGN(I)                                                       COPE1187
      IF (N.GT.0) ARRAY(M)=X(N)*AMULT(I)                               COPE1188
   10 CONTINUE                                                         COPE1189
      ICALC=3                                                          COPE1190
      NAN3=NAN3+1                                                      COPE1191
      CALL ANALIZ (ICALC)                                              COPE1192
      WRITE (6,70)                                                     COPE1193
      ITER1=ITER                                                       COPE1194
```

```
      X(1)=XSAV2
   20 CONTINUE
C     ------------------------------------------------------------------
C             TRANSFER DESIGN VARIABLES TO USER APRAY
C     ------------------------------------------------------------------
      N5=LOCR(5)
      M2=LOCI(2)
      DO 30 I=1,NDVTOT
      N=NDSGN(I)
      M=IDSGN(I)
      IF (N.GT.0) ARRAY(M)=X(N)*AMULT(I)
   30 CONTINUE
C     ------------------------------------------------------------------
C                                  ANALIZE
C     ------------------------------------------------------------------
      ICALC=2
      NAN2=NAN2+1
      CALL ANALIZ (ICALC)
C     SAVE X(1).
      XSAV1=X(1)
C     ------------------------------------------------------------------
C                                 OBJECTIVE
C     ------------------------------------------------------------------
   40 CONTINUE
      OBJ=-SGNOPT*ARRAY(IOBJ)
      IF (NCONA.EQ.0) RETURN
C     ------------------------------------------------------------------
C                             CONSTRAINT VALUES
C     ------------------------------------------------------------------
      J=1
      N=0
      DO 60 I=1,NCONA
C     PARAMETER IDENTIFIER.
      NN=ICON(I)
      CC=ARRAY(NN)
C     LOWER BOUND.
      BB=BLU(1,I)
      IF (BB.LT.-1.0E+15) GO TO 50
C     NORMALIZATION FACTOR.
      C1=BLU(2,I)
C     CONSTRAINT VALUE.
      N=N+1
      G(N)=(BB-CC)/C1
C     UPPER BOUND.
   50 BB=BLU(3,I)
C     NORMALIZATION FACTOR.
      C1=BLU(4,I)
      J=J+4
      IF (BB.GT.1.0E+15) GO TO 60
C     CONSTRAINT VALUE.
      N=N+1
      G(N)=(CC-BB)/C1
   60 CONTINUE
      RETURN
C
C     ------------------------------------------------------------------
C                                  FORMATS
C     ------------------------------------------------------------------
   70 FORMAT (1H1)
      END
```

```
      SUBROUTINE COPE04 (ARRAY,NARRAY,SENS,NSENSZ,ISENS,NSENS,TEMP,NN8,N
     1N9,NN10,RA,IA,NDRA,NDIA)
      COMMON /COPES1/ TITLE(20)
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,M
     12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCR
     21,ISCR2,NXAPRX,NPS,NPFS,NPA,NF,INOM,IPAPRX,KMIN,KMAX,XFACT1,XFACT2
     3,NAN2,NAN3,NPMAX,NPTOT,JNOM,MAXTRM
      COMMON /CNMN1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAX
     1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,IT
     2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER
      DIMENSION ARRAY(NARRAY), SENS(NN8), NSENSZ(NN9), ISENS(NN10), NSEN
     1S(NN10), TEMP(1), RA(NDRA), IA(NDIA)
C     ******************************************************************
C     ROUTINE TO PROVIDE SENSITIVITY INFORMATION WITH RESPECT TO
C     A PRESCRIBED SET OF DESIGN VARIABLES.
C     ******************************************************************
C     BY G. N. VANDERPLAATS                           MAR., 1973.
C     STORE OUTPUT ON UNIT ISCR1.
      REWIND ISCR1
      IF (IPDBG.LT.1) IPRINT=0
C     ------------------------------------------------------------------
C                 WRITE BASIC INFORMATION ON UNIT ISCR1
C     ------------------------------------------------------------------
C     TITLE.
      WRITE (ISCR1,330) (TITLE(I),I=1,20)
C     NCALC, NSV, NSOBJ
      WRITE (ISCR1,340) NCALC,NSV,NSOBJ
C     ISENS(I),I=1,NSV.
      WRITE (ISCR1,340) (ISENS(I),I=1,NSV)
C     NSENSZ(I),I=1,NSOBJ.
      WRITE (ISCR1,340) (NSENSZ(I),I=1,NSOBJ)
      JCALC=3
      ICALC=2
      NDVSAV=NDV
C     ------------------------------------------------------------------
C     **********                   NOMINAL                    **********
C     ------------------------------------------------------------------
      IF (NCALC.EQ.5) GO TO 10
C     STANDARD SENSITIVITY.
      NAN2=NAN2+1
      CALL ANALIZ (ICALC)
      IF (IPSENS.GT.0) NAN3=NAN3+1
      IF (IPSENS.GT.0) CALL ANALIZ (JCALC)
      GO TO 130
   10 CONTINUE
C     OPTIMUM SENSITIVITY.
C     SAVE X, VLB, VUB AND SCAL IN TEMPORARY STORAGE.
      N=4*NDV+8
      L=LOCR(24)
      DO 20 I=1,N
      RA(L)=RA(I)
   20 L=L+1
C     SAVE IDSGN AND NDSGN IN TEMPORARY STORAGE.
      N=2*NDVTOT
      L=LOCI(24)
      DO 30 I=1,N
      IA(L)=IA(I)
   30 L=L+1
C     SHIFT DESIGN VARIABLE INFORMATION IF ANY SENSITIVITY VARIABLE IS
C     ALSO A DESIGN VARIABLE.
```

```
      NDV2=NDV+2
      DO 90 I=1,NSV
C     GLOBAL SENSITIVITY VARIABLE LOCATION.
      N=ISENS(I)
C     IS THIS ALSO A DESIGN VARIABLE.
      M2=LOCI(2)
      DO 40 J=1,NDVTOT
      L=IA(J)
      IF (L.EQ.N) GO TO 50
   40 M2=M2+1
C     SENSITIVITY VARIABLE IS NOT A DESIGN VARIABLE.
      GO TO 90
   50 CONTINUE
C     SENSITIVITY VARIABLE IS ALSO A DESIGN VARIABLE.
      NDV=NDV-1
      IDV=IA(M2)
C     ELIMINATE THIS DESIGN VARIABLE AND REDUCE HIGHER NUMBER DESIGN
C     VARIABLES BY ONE.
      M2=LOCI(2)
      DO 70 J=1,NDVTOT
      IF (IA(M2).NE.IDV) GO TO 60
      IA(M2)=0
C     SET DESIGN VARIABLE VALUES TO SENSITIVITY VARIABLE VALUE.
      K=IA(J)
      M5=LOCR(5)+J-1
      ARRAY(K)=ARRAY(N)*RA(M5)
   60 CONTINUE
      IF (IA(M2).GT.IDV) IA(M2)=IA(M2)-1
   70 M2=M2+1
      IF (IDV.EQ.NDV) GO TO 90
C     SHIFT X, VLB, VUB AND SCAL.
      DO 80 J=IDV,NDV
C     X.
      RA(J)=RA(J+1)
C     VLB.
      K=J+NDV2
      RA(K)=RA(K+1)
C     VUB.
      K=K+NDV2
      RA(K)=RA(K+1)
C     SCAL.
      K=K+NDV2
   80 RA(K)=RA(K+1)
   90 CONTINUE
      NAN2=NAN2+1
      CALL ANALIZ (ICALC)
      IF (NDV.LE.0) GO TO 100
      CALL COPE02 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)
  100 CONTINUE
      IF (IPSENS.GT.0) NAN3=NAN3+1
      IF (IPSENS.GT.0) CALL ANALIZ (JCALC)
C     PUT X, VLB, VUB AND SCAL BACK.
      L=LOCR(24)
      N=4*NDVSAV+8
      DO 110 I=1,N
      RA(I)=RA(L)
  110 L=L+1
C     PUT IDSGN AND NDSGN BACK.
      N=2*NDVTOT
      L=LOCI(24)
      DO 120 I=1,N
      IA(I)=IA(L)
  120 L=L+1
  130 CONTINUE
C     ------------------------------------------------------------------
C                   WRITE NOMINAL RESULTS ON UNIT ISCR1
C     ------------------------------------------------------------------
C     SENS(I,1).
      M=1
      DO 140 I=1,NSV
      TEMP(I)=SENS(M)
  140 M=M+NSENS(I)
      WRITE (ISCR1,350) (TEMP(I),I=1,NSV)
C     SENSITIVITY OBJECTIVES, OBJZ.
      DO 150 I=1,NSOBJ
      M=NSENSZ(I)
  150 TEMP(I)=ARRAY(M)
      WRITE (ISCR1,350) (TEMP(I),I=1,NSOBJ)
C     ------------------------------------------------------------------
C     **********              SENSITIVITIES                   **********
C     ------------------------------------------------------------------
      NSVAL1=0
      DO 320 II=1,NSV
C     GLOBAL LOCATION OF SENSITIVITY VARIABLE.
      ISENSV=ISENS(II)
C     NUMBER OF SENSITIVITY VARIABLES, NSENSV.
      NSENSV=NSENS(II)
C     WRITE ISENSV AND NSENSV-1 ON UNIT ISCR1.
      NSENSI=NSENSV-1
      WRITE (ISCR1,340) ISENSV,NSENSI
      IF (NSENSV.LE.1) GO TO 320
      ID1=0
      IF (NCALC.NE.5) GO TO 210
C     IS THIS SENSITIVITY VARIABLE ALSO A DESIGN VARIABLE.
      NDV=NDVSAV
      DO 160 I=1,NDVTOT
      JJ=I
      IF (IA(I).EQ.ISENSV) GO TO 170
  160 CONTINUE
C     ISENSV IS NOT A DESIGN VARIABLE.
      GO TO 210
  170 CONTINUE
C     ISENSV IS A DESIGN VARIABLE.  MODIFY OPTIMIZATION INFORMATION.
      NDV2=NDV+2
      NDV=NDV-1
C     SAVE X, VLB, VUB AND SCAL FOR THIS DESIGN VARIABLE AND SHIFT
C     REMAINING VARIABLES.
C     SAVE.
      M2=LOCI(2)+JJ-1
      ID1=IA(M2)
      SAVX=RA(ID1)
      K=ID1+NDV2
      SAVL=RA(K)
      K=K+NDV2
      SAVU=RA(K)
      K=K+NDV2
      SAVS=RA(K)
C     SHIFT
      IF (ID1.GT.NDV) GO TO 190
      DO 180 I=ID1,NDV
```

```
      RA(I)=RA(I+1)
      K=I+NDV2
      RA(K)=RA(K+1)
      K=K+NDV2
      RA(K)=RA(K+1)
      K=K+NDV2
  180 RA(K)=RA(K+1)
  190 CONTINUE
C     MODIFY NDSGN.
      M2=LOCI(2)
      DO 200 I=1,NDVTOT
      IF (IA(M2).EQ.ID1) IA(M2)=0
      IF (IA(M2).GT.ID1) IA(M2)=IA(M2)-1
  200 M2=M2+1
  210 CONTINUE
C     ------------------------------------------------------------------
C              VARY THE VALUE OF THE SENSITIVITY PARAMETER
C     ------------------------------------------------------------------
      NSVAL1=NSVAL1+1
      NSVALN=NSVAL1
      DO 280 JJ=2,NSENSV
      NSVAL1=NSVAL1+1
      ARRAY(ISENSV)=SENS(NSVAL1)
C     WRITE SENS(I,J) ON UNIT ISCR1.
      WRITE (ISCR1,350) SENS(NSVAL1)
C     ANALIZE.
      IF (NCALC.EQ.5) GO TO 220
C     STANDARD SENSITIVITY.
      NAN2=NAN2+1
      CALL ANALIZ (ICALC)
      IF (IPSENS.GT.0) NAN3=NAN3+1
      IF (IPSENS.GT.0) CALL ANALIZ (JCALC)
      GO TO 260
  220 CONTINUE
C     OPTIMUM SENSITIVITY.
      IF (NDV.EQ.NDVSAV) GO TO 240
C     SET LINKED DESIGN VARIABLE VALUES TO PRESCRIBED VALUE.
      M2=LOCI(2)
      DO 230 I=1,NDVTOT
      IF (IA(M2).NE.0) GO TO 230
      L=IA(I)
      M5=LOCR(5)+I-1
      ARRAY(L)=ARRAY(ISENSV)*RA(M5)
  230 M2=M2+1
  240 CONTINUE
      NAN2=NAN2+1
      CALL ANALIZ (ICALC)
      IF (NDV.LE.0) GO TO 250
      CALL COPE02 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)
  250 CONTINUE
      IF (IPSENS.GT.0) NAN3=NAN3+1
      IF (IPSENS.GT.0) CALL ANALIZ (JCALC)
  260 CONTINUE
C     ------------------------------------------------------------------
C              WRITE SENSITIVITY RESULTS ON UNIT ISCR1
C     ------------------------------------------------------------------
C     OBJZ.
      DO 270 I=1,NSOBJ
      M=NSENSZ(I)
  270 TEMP(I)=ARRAY(M)
      WRITE (ISCR1,350) (TEMP(I),I=1,NSOBJ)
  280 CONTINUE
      ARRAY(ISENSV)=SENS(NSVALN)
      IF (NCALC.NE.5.OR.ID1.EQ.0) GO TO 320
C     RESTORE X, VLB, VUB AND SCAL.
      NDV=NDVSAV
      IF (ID1.EQ.NDV) GO TO 300
      L=NDV-1
      L1=L
      DO 290 I=ID1,L1
      RA(L+1)=RA(L)
      K=L+NDV2
      RA(K+1)=RA(K)
      K=K+NDV2
      RA(K+1)=RA(K)
      K=K+NDV2
      RA(K+1)=RA(K)
  290 L=L-1
      RA(ID1)=SAVX
      K=ID1+NDV2
      RA(K)=SAVL
      K=K+NDV2
      RA(K)=SAVU
      K=K+NDV2
      RA(K)=SAVS
  300 CONTINUE
C     RESTORE NDSGN.
      M2=LOCI(2)
      DO 310 I=1,NDVTOT
      IF (IA(M2).GE.ID1) IA(M2)=IA(M2)+1
      IF (IA(M2).EQ.0) IA(M2)=ID1
  310 M2=M2+1
  320 CONTINUE
      RETURN
C     ------------------------------------------------------------------
C                                  FORMATS
C     ------------------------------------------------------------------
  330 FORMAT (20A4)
  340 FORMAT (16I5)
  350 FORMAT (5E15.8)
      END
      SUBROUTINE COPE05 (RA,IA,NDRA,NDIA,ISCR1)
      DIMENSION RA(NDRA), IA(NDIA)
C     ******************************************************************
C     ROUTINE TO PRINT SENSITIVITY INFORMATION STORED ON UNIT ISCR1.
C     ******************************************************************
C     BY G. N. VANDERPLAATS                          JULY, 1974.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
      REWIND ISCR1
C     ------------------------------------------------------------------
C                          GENERAL INFORMATION
C     ------------------------------------------------------------------
C     TITLE.
      READ (ISCR1,70) (RA(I),I=1,20)
C     NCALC, NSV, NSOBJ
      READ (ISCR1,80) NCALC,NSV,NSOBJ
      IF (NCALC.NE.3.AND.NCALC.NE.5) RETURN
      IF (NCALC.EQ.3) WRITE (6,90)
      IF (NCALC.EQ.5) WRITE (6,50)
      WRITE (6,60) (RA(I),I=1,20)
```

```
      WRITE (6,100) NSV,NSOBJ                                          COPE1555
C     ISENS(I),I=1,NSV.                                                COPE1556
      READ (ISCR1,80) (IA(I),I=1,NSV)                                  COPE1557
      WRITE (6,110)                                                    COPE1558
      WRITE (6,120) (IA(I),I=1,NSV)                                    COPE1559
C     NSENSZ(I),I=1,NSOBJ.                                             COPE1560
      READ (ISCR1,80) (IA(I),I=1,NSOBJ)                                COPE1561
      WRITE (6,130)                                                    COPE1562
      WRITE (6,120) (IA(I),I=1,NSOBJ)                                  COPE1563
C     ---------------------------------------------------------------- COPE1564
C                         NOMINAL INFORMATION                          COPE1565
C     ---------------------------------------------------------------- COPE1566
C     SENS(I),I=1,NSV.                                                 COPE1567
      READ (ISCR1,140) (RA(I),I=1,NSV)                                 COPE1568
      WRITE (6,150)                                                    COPE1569
      WRITE (6,160) (RA(I),I=1,NSV)                                    COPE1570
C     OBJZ(I),I=1,NSOBJ.                                               COPE1571
      READ (ISCR1,140) (RA(I),I=1,NSOBJ)                               COPE1572
      WRITE (6,170)                                                    COPE1573
      WRITE (6,160) (RA(I),I=1,NSOBJ)                                  COPE1574
C     -----------------------------------------------------------------COPE1575
C     **********        SENSITIVITY INFORMATION              **********COPE1576
C     -----------------------------------------------------------------COPE1577
      WRITE (6,180)                                                    COPE1578
      DO 40 ISENS=1,NSV                                                COPE1579
C     ISENSI, NSENSI                                                   COPE1580
      READ (ISCR1,80) ISENSI,NSENSI                                    COPE1581
      WRITE (6,190) ISENSI                                             COPE1582
      IF (NSENSI.EQ.0) WRITE (6,200)                                   COPE1583
      IF (NSENSI.EQ.0) GO TO 40                                        COPE1584
      DO 30 JJ=1,NSENSI                                                COPE1585
C     SENS(I,J).                                                       COPE1586
      READ (ISCR1,140) SENSIJ                                          COPE1587
C     OBJZ(I),I=1,NSOBJ.                                               COPE1588
      READ (ISCR1,140) (RA(I),I=1,NSOBJ)                               COPE1589
      N=MIN0(4,NSOBJ)                                                  COPE1590
      WRITE (6,210) SENSIJ,(RA(I),I=1,N)                               COPE1591
      N=(NSOBJ-1)/4                                                    COPE1592
      IF (N.LT.1) GO TO 20                                             COPE1593
      L1=5                                                             COPE1594
      DO 10 I=1,N                                                      COPE1595
      L2=L1+3                                                          COPE1596
      L2=MIN0(L2,NSOBJ)                                                COPE1597
      WRITE (6,220) (RA(J),J=L1,L2)                                    COPE1598
10    L1=L1+4                                                          COPE1599
20    CONTINUE                                                         COPE1600
30    CONTINUE                                                         COPE1601
40    CONTINUE                                                         COPE1602
      RETURN                                                           COPE1603
C     -----------------------------------------------------------------COPE1604
C                               FORMATS                                COPE1605
C     -----------------------------------------------------------------COPE1606
50    FORMAT (1H1,4X,46HOPTIMUM SENSITIVITY ANALYSIS RESULTS (NCALC=5)) COPE1607
60    FORMAT (//5X,5HTITLE/5X,20A4)                                    COPE1608
70    FORMAT (20A4)                                                    COPE1609
80    FORMAT (16I5)                                                    COPE1610
90    FORMAT (1H1,4X,47HSTANDARD SENSITIVITY ANALYSIS RESULTS (NCALC=3))COPE1611
100   FORMAT (//5X,36HNUMBER OF SENSITIVITY VARIABLES, NSV,9X,1H=,I5/5X,COPE1612
     139HNUMBER OF SENSITIVITY OBJECTIVES, NSOBJ,6X,1H=,I5)            COPE1613
110   FORMAT (//5X,52HGLOBAL NUMBERS ASSOCIATED WITH SENSITIVITY VARIABLCOPE1614
     1ES)                                                              COPE1615
120   FORMAT (5X,10I5)                                                 COPE1616
130   FORMAT (//5X,53HGLOBAL NUMBERS ASSOCIATED WITH SENSITIVITY OBJECTICOPE1617
     1VES)                                                             COPE1618
140   FORMAT (5E15.8)                                                  COPE1619
150   FORMAT (////5X,26HNOMINAL DESIGN INFORMATION//5X,31HVALUES OF SENSCOPE1620
     1ITIVITY VARIABLES)                                               COPE1621
160   FORMAT (5X,5E13.5)                                               COPE1622
170   FORMAT (//5X,41HVALUES OF SENSITIVITY OBJECTIVE FUNCTIONS)       COPE1623
180   FORMAT (/////5X,28HSENSITIVITY ANALYSIS RESULTS)                 COPE1624
190   FORMAT (//5X,15HGLOBAL VARIABLE,I5//10X,1HX,20X,4HF(X))          COPE1625
200   FORMAT (/5X,35HTHE NOMINAL VALUE IS THE ONLY VALUE/5X,27HSPECIFIEDCOPE1626
     1 FOR THIS VARIABLE)                                              COPE1627
210   FORMAT (/3X,E12.4,3X,4E13.4)                                     COPE1628
220   FORMAT (18X,4E13.4)                                              COPE1629
      END                                                              COPE1630
      SUBROUTINE COPE06 (ARRAY,RA,IA,NARRAY,NDRA,NDIA)                 COPE1631
      COMMON /COPES1/ TITLE(20)                                        COPE1632
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,MCOPE1633
     12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCRCOPE1634
     21,ISCR2,NXAPRX,NPS,NPFS,NPA,NF,INOM,IPAPRX,KMIN,KMAX,XFACT1,XFACT2COPE1635
     3,NAN2,NAN3,NFMAX,NPTOT,JNOM,MAXTRM                               COPE1636
      DIMENSION ARRAY(NARRAY), RA(NDRA), IA(NDIA)                      COPE1637
C     *****************************************************************COPE1638
C     ROUTINE TO CALCULATE FUNCTIONS OF TWO DESIGN VARIABLES FOR ALL   COPE1639
C     COMBINATIONS OF A SET OF PRESCRIBED VALUES OF THESE VARIABLES.   COPE1640
C     *****************************************************************COPE1641
C     WRITE OUTPUT INFORMATION ON SCRATCH UNIT ISCR1.                  COPE1642
C     BY G. N. VANDERPLAATS                         AUG., 1974.        COPE1643
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.                 COPE1644
      REWIND ISCR1                                                     COPE1645
C     -----------------------------------------------------------------COPE1646
C                            UNIT ISCR1 WRITE                          COPE1647
C     -----------------------------------------------------------------COPE1648
      WRITE (ISCR1,40) (TITLE(I),I=1,20)                               COPE1649
      WRITE (ISCR1,50) NCALC,N2VAR,M2VX,N2VX,M2VY,N2VY                 COPE1650
C     N2VZ.                                                            COPE1651
      M20=LOCI(20)                                                     COPE1652
      M21=LOCI(21)-1                                                   COPE1653
      WRITE (ISCR1,50) (IA(I),I=M20,M21)                               COPE1654
C     -----------------------------------------------------------------COPE1655
C                        TWO-VARIABLE FUNCTION SPACE                   COPE1656
C     -----------------------------------------------------------------COPE1657
      ICALC=2                                                          COPE1658
      KCALC=3                                                          COPE1659
      ISIGN=1                                                          COPE1660
      N20=LOCR(20)                                                     COPE1661
      N21=LOCR(21)-1                                                   COPE1662
      DO 30 I=1,M2VX                                                   COPE1663
      ARRAY(N2VX)=RA(N20)                                              COPE1664
      DO 20 J=1,M2VY                                                   COPE1665
      N21=N21+ISIGN                                                    COPE1666
      ARRAY(N2VY)=RA(N21)                                              COPE1667
C     ANALIZE.                                                         COPE1668
      NAN2=NAN2+1                                                      COPE1669
      CALL ANALIZ (ICALC)                                              COPE1670
      IF (IP2VAR.GT.0) CALL ANALIZ (KCALC)                             COPE1671
      IF (IP2VAR.GT.0) NAN3=NAN3+1                                     COPE1672
C     -----------------------------------------------------------------COPE1673
C                            UNIT ISCR1 WRITE                          COPE1674
```

```
C     ----------------------------------------------------------------COPE1675
C     WRITE X, Y.                                                     COPE1676
      WRITE (ISCR1,60) RA(N20),RA(N21)                                COPE1677
C     F(X,Y) VALUES.                                                  COPE1678
      N23=LOCR(23)                                                    COPE1679
      N24=N23                                                         COPE1680
      M20=LOCI(20)                                                    COPE1681
      DO 10 K=1,N2VAR                                                 COPE1682
      N=IA(M20)                                                       COPE1683
      RA(N24)=ARRAY(N)                                                COPE1684
      N24=N24+1                                                       COPE1685
      M20=M20+1                                                       COPE1686
10    CONTINUE                                                        COPE1687
      N24=N23+N2VAR-1                                                 COPE1688
      WRITE (ISCR1,60) (RA(K),K=N23,N24)                              COPE1689
20    CONTINUE                                                        COPE1690
      N21=N21+ISIGN                                                   COPE1691
      N20=N20+1                                                       COPE1692
      ISIGN=-ISIGN                                                    COPE1693
30    CONTINUE                                                        COPE1694
      RETURN                                                          COPE1695
C     ----------------------------------------------------------------COPE1696
C                                FORMATS                              COPE1697
C     ----------------------------------------------------------------COPE1698
C                                                                     COPE1699
C                                                                     COPE1700
C                                                                     COPE1701
40    FORMAT (20A4)                                                   COPE1702
50    FORMAT (16I5)                                                   COPE1703
60    FORMAT (5E15.8)                                                 COPE1704
      END                                                             COPE1705
      SUBROUTINE COPE07 (RA,IA,NDRA,NDIA,ISCR1)                       COPE1706
      DIMENSION RA(NDRA), IA(NDIA)                                    COPE1707
C     ****************************************************************COPE1708
C     ROUTINE TO PRINT TWO VARIABLE FUNCTION SPACE INFORMATION STORED ONCOPE1709
C     UNIT ISCR1.                                                     COPE1710
C     ****************************************************************COPE1711
C     BY G. N. VANDERPLAATS                         AUG., 1974.       COPE1712
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.                COPE1713
      REWIND ISCR1                                                    COPE1714
C     ----------------------------------------------------------------COPE1715
C                          GENERAL INFORMATION                        COPE1716
C     ----------------------------------------------------------------COPE1717
C     TITLE.                                                          COPE1718
      READ (ISCR1,60) (RA(I),I=1,20)                                  COPE1719
      READ (ISCR1,70) NCALC,N2VAR,M2VX,N2VX,M2VY,N2VY                 COPE1720
      IF (NCALC.NE.4.AND.NCALC.NE.6) RETURN                           COPE1721
      WRITE (6,50)                                                    COPE1722
      WRITE (6,40) (RA(I),I=1,20)                                     COPE1723
C     N2VZ(I),I=1,N2VAR.                                              COPE1724
      READ (ISCR1,70) (IA(I),I=1,N2VAR)                               COPE1725
C     N2VX, N2VY.                                                     COPE1726
      WRITE (6,120) N2VX,N2VY                                         COPE1727
C     N2VZ.                                                           COPE1728
      WRITE (6,130)                                                   COPE1729
      WRITE (6,80) (IA(I),I=1,N2VAR)                                  COPE1730
C     ----------------------------------------------------------------COPE1731
C               TWO-VARIABLE FUNCTION SPACE INFORMATION               COPE1732
C     ----------------------------------------------------------------COPE1733
      DO 30 I=1,M2VX                                                  COPE1734
      WRITE (6,140)                                                   COPE1735
      DO 30 J=1,M2VY                                                  COPE1736
C     X, Y.                                                           COPE1737
      READ (ISCR1,150) XX,YY                                          COPE1738
C     F(X,Y).                                                         COPE1739
      READ (ISCR1,150) (RA(K),K=1,N2VAR)                              COPE1740
      N=4                                                             COPE1741
      IF (N2VAR.LT.4) N=N2VAR                                         COPE1742
      IF (J.EQ.1) WRITE (6,100) XX,YY,(RA(K),K=1,N)                   COPE1743
      IF (J.GT.1) WRITE (6,90) YY,(RA(K),K=1,N)                       COPE1744
      IF (N.GE.N2VAR) GO TO 20                                        COPE1745
      N=5                                                             COPE1746
      M=(N2VAR-1)/4                                                   COPE1747
      DO 10 K=1,M                                                     COPE1748
      L=N+3                                                           COPE1749
      IF (L.GT.N2VAR) L=N2VAR                                         COPE1750
      WRITE (6,110) (RA(KK),KK=N,L)                                   COPE1751
10    N=L+1                                                           COPE1752
20    CONTINUE                                                        COPE1753
30    CONTINUE                                                        COPE1754
      RETURN                                                          COPE1755
C     ----------------------------------------------------------------COPE1756
C                                FORMATS                              COPE1757
C     ----------------------------------------------------------------COPE1758
40    FORMAT (//5X,5HTITLE/5X,20A4)                                   COPE1759
50    FORMAT (1H1,4X,35HTWO-VARIABLE FUNCTION SPACE RESULTS)          COPE1760
60    FORMAT (20A4)                                                   COPE1761
70    FORMAT (16I5)                                                   COPE1762
80    FORMAT (5X,10I5)                                                COPE1763
90    FORMAT (/15X,E12.4,3X,4E13.4)                                   COPE1764
100   FORMAT (/3X,2E12.4,3X,4E13.4)                                   COPE1765
110   FORMAT (30X,4E13.4)                                             COPE1766
120   FORMAT (////5X,48HGLOBAL NUMBER ASSOCIATED WITH X-VARIABLE, N2VX =COPE1767
     1,I5//5X,48HGLOBAL NUMBER ASSOCIATED WITH Y-VARIABLE, N2VY =,I5) COPE1768
130   FORMAT (//5X,37HGLOBAL NUMBERS ASSOCIATED WITH F(X,Y))          COPE1769
140   FORMAT (//10X,1HX,11X,1HY,20X,6HF(X,Y))                         COPE1770
150   FORMAT (5E15.8)                                                 COPE1771
      END                                                             COPE1772
      SUBROUTINE COPE08 (A,B,IFORM,NFLD)                              COPE1773
      DIMENSION A(1), B(1), C(10)                                     COPE1774
      DATA COMMA/1H,/,BLANK/1H /                                      COPE1775
C     ROUTINE TO CONVEPT UNFORMATTED DATA TO FORMATTED DATA IN FIELDS COPE1776
C     OF 10,EACH FIELD RIGHT JUSTIFIED.                               COPE1777
C     BY G. N. VANDERPLAATS                            AUG., 1978.    COPE1778
C     NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.                COPE1779
C --- INPUT.                                                          COPE1780
C     A       - ARRAY OF DATA SEPERATED BY COMMAS.  MINIMUM DIMENSION OF COPE1781
C               IS A(80).                                             COPE1782
C --- OUTPUT.                                                         COPE1783
C     B       - ARRAY OD DATA IN FIELDS OF 10 AND RIGHT JUSTIFIED.    COPE1784
C               MINIMUM DIMENSION OF B IS 10*NFLD.                    COPE1785
C     IFORM   - 0 IF A WAS UNFORMATTED.                               COPE1786
C               1 IF A WAS ALLREADY FORMATTED.                        COPE1787
C     NFLD    - NUMBER OF FIELDS OF DATA.                             COPE1788
C --- NOTE.                                                           COPE1789
C     1)  DATA IS ASSUMED TO BE REAL OR INTEGER DATA WITH NO EMBEDDED COPE1790
C         BLANKS WITHIN A GIVEN FIELD.                                COPE1791
C     2)  DATA IS CONSIDERED UNFORMATTED IF                           COPE1792
C         A)  A COMMA IS FOUND                                        COPE1793
C         B)  LAST NON-BLANK CHARACTER IS IN COLUMN 1-10 AND THERE IS COPE1794
```

```
C             NO DECIMAL AND IT IS NOT RIGHT JUSTIFIED.                 COPE1795
C                                                                       COPE1796
      IFORM=0                                                           COPE1797
C     SEARCH FOR LAST NON-BLANK CHARACTER AND SEARCH FOR COMMA.         COPE1798
C    CALCULATE NUMBER OF NON-BLANK SETS.                                COPE1799
      INON=0                                                            COPE1800
      KNON=0                                                            COPE1801
      LST=0                                                             COPE1802
      DO 10 I=1,80                                                      COPE1803
      IF (A(I).EQ.COMMA) GO TO 20                                       COPE1804
      JNON=INON                                                         COPE1805
      IF (A(I).EQ.BLANK) INON=0                                         COPE1806
      IF (A(I).NE.BLANK) INON=1                                         COPE1807
      IF (INON.GT.JNON) KNON=KNON+1                                     COPE1808
      IF (A(I).NE.BLANK) LST=I                                          COPE1809
10    CONTINUE                                                          COPE1810
C     NO COMMA WAS FOUND.  DATA MAY BE FORMATTED.                       COPE1811
      IF (LST.GE.10) GO TO 90                                           COPE1812
C     IF MORE THAN ONE SETS OF CHARACTERS, DATA IS ASSUMED FORMATTED.   COPE1813
      IF (KNON.GT.1) GO TO 90                                           COPE1814
20    CONTINUE                                                          COPE1815
C     DATA IS UNFORMATTED.                                              COPE1816
      K2=10                                                             COPE1817
      NFLD=0                                                            COPE1818
      I=0                                                               COPE1819
30    CONTINUE                                                          COPE1820
      I=I+1                                                             COPE1821
      IF (I.GT.80) GO TO 110                                            COPE1822
C     IGNORE LEADING BLANKS.                                            COPE1823
      IF (A(I).EQ.BLANK) GO TO 30                                       COPE1824
C     CALCULATE NUMBER OF NON-BLANK CHARACTERS IN THIS FIELD.           COPE1825
      JJ=0                                                              COPE1826
      DO 40 J=I,80                                                      COPE1827
      IF (A(J).EQ.COMMA.OR.A(J).EQ.BLANK) GO TO 50                      COPE1828
      JJ=JJ+1                                                           COPE1829
40    C(JJ)=A(J)                                                        COPE1830
50    NFLD=NFLD+1                                                       COPE1831
      I=I+JJ                                                            COPE1832
C     BLANK FIELD NFLD OF B.                                            COPE1833
      K1=K2-9                                                           COPE1834
      DO 60 K=K1,K2                                                     COPE1835
60    B(K)=BLANK                                                        COPE1836
C     STORE C IN FIELD NFLD OF B, RIGHT JUSTIFIED.                      COPE1837
      IF (JJ.EQ.0) GO TO 80                                             COPE1838
      J1=JJ                                                             COPE1839
      K=K2                                                              COPE1840
      DO 70 L=1,JJ                                                      COPE1841
      B(K)=C(J1)                                                        COPE1842
      K=K-1                                                             COPE1843
70    J1=J1-1                                                           COPE1844
80    K2=K2+10                                                          COPE1845
      GO TO 30                                                          COPE1846
90    CONTINUE                                                          COPE1847
C     FORMATTED INPUT.  STORE A DIRECTLY IN B.                          COPE1848
      IFORM=1                                                           COPE1849
      NFLD=8                                                            COPE1850
      DO 100 I=1,80                                                     COPE1851
100   B(I)=A(I)                                                         COPE1852
110   CONTINUE                                                          COPE1853
      RETURN                                                            COPE1854
      END                                                               COPE1855
      SUBROUTINE COPE09                                                 COPE1856
      COMMON /CNMN1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAXCOPE1857
     1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,ITCOPE1858
     2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER                               COPE1859
      COMMON /COPES2/ RA(5000),IA(1000)                                 COPE1860
      COMMON /COPES3/ SGNOPT,NCALC,IOBJ,NSV,NSOBJ,NCONA,N2VX,M2VX,N2VY,MCOPE1861
     12VY,N2VAR,IPSENS,IP2VAR,IPDBG,NACMX1,NDVTOT,LOCR(25),LOCI(25),ISCRCOPE1862
     21,ISCR2,NXAPRX,NPS,NPFS,NPA,NF,INOM,IPAPRX,KMIN,KMAX,XFACT1,XFACT2COPE1863
     3,NAN2,NAN3,NPMAX,NPTOT,JNOM,MAXTRM                                COPE1864
      COMMON /GLOBCM/ ARRAY(1500)                                       COPE1865
C     ******************************************************************COPE1866
C                     ROUTINE TO DO APPROXIMATE OPTIMIZATION.           COPE1867
C     ******************************************************************COPE1868
C                                                                       COPE1869
C     BY G. N. VANDERPLAATS                                 JAN., 1979. COPE1870
C     NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.                  COPE1871
C                                                                       COPE1872
      NFDGSV=NFDG                                                       COPE1873
      NFDG=1                                                            COPE1874
      IF (NFDGSV.LT.0) NFDG=0                                           COPE1875
      CTMIN=ABS(CTMIN)                                                  COPE1876
      CTLMIN=ABS(CTLMIN)                                                COPE1877
      IF (CTMIN.LE.0.) CTMIN=.0005                                      COPE1878
      IF (CTLMIN.LE.0.) CTLMIN=.0001                                    COPE1879
      IF (ABS(CT).LE.0.) CT=-.01                                        COPE1880
      IF (ABS(CTL).LE.0.) CTL=-.001                                     COPE1881
      IF (DELFUN.LE.0.) DELFUN=.0001                                    COPE1882
      IF (IPDBG.LT.1) IPRINT=0                                          COPE1883
      IF (ITMAX.LT.50) ITMAX=50                                         COPE1884
      NSIDE=1                                                           COPE1885
      KOUNT=0                                                           COPE1886
C     ------------------------------------------------------------------COPE1887
C                            ARRAY STARTING LOCATIONS.                  COPE1888
C     ------------------------------------------------------------------COPE1889
      NN1=NXAPRX+2                                                      COPE1890
      NXV=LOCR(23)                                                      COPE1891
      NVLB=NXV+NN1                                                      COPE1892
      NVUB=NVLB+NN1                                                     COPE1893
      NXNOM=NVUB+NN1                                                    COPE1894
      NDX=NXNOM+NXAPRX                                                  COPE1895
      NFNOM=NDX+NDV                                                     COPE1896
      NFNEW=NFNOM+NF                                                    COPE1897
      NBTAY=NFNEW+NF                                                    COPE1898
      NBR=NXAPRX+(NXAPRX*(NXAPRX+1))/2                                  COPE1899
      IF (MAXTRM.LT.3) NBR=MAXTRM*NXAPRX                                COPE1900
      NTMP=NBTAY+NBR*NF                                                 COPE1901
      NBLU=LOCR(6)                                                      COPE1902
      NISC=LOCI(4)                                                      COPE1903
      NIGFN=LOCI(23)                                                    COPE1904
      NIDV=NIGFN+NCONA                                                  COPE1905
      IF (KMAX.LT.0) GO TO 160                                          COPE1906
C     CONMIN ARRAYS.                                                    COPE1907
C     DIMENSIONS.                                                       COPE1908
      NN1=NDV+2                                                         COPE1909
      NN2=2*NDV+NCON                                                    COPE1910
      NN3=NACMX1                                                        COPE1911
      NN4=NN3                                                           COPE1912
      IF (NDV.GT.NN4) NN4=NDV                                           COPE1913
      NN5=2*NN4                                                         COPE1914
```

```
C     SCAL, DF, G, A, S, G1, G2, C, B, ISC, IC, NS1.
      NNSCAL=LOCR(4)
      NDF=NTMP
      NG=NDF+NN1
      NA=NG+NN2
      NS=NA+NN1*NN3
      NG1=NS+NN1
      NG2=NG1+NN2
      NC=NG2+NN2
      NB=NC+NN4
      NISC=LOCI(4)
      NIC=NIDV+NXAPRX
      NMS1=NIC+NN3
C     -----------------------------------------------------------------
C                     DETERMINE IOBJA, ARRAYS IGFN AND IDV.
C     -----------------------------------------------------------------
C     IOBJA.
      M6=LOCI(6)
      DO 10 I=1,NF
      J=IA(M6)
      IOBJA=I
      IF (J.EQ.IOBJ) GO TO 20
   10 M6=M6+1
C     ERROR - IOBJA NOT FOUND.
   20 CONTINUE
      IF (NCONA.EQ.0) GO TO 60
C     IGFN ARRAY.
      M3=LOCI(3)
      M23=LOCI(23)
      DO 50 I=1,NCONA
C     GLOBAL LOCATIONS OF CONSTRAINED PARAMETERS.
      J=IA(M3)
      M3=M3+1
C     LOCAL VARIABLE, F, LOCATION.
      M6=LOCI(6)
      DO 30 K=1,NF
      KK=K
      L=IA(M6)
      IF (L.EQ.J) GO TO 40
   30 M6=M6+1
C     ERROR - CONSTRAINED VARIABLE IS NOT AN APPROXIMATE FUNCTION.
   40 CONTINUE
      IA(M23)=KK
   50 M23=M23+1
   60 CONTINUE
C     IDV ARRAY.
      N3=NIDV
      N5=LOCI(5)
      DO 90 I=1,NXAPRX
      IA(N3)=0
C     GLOBAL LOCATION.
      II=IA(N5)
      N5=N5+1
C     FIND CORRESPONDING DESIGN VARIABLE.
      N1=LOCI(1)
      N2=LOCI(2)
      DO 70 J=1,NDVTOT
      IF (IA(N1).EQ.II) GO TO 80
      N1=N1+1
   70 N2=N2+1
C     THIS APPROXIMATING VARIABLE IS NOT A DESIGN VARIABLE.
      GO TO 90
   80 CONTINUE
C     DESIGN VARIABLE NUMBER.
      IA(N3)=IA(N2)
   90 N3=N3+1
C     CHECK TO BE SURE EACH INDEPENDENT DESIGN VARIABLE IS ALSO AN
C     APPROXIMATING VARIABLE.
      DO 120 I=1,NDV
      N1=LOCI(1)
      N2=LOCI(2)
      DO 110 J=1,NDVTOT
C     IS THIS DESIGN VARIABLE I.
      IF (IA(N2).NE.I) GO TO 110
C     YES.
C     GLOBAL VARIABLE NUMBER.
      IGLOB=IA(N1)
      N1=N1+1
      N5=LOCI(5)
      DO 100 K=1,NXAPRX
C     IS THIS THE SAME AS IGLOB.
      IF (IA(N5).EQ.IGLOB) GO TO 120
C     NO.
  100 N5=N5+1
  110 N2=N2+1
C     ERROR - DESIGN VARIABLE IS NOT AN APPROXIMATING VARIABLE.
  120 CONTINUE
C     -----------------------------------------------------------------
C                   BEGIN SEQUENTIAL APPROXIMATE OPTIMIZATION.
C     -----------------------------------------------------------------
      ICK1=0
      ICK2=0
      ICK3=0
  130 CONTINUE
      KOUNT=KOUNT+1
      IF (IPAPRX.LT.1.OR.IPAPRX.EQ.3) GO TO 160
      IF (KMAX.LT.0) GO TO 160
C     PRINT INITIAL INFORMATION.
C     TITLE.
      IF (KOUNT.GT.1) GO TO 150
      WRITE (6,670)
C     OBJECTIVE FUNCTION.
      WRITE (6,550) IOBJA
      IF (NCONA.EQ.0) GO TO 140
C     CONSTRAINTS.
      WRITE (6,560)
      N1=NIGFN
      N2=N1+NCONA-1
      WRITE (6,570) (IA(I),I=N1,N2)
  140 CONTINUE
C     DESIGN VARIABLES.
      WRITE (6,580)
      N1=NIDV
      N2=N1+NXAPRX-1
      WRITE (6,570) (IA(I),I=N1,N2)
  150 CONTINUE
C     ITERATION NUMBER.
      WRITE (6,680) KOUNT
  160 CONTINUE
      NP=NPTOT-1
```

```
C     -----------------------------------------------------------------COPE2035
C               SET UP ARRAYS XNOM, FNOM, XI, Y.                       COPE2036
C     -----------------------------------------------------------------COPE2037
      NXI=NTMP                                                         COPE2038
      NY=NXI+NXAPRX*NPTOT                                              COPE2039
      NWGHT=NY+NF*NPTOT                                                COPE2040
      IF (KMAX.LT.0.AND.INOM.LT.1) INOM=1                              COPE2041
      CALL COPE10 (RA(NXI),RA(NY),RA(NXNOM),RA(NFNOM),NPTOT,KOUNT,RA(NBLCOPE2042
     1U),IA(NIGFN),IOBJA,IA(NISC),NXAPRX,NF,NCONA,SGNOPT,CTMIN,CTLMIN,ISCOPE2043
     2CR2,RA(NWGHT),INOM,NPMAX,JNOM)                                   COPE2044
      IF (KMAX.LT.0) GO TO 170                                         COPE2045
C     CURRENT OBJECTIVE.                                               COPE2046
      N1=NFNOM+IOBJA-1                                                 COPE2047
      OBJSAV=-RA(N1)*SGNOPT                                            COPE2048
      IF (KOUNT.LT.2.AND.DABFUN.LE.0.) DABFUN=.001*ABS(OBJSAV)         COPE2049
      IF (DABFUN.LT.1.0E-10) DABFUN=1.0E-10                            COPE2050
      IF (IPAPRX.LT.1.OR.IPAPRX.EQ.3) GO TO 170                        COPE2051
C     PRINT CURRENT NOMINAL.                                           COPE2052
      WRITE (6,690) INOM                                               COPE2053
      N2=NXNOM+NXAPRX-1                                                COPE2054
      WRITE (6,700) (RA(I),I=NXNOM,N2)                                 COPE2055
      WRITE (6,710)                                                    COPE2056
      N2=NFNOM+NF-1                                                    COPE2057
      WRITE (6,700) (RA(I),I=NFNOM,N2)                                 COPE2058
170   CONTINUE                                                         COPE2059
C     -----------------------------------------------------------------COPE2060
C               LEAST SQUARES FIT FOR TAYLER SERIES EXPANSION.         COPE2061
C     -----------------------------------------------------------------COPE2062
      NX1=NXAPRX                                                       COPE2063
      M=NX1+(NX1*(NX1+1))/2                                            COPE2064
      IF (M.GT.NP) M=NP                                                COPE2065
      IF (NP.LT.NXAPRX) M=NXAPRX                                       COPE2066
      MMAX=NXAPRX*MAXTRM                                               COPE2067
      IF(MAXTRM.GT.2) MMAX=M                                           COPE2068
      IF (M.GT.MMAX) M=MMAX                                            COPE2069
      NAA=NWGHT+NP                                                     COPE2070
      NFF=NAA+(M*(M+1))/2                                              COPE2071
      NGG=NFF+M                                                        COPE2072
      CALL COPE12 (RA(NXI),RA(NY),NX1,NP,NF,M,RA(NBTAY),RA(NAA),RA(NFF),COPE2073
     1RA(NGG),NXAPRX,NF,NBR,RA(NWGHT),NER)                             COPE2074
      IF (NER.GT.0) WRITE (6,590)                                      COPE2075
      IF (KMAX.LT.0) GO TO 530                                         COPE2076
      IF (IPAPRX.LT.2.AND.IPAPRX.NE.4) GO TO 190                       COPE2077
C     PRINT TAYLER SERIES COEFFICIENTS.                                COPE2078
      WRITE (6,720)                                                    COPE2079
      N3=LOCI(6)                                                       COPE2080
      N1=NBTAY                                                         COPE2081
      DO 180 J=1,NF                                                    COPE2082
      N2=N1+M-1                                                        COPE2083
      WRITE (6,600) J,IA(N3)                                           COPE2084
      WRITE (6,700) (RA(I),I=N1,N2)                                    COPE2085
      N1=N1+NBR                                                        COPE2086
      N3=N3+1                                                          COPE2087
180   CONTINUE                                                         COPE2088
190   CONTINUE                                                         COPE2089
      IF (KOUNT.GT.KMAX) GO TO 470                                     COPE2090
C     -----------------------------------------------------------------COPE2091
C                         INITIALIZE XV, DX, VLB, VUB.                 COPE2092
C     -----------------------------------------------------------------COPE2093
      N1=NXV                                                           COPE2094
      DO 200 I=1,NXAPRX                                                COPE2095
      RA(N1)=0.                                                        COPE2096
200   N1=N1+1                                                          COPE2097
      N2=NVLB                                                          COPE2098
      N3=NVUB                                                          COPE2099
      N4=NXNOM                                                         COPE2100
      N5=LOCP(2)                                                       COPE2101
      N6=LOCR(3)                                                       COPE2102
      N7=LOCR(7)                                                       COPE2103
      N8=NDX                                                           COPE2104
      L1=M-NDV                                                         COPE2105
      L2=L1-NDV                                                        COPE2106
      ICK=ICK1+ICK2+ICK3                                               COPE2107
      DO 210 I=1,NDV                                                   COPE2108
      RA(N8)=0.                                                        COPE2109
      XFACT=1.                                                         COPE2110
      IF (I.LE.L1) XFACT=XFACT1                                        COPE2111
      L2=L2-NDV+I                                                      COPE2112
      IF (L2.GE.0) XFACT=XFACT2                                        COPE2113
C     REDUCE BOUNDS IF ANY ICK.GT.0.                                   COPE2114
      IF (ICK.GT.0) XFACT=.5                                           COPE2115
      DX=XFACT*RA(N7)                                                  COPE2116
      XX=RA(N4)                                                        COPE2117
      DXL=XX-RA(N5)                                                    COPE2118
      IF (DXL.GT.DX) DXL=DX                                            COPE2119
      DXU=RA(N6)-XX                                                    COPE2120
      IF (DXU.GT.DX) DXU=DX                                            COPE2121
      RA(N2)=-DXL                                                      COPE2122
      RA(N3)=DXU                                                       COPE2123
      N2=N2+1                                                          COPE2124
      N3=N3+1                                                          COPE2125
      N4=N4+1                                                          COPE2126
      N5=N5+1                                                          COPE2127
      N6=N6+1                                                          COPE2128
      N7=N7+1                                                          COPE2129
210   N8=N8+1                                                          COPE2130
      IF (IPAPRX.LT.2.AND.IPAPRX.NE.4) GO TO 220                       COPE2131
      WRITE (6,610)                                                    COPE2132
      N1=NVLB+NDV-1                                                    COPE2133
      WRITE (6,700) (RA(I),I=NVLB,N1)                                  COPE2134
      WRITE (6,620)                                                    COPE2135
      N1=NVUB+NDV-1                                                    COPE2136
      WRITE (6,700) (RA(I),I=NVUB,N1)                                  COPE2137
220   CONTINUE                                                         COPE2138
C     -----------------------------------------------------------------COPE2139
C                         OPTIMIZE APPROXIMATE FUNCTION.               COPE2140
C     -----------------------------------------------------------------COPE2141
C     OPTIMIZATION.                                                    COPE2142
      IGOTO=0                                                          COPE2143
C     CALL CONMIN (X,VLB,VUB,G,SCAL,DF,A,S,G1,G2,B,C,ISC,IC,MS1,N1,N2,N3COPE2144
C    *,N4,N5)                                                          COPE2145
230   CONTINUE                                                         COPE2146
      CALL CONMIN (RA(NDX),RA(NVLB),RA(NVUB),RA(NG),RA(NNSCAL),RA(NDF),RCOPE2147
     1A(NA),RA(NS),RA(NG1),RA(NG2),RA(NB),RA(NC),IA(NISC),IA(NIC),IA(NMSCOPE2148
     21),NN1,NN2,NN3,NN4,NN5)                                          COPE2149
C     TRANSFER VARIABLES FROM DX TO XV.                                COPE2150
      N1=NIDV                                                          COPE2151
      N2=NXV                                                           COPE2152
      DO 240 I=1,NXAPRX                                                COPE2153
      II=IA(N1)                                                        COPE2154
```

```
      N4=NDX+II-1                                                       COPE2155
      IF (II.GT.0.AND.II.LE.NDV) RA(N2)=RA(N4)                          COPE2156
      N1=N1+1                                                           COPE2157
      N2=N2+1                                                           COPE2158
  240 N3=N3+1                                                           COPE2159
C     APPROXIMATE ANALYSIS.                                             COPE2160
      CALL COPE15 (RA(NXV),RA(NG),RA(NDF),RA(NA),IA(NISC),IA(NIC),NN1,RACOPE2161
     1(NBLU),NX1,IOBJA,M,RA(NFNOM),RA(NFNEW),RA(NBTAY),NBR,IA(NIGFN),CT,COPE2162
     2CTL,INFO,NAC,NCONA,NDV,NF,OBJ,SGNOPT)                             COPE2163
      IF (IGOTO.GT.0) GO TO 230                                         COPE2164
C     IF DESIGN PRODUCED ZERO DELTA-X TWICE IN A ROW AND KOUNT.GE.KMIN, COPE2165
C     TERMINATE.                                                        COPE2166
      ICK1=ICK1+1                                                       COPE2167
      SUM=0.                                                            COPE2168
      N1=NXV                                                            COPE2169
      DO 250 I=1,NDV                                                    COPE2170
      SUM=SUM+RA(N1)**2                                                 COPE2171
  250 N1=N1+1                                                           COPE2172
      IF (SUM.GT.1.0E-10) ICK1=0                                        COPE2173
      IF (IPAPRX.GT.0.AND.IPAPRX.NE.3) WRITE (6,730)                    COPE2174
      IF (ICK1.GE.2.AND.KOUNT.GE.KMIN) GO TO 360                        COPE2175
C     ------------------------------------------------------------------COPE2176
C               INSURE NEW X-VECTOR IS INDEPENDENT                      COPE2177
C     ------------------------------------------------------------------COPE2178
      JJJ=0                                                             COPE2179
  260 JJJ=JJJ+1                                                         COPE2180
C     NOMINAL X-VECTOR.                                                 COPE2181
      N1=NTMP                                                           COPE2182
      N2=NXNOM                                                          COPE2183
      N3=NXV                                                            COPE2184
      DO 270 I=1,NXAPRX                                                 COPE2185
      RA(N1)=RA(N2)+RA(N3)                                              COPE2186
      N1=N1+1                                                           COPE2187
      N2=N2+1                                                           COPE2188
  270 N3=N3+1                                                           COPE2189
C     READ X-VECTORS ONE AT A TIME AND COMPARE TO XNOM.                 COPE2190
      REWIND ISCR2                                                      COPE2191
      N1=NTMP+NXAPRX                                                    COPE2192
      N2=N1+NXAPRX-1                                                    COPE2193
      N3=N2+1                                                           COPE2194
      N4=N3+NF-1                                                        COPE2195
      DO 290 J=1,NPTOT                                                  COPE2196
      KK=J                                                              COPE2197
C     X-VECTOR.                                                         COPE2198
      READ (ISCR2) (RA(I),I=N1,N2)                                      COPE2199
C     Y-VECTOR.  NOT USED.  READ TO POSITION ISCR2.                     COPE2200
      READ (ISCR2) (RA(I),I=N3,N4)                                      COPE2201
C     COMPARE X WITH XNOM.                                              COPE2202
      N5=N1                                                             COPE2203
      N6=NTMP                                                           COPE2204
      SUM=0.                                                            COPE2205
      DO 280 I=1,NXAPRX                                                 COPE2206
      SUM=SUM+(RA(N5)-RA(N6))**2                                        COPE2207
      N5=N5+1                                                           COPE2208
  280 N6=N6+1                                                           COPE2209
      IF (SUM.LT.1.0E-10) GO TO 300                                     COPE2210
  290 CONTINUE                                                          COPE2211
      GO TO 360                                                         COPE2212
  300 CONTINUE                                                          COPE2213
C     THIS DESIGN IS SAME AS A PREVIOUS DESIGN.                         COPE2214
C     MODIFY DELTA-X VECTOR.                                            COPE2215
      N6=NVUB                                                           COPE2216
      N7=NXV                                                            COPE2217
      N8=NVLB                                                           COPE2218
      N9=NXV+NDV-1                                                      COPE2219
      IF (IPAPRX.LT.1.OR.IPAPRX.EQ.3) GO TO 310                         COPE2220
      WRITE (6,630)                                                     COPE2221
      WRITE (6,740)                                                     COPE2222
      WRITE (6,700) (RA(I),I=N7,N9)                                     COPE2223
      WRITE (6,750)                                                     COPE2224
      N9=NTMP+NXAPRX-1                                                  COPE2225
      WRITE (6,700) (RA(I),I=NTMP,N9)                                   COPE2226
      WRITE (6,640)                                                     COPE2227
  310 CONTINUE                                                          COPE2228
      AMULT=.01*FLOAT(JJJ)                                              COPE2229
      DO 320 I=1,NDV                                                    COPE2230
      BU=RA(N6)                                                         COPE2231
      BL=RA(N8)                                                         COPE2232
      IF (BL.LT.-1.0E+15) BL=0.                                         COPE2233
      IF (BU.GT.1.0E+15) BU=0.                                          COPE2234
      DB=ABS(BU-BL)                                                     COPE2235
      IF (DB.LT.1.0E-6) DB=.1                                           COPE2236
      DX=RA(N7)+AMULT*DB                                                COPE2237
      IF (DX.GT.RA(N6)) DX=DX-2.*AMULT*DB                               COPE2238
      IF (DX.LT.RA(N8)) DX=DX+1.5*AMULT*DB                              COPE2239
      RA(N7)=DX                                                         COPE2240
      N6=N6+1                                                           COPE2241
      N7=N7+1                                                           COPE2242
      N8=N8+1                                                           COPE2243
  320 CONTINUE                                                          COPE2244
      CALL COPE15 (RA(NXV),RA(NG),RA(NDF),RA(NA),IA(NISC),IA(NIC),NN1,RACOPE2245
     1(NBLU),NX1,IOBJA,M,RA(NFNOM),RA(NFNEW),RA(NBTAY),NBR,IA(NIGFN),CT,COPE2246
     2CTL,INFO,NAC,NCONA,NDV,NF,OBJ,SGNOPT)                             COPE2247
      IF (JJJ.LT.4) GO TO 260                                           COPE2248
C     FOUR TRIES HAVE FAILED TO PRODUCE A USABLE X-VECTOR.              COPE2249
C     USE LATEST TRY.                                                   COPE2250
C     POSITION ISCR2 IF NEEDED.                                         COPE2251
      IF (KK.EQ.NPTOT) GO TO 340                                        COPE2252
      KK=KK+1                                                           COPE2253
      DO 330 J=KK,NPTOT                                                 COPE2254
      READ (ISCR2) (RA(I),I=N1,N2)                                      COPE2255
  330 READ (ISCR2) (RA(I),I=N3,N4)                                      COPE2256
  340 CONTINUE                                                          COPE2257
      IF (IPAPRX.LT.1.OR.IPAPRX.EQ.3) GO TO 350                         COPE2258
      IF (JJJ.EQ.4) WRITE (6,650)                                       COPE2259
  350 CONTINUE                                                          COPE2260
  360 CONTINUE                                                          COPE2261
C     ------------------------------------------------------------------COPE2262
C                           UPDATE ANALYSIS.                            COPE2263
C     ------------------------------------------------------------------COPE2264
C     XNOM.                                                             COPE2265
      N1=NXNOM                                                          COPE2266
      N2=NXV                                                            COPE2267
      DO 370 I=1,NXAPRX                                                 COPE2268
      RA(N1)=RA(N1)+RA(N2)                                              COPE2269
      N1=N1+1                                                           COPE2270
  370 N2=N2+1                                                           COPE2271
C     GLOBAL VARIABLES.                                                 COPE2272
      N3=NXNOM                                                          COPE2273
      N4=NIDV                                                           COPE2274
```

```
      DO 410 I=1,NXAPRX
C     DESIGN VARIABLE NUMBER.
      II=IA(N4)
      IF (II.EQ.0) GO TO 400
C     DESIGN VARIABLE UPDATE.
      N1=LOCI(1)
      N2=LOCI(2)
      N5=LOCR(5)
      DO 390 J=1,NDVTOT
      IF (IA(N2).NE.II) GO TO 380
C     UPDATE VARIABLE J.
      JJ=IA(N1)
      ARRAY(JJ)=RA(N3)*RA(N5)
  380 N1=N1+1
      N2=N2+1
      N5=N5+1
  390 CONTINUE
  400 N3=N3+1
  410 N4=N4+1
      IF (IPAPRX.LT.1.OR.IPAPRX.EQ.3) GO TO 420
C     PRINT APPROXIMATE OPTIMIZATION INFORMATION.
      WRITE (6,740)
      N2=NXV+NDV-1
      WRITE (6,700) (RA(I),I=NXV,N2)
      WRITE (6,750)
      N2=NXNOM+NXAPRX-1
      WRITE (6,700) (RA(I),I=NXNOM,N2)
      WRITE (6,760)
      N2=NFNEW+NF-1
      WRITE (6,700) (RA(I),I=NFNEW,N2)
  420 CONTINUE
      IF ((ICK1.GE.2.AND.KOUNT.GE.KMIN).AND.(IPAPRX.GT.0.AND.IPAPRX.NE.3
     1)) WRITE (6,540)
      IF (ICK1.GE.2.AND.KOUNT.GE.KMIN) GO TO 460
      ICALC=2
      NAN2=NAN2+1
      CALL ANALIZ (ICALC)
C     NEW FUNCTION VALUES.
      N1=NFNOM
      M6=LOCI(6)
      DO 430 I=1,NF
      II=IA(M6)
      M6=M6+1
      RA(N1)=ARRAY(II)
  430 N1=N1+1
      IF (IPDBG.LT.1) GO TO 440
C     DEBUG OUTPUT.
      NAN3=NAN3+1
      ICALC=3
      CALL ANALIZ (ICALC)
  440 CONTINUE
      IF (IPAPRX.LT.1.OR.IPAPRX.EQ.3) GO TO 450
C     PRINT PRECISE FUNCTION VALUES.
      WRITE (6,770)
      N2=NFNOM+NF-1
      WRITE (6,700) (RA(I),I=NFNOM,N2)
  450 CONTINUE
C     NEW OBJECTIVE.
      N1=NFNOM+IOBJA-1
      OBJ=-RA(N1)*SGNOPT
C     ------------------------------------------------------------------
C                   WRITE NEW X AND F VALUES ON ISCR2.
C     ------------------------------------------------------------------
C     X-VECTOR.
      N1=NXNOM+NXAPRX-1
      WRITE (ISCR2) (RA(I),I=NXNOM,N1)
C     FUNCTIONS.
      N1=NFNOM+NF-1
      WRITE (ISCR2) (RA(I),I=NFNOM,N1)
C     UPDATE PARAMETERS.
      NPTOT=NPTOT+1
      IF (JJJ.LT.2.OR.KOUNT.LT.KMIN) INOM=NPTOT
C     ------------------------------------------------------------------
C                        CONVERGENCE CHECK.
C     ------------------------------------------------------------------
      IF (KOUNT.LT.KMIN) GO TO 130
      ICK2=ICK2+1
      ICK3=ICK3+1
      DEL=ABS(OBJ)
      IF (DEL.LT.1.0E-6) DEL=1.0E-6
      DEL=(OBJ-OBJSAV)/DEL
      DEL=ABS(DEL)
      DEL=(OBJ-OBJSAV)/DEL
      DEL=ABS(DEL)
      IF (DEL.GT.DELFUN) ICK2=0
      DEL=ABS(OBJ-OBJSAV)
      IF (DEL.GT.DABFUN) ICK3=0
      IF (ICK2.GE.2.OR.ICK3.GE.2) GO TO 460
      IF (KOUNT.LT.KMAX) GO TO 130
  460 CONTINUE
C     ------------------------------------------------------------------
C                       FINAL INFORMATION.
C     ------------------------------------------------------------------
      IF (IPAPRX.GT.0.AND.IPAPRX.NE.3) WRITE (6,660)
C     GO BACK AND PICK BEST DESIGN.
      INOM=0
      KOUNT=KMAX+1
      IF (KOUNT.LT.JNOM) KOUNT=JNOM+1
      CTSAV=CT
      CTLSAV=CTL
      IF (ABS(CT).LT.1.0E-10) CT=-.004
      IF (ABS(CTL).LT.1.0E-10) CTL=-.001
      GO TO 160
  470 CONTINUE
      CT=CTSAV
      CTL=CTLSAV
C     STORE FINAL VALUES OF XNOM IN GLOBAL ARRAY.
      N3=NXNOM
      N4=NIDV
      DO 510 I=1,NXAPRX
C     DESIGN VARIABLE NUMBER.
      II=IA(N4)
      IF (II.EQ.0) GO TO 500
C     DESIGN VARIABLE UPDATE.
      N1=LOCI(1)
      N2=LOCI(2)
      N5=LOCR(5)
      DO 490 J=1,NDVTOT
      IF (IA(N2).NE.II) GO TO 480
C     UPDATE VARIABLE J.
```

```
      JJ=IA(N1)                                                         COPE2395
      ARRAY(JJ)=RA(N3)*RA(N5)                                           COPE2396
  480 N1=N1+1                                                           COPE2397
      N2=N2+1                                                           COPE2398
      N5=N5+1                                                           COPE2399
  490 CONTINUE                                                          COPE2400
  500 N3=N3+1                                                           COPE2401
  510 N4=N4+1                                                           COPE2402
C     STORE FINAL VALUES OF FNOM IN GLOBAL ARRAY.                       COPE2403
      M6=LOCI(6)                                                        COPE2404
      N1=NFNOM                                                          COPE2405
      DO 520 I=1,NF                                                     COPE2406
      II=IA(M6)                                                         COPE2407
      M6=M6+1                                                           COPE2408
      ARRAY(II)=RA(N1)                                                  COPE2409
  520 N1=N1+1                                                           COPE2410
  530 CONTINUE                                                          COPE2411
      RETURN                                                            COPE2412
C     ------------------------------------------------------------------COPE2413
C                                FORMATS                                COPE2414
C     ------------------------------------------------------------------COPE2415
  540 FORMAT (//5X,71HTWO CONSECUTIVE APPROXIMATE OPTIMIZATIONS HAVE PROCOPE2416
     1DUCED THE SAME DESIGN//5X,23HOPTIMIZATION TERMINATED)             COPE2417
  550 FORMAT (/5X,22HAPPROXIMATING FUNCTION,I5,17H IS THE OBJECTIVE)    COPE2418
  560 FORMAT (/5X,51HAPPROXIMATING FUNCTIONS ASSOCIATED WITH CONSTRAINTSCOPE2419
     1)                                                                 COPE2420
  570 FORMAT (5X,10I5)                                                  COPE2421
  580 FORMAT (//5X,63HDESIGN VARIABLE NUMBERS ASSOCIATED WITH APPROXIMATCOPE2422
     1ING VARIABLES)                                                    COPE2423
  590 FORMAT (//5X,59H* * LEAST SQUARES FIT TO APPROXIMATION DATA IS SINCOPE2424
     1GULAR * */5X,24HRESULTS MAY NOT BE VALID)                         COPE2425
  600 FORMAT (/5X,15HFUNCTION NUMBER,I5,25H   GLOBAL VARIABLE NUMBER,I5/COPE2426
     15X,12HCOEFFICIENTS)                                               COPE2427
  610 FORMAT (/5X,44HSIDE CONSTRAINTS ON APPROXIMATE OPTIMIZATION//5X,12COPE2428
     1HLOWER BOUNDS)                                                    COPE2429
  620 FORMAT (/5X,12HUPPER BOUNDS)                                      COPE2430
  630 FORMAT (//5X,76HOPTIMIZATION HAS PRODUCED AN X-VECTOR WHICH IS THECOPE2431
     1 SAME AS A PREVIOUS DESIGN)                                       COPE2432
  640 FORMAT (/5X,51HTHE FOLLOWING DESIGN IS NOT THE APPROXIMATE OPTIMUMCOPE2433
     1)                                                                 COPE2434
  650 FORMAT (/5X,60HFOUR ATTEMPTS HAVE FAILED TO PRODUCE AN INDEPENDENTCOPE2435
     1 X-VECTOR/5X,52HOPTIMIZATION WILL CONTINUE WITH MOST RECENT X-VECTCOPE2436
     2OR)                                                               COPE2437
  660 FORMAT (1H1,4X,40HFINAL RESULT OF APPROXIMATE OPTIMIZATION)       COPE2438
  670 FORMAT (1H1,4X,42HAPPROXIMATE OPTIMIZATION ITERATION HISTORY)     COPE2439
  680 FORMAT (///5X,22HBEGIN ITERATION NUMBER,I5)                       COPE2440
  690 FORMAT (/5X,23HNOMINAL DESIGN NUMBER =,I5//5X,8HX-VECTOR)         COPE2441
  700 FORMAT (5X,5E13.5)                                                COPE2442
  710 FORMAT (/5X,15HFUNCTION VALUES)                                   COPE2443
  720 FORMAT (//5X,26HTAYLOR SERIES COEFFICIENTS)                       COPE2444
  730 FORMAT (//5X,35HRESULTS OF APPROXIMATE OPTIMIZATION)              COPE2445
  740 FORMAT (/5X,14HDELTA-X VECTOR)                                    COPE2446
  750 FORMAT (/5X,8HX-VECTOR)                                           COPE2447
  760 FORMAT (/5X,27HAPPPOXIMATE FUNCTION VALUES)                       COPE2448
  770 FORMAT (/5X,23HPRECISE FUNCTION VALUES)                           COPE2449
      END                                                               COPE2450
      SUBROUTINE COPE10 (XI,Y,XNOM,FNOM,NPTOT,KOUNT,BLU,IGFN,IOBJA,ISC,NCOPE2451
     1XAPRX,NF,NCONA,SGNOPT,CTMIN,CTLMIN,ISCR2,WGHT,INOM,NFMAX,JNOM)    COPE2452
      DIMENSION XI(NXAPRX,1), Y(NF,1), XNOM(1), FNOM(1), BLU(4,1), IGFN(COPE2453
     11), ISC(1), WGHT(1)                                               COPE2454
C     ******************************************************************COPE2455
C        ROUTINE TO SET UP ARRAYS FOR TAYLER SERIES EXPANSION.          COPE2456
C     ******************************************************************COPE2457
C                                                                       COPE2458
C     BY G. N. VANDERPLAATS                              JAN., 1979.    COPE2459
C     NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.                  COPE2460
C                                                                       COPE2461
      REWIND ISCR2                                                      COPE2462
      DO 10 J=1,NPTOT                                                   COPE2463
      READ (ISCR2) (XI(I,J),I=1,NXAPRX)                                 COPE2464
   10 READ (ISCR2) (Y(I,J),I=1,NF)                                      COPE2465
C     ------------------------------------------------------------------COPE2466
C                       FIND BEST NOMINAL IF REQUIRED.                  COPE2467
C     ------------------------------------------------------------------COPE2468
      IF (KOUNT.LE.1.AND.INOM.GT.0) GO TO 20                            COPE2469
      IF (KOUNT.GT.1.AND.KOUNT.LE.JNOM) GO TO 20                        COPE2470
      CALL COPE11 (NPTOT,Y,NF,INOM,BLU,NCONA,IGFN,IOBJA,SGNOPT,CTMIN,CTLCOPE2471
     1MIN,ISC)                                                          COPE2472
   20 CONTINUE                                                          COPE2473
C     ------------------------------------------------------------------COPE2474
C                          CREATE XNOM AND FNOM.                        COPE2475
C     ------------------------------------------------------------------COPE2476
      DO 30 I=1,NXAPRX                                                  COPE2477
   30 XNOM(I)=XI(I,INOM)                                                COPE2478
      DO 40 I=1,NF                                                      COPE2479
   40 FNOM(I)=Y(I,INOM)                                                 COPE2480
      NP=NPTOT-1                                                        COPE2481
      IF (INOM.EQ.NPTOT) GO TO 80                                       COPE2482
C     ------------------------------------------------------------------COPE2483
C                             SHIFT XI AND Y.                           COPE2484
C     ------------------------------------------------------------------COPE2485
      DO 70 J=INOM,NP                                                   COPE2486
      DO 50 I=1,NXAPRX                                                  COPE2487
   50 XI(I,J)=XI(I,J+1)                                                 COPE2488
      DO 60 I=1,NF                                                      COPE2489
   60 Y(I,J)=Y(I,J+1)                                                   COPE2490
   70 CONTINUE                                                          COPE2491
   80 CONTINUE                                                          COPE2492
C     ------------------------------------------------------------------COPE2493
C                REPLACE XI BY DELTA-XI AND Y BY DELTA-Y.               COPE2494
C     ------------------------------------------------------------------COPE2495
      DO 110 J=1,NP                                                     COPE2496
      DO 90 I=1,NXAPRX                                                  COPE2497
   90 XI(I,J)=XI(I,J)-XNOM(I)                                           COPE2498
      DO 100 I=1,NF                                                     COPE2499
  100 Y(I,J)=Y(I,J)-FNOM(I)                                             COPE2500
  110 CONTINUE                                                          COPE2501
C     ------------------------------------------------------------------COPE2502
C                           WEIGHTING FACTORS.                          COPE2503
C     ------------------------------------------------------------------COPE2504
      SMAX=1.0E-10                                                      COPE2505
      DO 130 J=1,NP                                                     COPE2506
      SUM=0.                                                            COPE2507
      DO 120 I=1,NXAPRX                                                 COPE2508
  120 SUM=SUM+XI(I,J)**2                                                COPE2509
      IF (SUM.GT.SMAX) SMAX=SUM                                         COPE2510
  130 WGHT(J)=SQRT(SUM)                                                 COPE2511
      SMAX=SQRT(SMAX)                                                   COPE2512
      DO 140 I=1,NP                                                     COPE2513
  140 WGHT(I)=2.-WGHT(I)/SMAX                                           COPE2514
```

```
      IF (NP.LE.NPMAX) RETURN
C     -----------------------------------------------------------------
C           REDUCE THE NUMBER OF DESIGNS TO NPMAX.
C     -----------------------------------------------------------------
      NPMX1=NPMAX+1
      NPSAV=NP
      DO 200 II=NPMX1,NPSAV
C     FIND DESIGN WITH MINIMUM WEIGHTING FACTOR.
      WMIN=WGHT(1)
      IMN=1
      DO 150 I=2,NP
      IF (WGHT(I).GE.WMIN) GO TO 150
      WMIN=WGHT(I)
      IMN=I
150   CONTINUE
      IF (IMN.EQ.NP) GO TO 190
C     SHIFT XI, Y AND WGHT.
      NPM1=NP-1
      DO 180 J=IMN,NPM1
      DO 160 I=1,NXAPRX
160   XI(I,J)=XI(I,J+1)
      DO 170 I=1,NF
170   Y(I,J)=Y(I,J+1)
180   WGHT(I)=WGHT(I+1)
190   NP=NP-1
200   CONTINUE
      RETURN
      END
      SUBROUTINE COPE11 (NPTOT,Y,NYR,INOM,BLU,NCONA,IGFN,IOBJA,SGNOPT,CT
     1MIN,CTLMIN,ISC)
      DIMENSION Y(NYR,1), BLU(4,1), IGFN(1), ISC(1)
C     *****************************************************************
C     ROUTINE TO DETERMINE NOMINAL DESIGN FOR APPROXIMATE OPTIMIZATION.
C     *****************************************************************
C
C     BY G. N. VANDERPLAATS                            JAN., 1979.
C     NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C     NOMINAL DESIGN IS THE ONE WITH LOWEST OBJECTIVE SATISFYING ALL
C     CONSTRAINTS.  IF ALL DESIGNS VIOLATE CONSTRAINTS, THE DESIGN WITH
C     THE LEAST VIOLATION IS FOUND.
C     I1 IS DESIGN WITH LOWEST MAXIMUM CONSTRAINT VALUE.
C     I2 IS THE DESIGN WITH THE LOWEST OBJECTIVE SATISFYING ALL
C     CONSTRAINTS.
      CT1=ABS(CTMIN)
      IF (CT1.LT.0.004) CT1=0.004
      CTL1=ABS(CTLMIN)
      IF (CTL1.LT.0.001) CTL1=0.001
C     -----------------------------------------------------------------
C                         FIND MAXIMUM OBJECTIVE.
C     -----------------------------------------------------------------
      OBJMAX=-Y(IOBJA,1)*SGNOPT
      DO 10 J=2,NPTOT
      OBJ=-Y(IOBJA,J)*SGNOPT
      IF (OBJ.GT.OBJMAX) OBJMAX=OBJ
10    CONTINUE
C     -----------------------------------------------------------------
C     NOW FIND DESIGN VARIABLE WITH LOWEST OBJECTIVE SATISFYING ALL
C     CONSTRAINTS AND DESIGN WITH LEAST CONSTRAINT VIOLATION.
C     -----------------------------------------------------------------
      GMAX=1.0E+20
      I1=1
      I2=0
      DO 50 J=1,NPTOT
C     OBJECTIVE.
      OBJ=-Y(IOBJA,J)*SGNOPT
C     CONSTRAINTS.
      ICON=0
      G1=-1.
      DO 30 I=1,NCONA
      II=IGFN(I)
      GG=Y(II,J)
C     LOWER BOUND.
      IF (BLU(1,I).LT.-1.0E+15) GO TO 20
      ICON=ICON+1
      CT=CT1
      IF (ISC(ICON).GT.0) CT=CTL1
      G=(BLU(1,I)-GG)/BLU(2,I)-CT
      IF (G.GT.G1) G1=G
C     UPPER BOUND.
20    IF (BLU(3,I).GT.1.0E+15) GO TO 30
      ICON=ICON+1
      CT=CT1
      IF (ISC(ICON).GT.0) CT=CTL1
      G=(GG-BLU(3,I))/BLU(4,I)-CT
      IF (G.GT.G1) G1=G
30    CONTINUE
      IF (G1.LT.0..OR.G1.GT.GMAX) GO TO 40
      I1=J
      GMAX=G1
40    IF (OBJ.GT.OBJMAX.OR.G1.GT.0.) GO TO 50
      I2=J
      OBJMAX=OBJ
50    CONTINUE
      INOM=I1
      IF (I2.GT.0) INOM=I2
      RETURN
      END
C     SUBROUTINE COPE12
C     *****************************************************************
C     ROUTINE TO PERFORM A LEAST SQUARES FIT OF AN ARBITRARY FUNCTION OF
C     NV VARIABLES.
C     Y = F(X1,X2,...,XNX) = B(1)*F(1) + B(2)*F(2) + ... + B(M)*F(M)
C     *****************************************************************
C
C     BY G. N. VANDERPLAATS                            JAN., 1979.
C     NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C     ARGUEMENTS,
C     X,Y  - INPUT ARRAYS OF OBSERVATIONS OF NP POINTS
C            X(NX,NP), Y(NF,NP)
C     NX   - NUMBER OF INDEPENDENT VARIABLES OF WHICH Y IS A FUNCTION.
C     NP   - NUMBER OF OBSERVATION POINTS.
C     NF   - NUMBER OF SEPERATE CURVE FITS BEING DONE SIMULTANEOUSLY.
C            THIS IS THE NUMBER OF SETS OF Y VALUES.
C     M    - NUMBER OF COMPONENTS OF THE FUNCTIONS TO BE FITTED.
C     B    - ARRAY OF M COEFFICIENTS OF FUNCTIONAL FIT TO DATE.
C     A    - M(M+1)/2 WORK VECTOR.
C     F    - WORK VECTOR - F(M).
C     G    - WORK VECTOR - G(NF).
C     NXR  - DIMENSIONED ROWS OF X.
C     NYR  - DIMENSIONED ROWS OF Y.
```

```
C     NBR  - DIMENSIONED ROWS OF B.                                    COPE2635
C     WGHT - ARRAY OF WEIGHTING FACTORS - WGHT(NP).                    COPE2636
C     NER  - ERROR FLAG.  IF NER.GT.0, DIAGONAL ELEMENT NER OF A IS    COPE2637
C            LESS THAN 1.0E-10.                                        COPE2638
C                                                                      COPE2639
C     USER SUPPLIED SUBROUTINE, COPE13.                                COPE2640
C     USAGE                                                            COPE2641
C     CALL COPE13(XI,F,NX,M)                                           COPE2642
C     ROUTINE TO EVALUATE COMPONENTS F(1),...,F(M) WHICH ARE TO BE     COPE2643
C     FITTED TO DATA.                                                  COPE2644
C     TO DATA.  ROUTINE EVALUATES THE FUNCTIONS FOR A SINGLE VECTOR OF COPE2645
C     XI AND STORES THE RESULTING VALUES IN VECTOR F.                  COPE2646
C     ARGUEMENTS.                                                      COPE2647
C     XI   - VECTOR OF INDEPENDENT VARIABLES AT WHICH FUNCTIONS ARE    COPE2648
C            TO BE EVALUATED.                                          COPE2649
C     F    - VECTOR OF FUNCTION VALUES.                                COPE2650
C     NX   - NUMBER OF INDEPENDENT VARIABLES OF WHICH Y IS A FUNCTION. COPE2651
C     M    - NUMBER OF FUNCTION COMPONENTS, ALSO REQUIRED DIMENSION OF FCOPE2652
C                                                                      COPE2653
      SUBROUTINE COPE12 (X,Y,NX,NP,NF,M,B,A,F,G,NXR,NYR,NBR,WGHT,NER)  COPE2654
      DIMENSION X(NXR,1), Y(NYR,1), B(NBR,1), A(1), F(1), G(1)         COPE2655
      DIMENSION WGHT(1)                                                COPE2656
      IF (NX.LE.NP) GO TO 50                                           COPE2657
C     SPECIAL CASE.  FEWER OBSERVATIONS THAN DESIGN VARIABLES.  USE    COPE2658
C     AVERAGE FINITE DIFFERENCE FOR FIRST ORDER EXPANSION.             COPE2659
      NP1=NP+1                                                         COPE2660
      AP=FLOAT(NP)                                                     COPE2661
      DO 20 I=1,NX                                                     COPE2662
      X(I,NP1)=AP                                                      COPE2663
      DO 10 J=1,NP                                                     COPE2664
      IF (ABS(X(I,J)).GT.1.0E-10) GO TO 10                             COPE2665
      X(I,NP1)=X(I,NP1)-1.                                             COPE2666
      X(I,J)=1.0E+20                                                   COPE2667
   10 CONTINUE                                                         COPE2668
      IF (X(I,NP1).LT.1.) X(I,NP1)=1.                                  COPE2669
   20 CONTINUE                                                         COPE2670
      DO 40 I=1,NX                                                     COPE2671
      DO 40 J=1,NF                                                     COPE2672
      B(I,J)=0.                                                        COPE2673
      DO 30 K=1,NP                                                     COPE2674
   30 B(I,J)=B(I,J)+Y(J,K)/X(I,K)                                      COPE2675
   40 B(I,J)=B(I,J)/X(I,NP1)                                           COPE2676
      NER=0                                                            COPE2677
      RETURN                                                           COPE2678
   50 CONTINUE                                                         COPE2679
C     GENERAL CASE.  DO LEAST SQUARES FIT.                             COPE2680
C     A=B=0.                                                           COPE2681
      DO 60 J=1,NF                                                     COPE2682
      DO 60 I=1,M                                                      COPE2683
   60 B(I,J)=0.                                                        COPE2684
      L=(M*(M+1))/2                                                    COPE2685
      DO 70 J=1,L                                                      COPE2686
   70 A(J)=0.                                                          COPE2687
C     LOWER TRIANGLE OF A IN SYMMETRIC MODE.                           COPE2688
      DO 100 K=1,NP                                                    COPE2689
      WGHTK=WGHT(K)                                                    COPE2690
      CALL COPE13 (X(1,K),F,NX,M)                                      COPE2691
      L=0                                                              COPE2692
      DO 80 J=1,M                                                      COPE2693
      DO 80 I=1,J                                                      COPE2694
      L=L+1                                                            COPE2695
   80 A(L)=A(L)+F(I)*F(J)*WGHTK                                        COPE2696
C     Y*F                                                              COPE2697
      DO 90 L=1,NF                                                     COPE2698
      YLK=Y(L,K)*WGHTK                                                 COPE2699
      DO 90 I=1,M                                                      COPE2700
   90 B(I,L)=B(I,L)+YLK*F(I)                                           COPE2701
  100 CONTINUE                                                         COPE2702
C     SOLVE FOR B.                                                     COPE2703
      IF (M.LE.1) GO TO 200                                            COPE2704
C     LDU DECOMPOSITION.                                               COPE2705
      MM1=M-1                                                          COPE2706
      KK=0                                                             COPE2707
      DO 110 K=1,MM1                                                   COPE2708
      NER=K                                                            COPE2709
      KK=KK+K                                                          COPE2710
      IF (ABS(A(KK)).LT.1.0E-20) GO TO 220                             COPE2711
      FACT=1./A(KK)                                                    COPE2712
      A(KK)=FACT                                                       COPE2713
      KP1=K+1                                                          COPE2714
      KJ=KK                                                            COPE2715
      DO 110 J=KP1,M                                                   COPE2716
      KJ=KJ+J-1                                                        COPE2717
      GG=A(KJ)*FACT                                                    COPE2718
      KI=KK                                                            COPE2719
      IJ=KJ                                                            COPE2720
      DO 110 I=KP1,J                                                   COPE2721
      IJ=IJ+1                                                          COPE2722
      KI=KI+I-1                                                        COPE2723
      A(IJ)=A(IJ)-A(KI)*GG                                             COPE2724
  110 CONTINUE                                                         COPE2725
      KK=KK+M                                                          COPE2726
      NER=M                                                            COPE2727
      IF (ABS(A(KK)).LT.1.0E-20) GO TO 220                             COPE2728
      A(KK)=1./A(KK)                                                   COPE2729
C     FORWARD SUBSTITUTION                                             COPE2730
      MP1=M+1                                                          COPE2731
      KK=0                                                             COPE2732
      DO 130 K=1,MM1                                                   COPE2733
      KP1=K+1                                                          COPE2734
      KK=KK+K                                                          COPE2735
      AKK=A(KK)                                                        COPE2736
      DO 120 L=1,NF                                                    COPE2737
  120 B(K,L)=B(K,L)*AKK                                                COPE2738
      KI=KK                                                            COPE2739
      DO 130 I=KP1,M                                                   COPE2740
      KI=KI+I-1                                                        COPE2741
      AKI=A(KI)                                                        COPE2742
      DO 130 J=1,NF                                                    COPE2743
      B(I,J)=B(I,J)-AKI*B(K,J)                                         COPE2744
  130 CONTINUE                                                         COPE2745
      KK=KK+M                                                          COPE2746
      AKK=A(KK)                                                        COPE2747
      DO 140 J=1,NF                                                    COPE2748
  140 B(M,J)=B(M,J)*AKK                                                COPE2749
C     BACK SUBSTITUTION.                                               COPE2750
      DO 190 I=2,M                                                     COPE2751
      J=MP1-I                                                          COPE2752
      JJ=J*(J+1)/2                                                     COPE2753
      JK=JJ                                                            COPE2754
```

```
      JP1=J+1
      DO 150 L=1,NF
150   G(L)=0.
      DO 170 K=JP1,M
      JK=JK+K-1
      AJK=A(JK)
      DO 160 L=1,NF
160   G(L)=G(L)+AJK*B(K,L)
170   CONTINUE
      AJJ=A(JJ)
      DO 180 L=1,NF
180   B(J,L)=B(J,L)-AJJ*G(L)
190   CONTINUE
      NER=0
      RETURN
200   CONTINUE
      AKK=A(1)
      NER=1
      IF (ABS(AKK).LT.1.0E-20) GO TO 220
      AKK=1./AKK
      DO 210 J=1,NF
210   B(1,J)=B(1,J)*AKK
      NER=0
      RETURN
220   CONTINUE
      RETURN
      END
      SUBROUTINE COPE13 (X,F,NX,M)
      DIMENSION X(1), F(1)
C     *****************************************************************
C     ROUTINE TO CALCULATE F VALUES FOR LEAST SQUARES FIT TO QUADRATIC
C     Y-Y0 ■ DY-TRANSPOSE TIMES X + (1/2 X-TRANSPOSE TIMES H TIMES X.
C     *****************************************************************
C
C     BY G. N. VANDERPLAATS                           JAN., 1979.
C     X CONTAINS X-X0.
C     M = MAXIMUM NUMBER OF COEFFICIENTS TO BE CALCULATED.
C         M .LE. NX + (NX+1)/2.
C     DY COEF.
      DO 10 I=1,NX
10    F(I)=X(I)
C     H COEF. = X1*X1, X2*X2, ... XN*XN,  X1*X2... X1*XN...
      II=NX
C     -----------------------------------------------------------------
C                          DIAGONAL ELEMENTS.
C     -----------------------------------------------------------------
      DO 20 I=1,NX
      II=II+1
      IF (II.GT.M) GO TO 40
20    F(II)=.5*(X(I)**2)
      IF (NX.LT.2) RETURN
C     -----------------------------------------------------------------
C                        OFF-DIAGONAL ELEMENTS.
C     -----------------------------------------------------------------
      NXM1=NX-1
      DO 30 I=1,NXM1
      IP1=I+1
      DO 30 J=IP1,NX
      II=II+1
      IF (II.GT.M) GO TO 40
30    F(II)=X(I)*X(J)
40    CONTINUE
      RETURN
      END
      SUBROUTINE COPE14 (NXAPRX,NF,NPTOT,RA,IA,LOCR,LOCI,TITLE,INOM,NDV,
     1IPAPRX,ISCR2,MAXTRM)
      DIMENSION RA(1), IA(1), LOCR(1), LOCI(1), TITLE(1)
C     *****************************************************************
C     ROUTINE TO PRINT RESULTS OF APPROXIMATE ANALYSIS/OPTIMIZATION.
C     *****************************************************************
C
C     BY G. N. VANDERPLAATS                           JAN., 1979.
C     NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C
C     -----------------------------------------------------------------
C                                TITLE.
C     -----------------------------------------------------------------
      WRITE (6,90) (TITLE(I),I=1,20)
C     -----------------------------------------------------------------
C                     GLOBAL LOCATION OF X AND F(X)
C     -----------------------------------------------------------------
C     GLOBAL LOCATIONS OF X.
      M5=LOCI(5)
      MM5=M5+NXAPRX-1
      WRITE (6,100)
      WRITE (6,110) (IA(I),I=M5,MM5)
C     GLOBAL LOCATIONS OF F(X).
      M6=LOCI(6)
      MM6=M6+NF-1
      WRITE (6,140)
      WRITE (6,110) (IA(I),I=M6,MM6)
C     -----------------------------------------------------------------
C                     X-VALUES AND FUNCTIONS, F(X)
C     -----------------------------------------------------------------
C     X-VALUES.
      N1=LOCR(23)+3*NXAPRX+6
      N2=N1+NXAPRX-1
      WRITE (6,120) NPTOT,INOM
      WRITE (6,130) (RA(I),I=N1,N2)
C     F(X) VALUES.
      N1=N1+NXAPRX+NDV
      N2=N1+NF-1
      WRITE (6,150)
      WRITE (6,130) (RA(I),I=N1,N2)
C     -----------------------------------------------------------------
C                        TAYLER SERIES COEFFICIENTS.
C     -----------------------------------------------------------------
      WRITE (6,170)
      NP=NPTOT-1
      NBTAY=N1+2*NF
      NBR=NXAPRX+(NXAPRX*(NXAPRX+1))/2
      IF (MAXTRM.LT.3) NBR=MAXTRM*NXAPRX
      DO 30 JJ=1,NF
      M6=LOCI(6)+JJ-1
      M6=IA(M6)
      WRITE (6,80) JJ,M6
C     LINEAR TERMS.
      N2=NXAPRX
      IF (NP.LT.N2) N2=NP
      N2=N2+NBTAY-1
```

```
      WRITE (6,160)
      WRITE (6,130) (RA(I),I=NBTAY,N2)
      IF (NP.LE.NXAPRX) GO TO 20
      IF (MAXTRM.LT.2) GO TO 20
C     NON-LINEAR TERMS.
C     N1 = LOCATION OF FIRST DIAGONAL ELEMENT.
      N1=N2+1
C     N2 = LOCATION OF LAST DIAGONAL ELEMENT.
      N2=NXAPRX
      IF (N2.GT.NP) N2=NP
      N2=N2+N1-1
C     N3=LOCATION OF FIRST OFF-DIAGONAL ELEMENT.
      N3=N2+1
C     N4=LOCATION OF LAST OFF-DIAGONAL ELEMENT.
      N4=NBR
      IF (N4.GT.NP) N4=NP
      N4=N4-2*NXAPRX+N3-1
C     LL = LOCATION OF LAST OFF-DIAGONAL ELEMENT - THIS ROW.
      WRITE (6,190)
      II=1
      DO 10 I=N1,N2
      WRITE (6,180) II
      II=II+1
      LL=N3+NXAPRX-II
      IF (LL.GT.N4) LL=N4
      IF (LL.LT.N3) WRITE (6,130) RA(I)
      IF (LL.GE.N3) WRITE (6,130) RA(I),(RA(J),J=N3,LL)
10    N3=LL+1
20    CONTINUE
      NBTAY=NBTAY+NBR
30    CONTINUE
      IF (IPAPRX.LT.3) RETURN
      REWIND ISCR2
      WRITE (6,50)
      DO 40 I=1,NPTOT
C     X-VECTOR.
      READ (ISCR2) (RA(J),J=1,NXAPRX)
      WRITE (6,60) I
      WRITE (6,130) (RA(J),J=1,NXAPRX)
C     FUNCTION VALUES.
      READ (ISCR2) (RA(J),J=1,NF)
      WRITE (6,70)
      WRITE (6,130) (RA(J),J=1,NF)
40    CONTINUE
      RETURN
C     ------------------------------------------------------------------
C                                  FORMATS
C     ------------------------------------------------------------------
50    FORMAT (///5X,18HSUMMARY OF DESIGNS)
60    FORMAT (/5X,13HDESIGN NUMBER,I5/5X,8HX-VECTOR)
70    FORMAT (/5X,15HFUNCTION VALUES)
80    FORMAT (//5X,9HPARAMETER,I5,18H = GLOBAL VARIABLE,I5)
90    FORMAT (1H1,4X,44HRESULTS OF APPROXIMATE ANALYSIS/OPTIMIZATION///5
     1X,5HTITLE/5X,20A4)
100   FORMAT (///5X,31HGLOBAL LOCATIONS OF X-VARIABLES)
110   FORMAT (5X,10I5)
120   FORMAT (//5X,25HAPPROXIMATION IS BASED ON,I5,8H DESIGNS//5X,31HNOM
     1INAL DESIGN IS DESIGN NUMBER,I5//5X,21HVALUES OF X-VARIABLES)
130   FORMAT (5X,5E13.4)
140   FORMAT (//5X,35HGLOBAL LOCATIONS OF FUNCTIONS, F(X))
150   FORMAT (//5X,25HVALUES OF FUNCTIONS, F(X))
160   FORMAT (/5X,19HLINEAR TERMS, DEL F)
170   FORMAT (///5X,39HCOEFFICIENTS OF TAYLOR SERIES EXPANSION)
180   FORMAT (/5X,3HROW,I5)
190   FORMAT (//5X,51HNON-LINEAR TERMS, H, BEGINING WITH DIAGONAL ELEMEN
     1T)
      END
      SUBROUTINE COPE15 (XV,G,DF,A,ISC,IC,NN1,BLU,NX1,IOBJA,M,FNOM,FNEW,
     1BTAY,NBR,IGFN,CT,CTL,INFO,NAC,NCONA,NDV,NF,OBJ,SGNOPT)
      DIMENSION XV(1), FNOM(1), FNEW(1), A(NN1,1), BTAY(NBR,1), DF(1), I
     1GFN(1), ISC(1), IC(1), G(1), BLU(4,1)
C     ******************************************************************
C            FUNCTION EVALUATION FOR APPROXIMATE OPTIMIZATION.
C     ******************************************************************
C     BY G. N. VANDERPLAATS                                 JAN., 1979.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C
C     ------------------------------------------------------------------
C                                OBJECTIVE.
C     ------------------------------------------------------------------
      CALL COPE16 (NX1,XV,NF,FNOM,FNEW,BTAY,NBR,M)
      OBJ=-FNEW(IOBJA)*SGNOPT
      IF (INFO.EQ.1) GO TO 20
C     GRADIENT OF OBJECTIVE.
      CALL COPE17 (NX1,XV,IOBJA,BTAY,NBR,M,DF)
      DO 10 I=1,NDV
10    DF(I)=-DF(I)*SGNOPT
20    CONTINUE
      IF (NCONA.LE.0) GO TO 80
C     ------------------------------------------------------------------
C                                CONSTRAINTS.
C     ------------------------------------------------------------------
      IF (INFO.EQ.2) NAC=0
      ICON=0
      DO 70 I=1,NCONA
      J=IGFN(I)
      GG=FNEW(J)
C     LOWER BOUND.
      IF (BLU(1,I).LT.-1.0E+15) GO TO 40
      ICON=ICON+1
      G(ICON)=(BLU(1,I)-GG)/BLU(2,I)
      IF (INFO.EQ.1) GO TO 40
C     IS THIS CONSTRAINT ACTIVE OR VIOLATED.
      CTI=CT
      IF (ISC(ICON).GT.0) CTI=CTL
      IF (G(ICON).LT.CTI) GO TO 40
C     ACTIVE CONSTRAINT. CALCULATE GRADIENT.
      NAC=NAC+1
      IC(NAC)=ICON
      MM=M
      IF (ISC(ICON).GT.0) MM=NDV
      CALL COPE17 (NX1,XV,J,BTAY,NBR,MM,A(1,NAC))
      FF=1./BLU(2,I)
      DO 30 K=1,NDV
30    A(K,NAC)=-A(K,NAC)*FF
40    IF (BLU(3,I).GT.1.0E+15) GO TO 60
      ICON=ICON+1
      G(ICON)=(GG-BLU(3,I))/BLU(4,I)
      IF (INFO.EQ.1) GO TO 60
C     IS THIS CONSTRAINT ACTIVE OR VIOLATED.
```

```
      CTI=CT
      IF (ISC(ICON).GT.0) CTI=CTL
      IF (G(ICON).LT.CTI) GO TO 60
C     ACTIVE CONSTRAINT. CALCULATE GRADIENT.
      NAC=NAC+1
      IC(NAC)=ICON
      MM=M
      IF (ISC(ICON).GT.0) MM=NDV
      CALL COPE17 (NX1,XV,J,BTAY,NBR,MM,A(1,NAC))
      FF=1./BLU(4,I)
      DO 50 K=1,NDV
50    A(K,NAC)=A(K,NAC)*FF
60    CONTINUE
70    CONTINUE
80    CONTINUE
      RETURN
      END
      SUBROUTINE COPE16 (NX,X,NF,FNOM,FNEW,B,NBR,M)
C     ******************************************************************
C     ROUTINE TO EVALUATE FUNCTIONS APPROXIMATED BY TAYLER SERIES
C     EXPANSION UP TO SECOND ORDER.
C     ******************************************************************
C     BY G. N. VANDERPLAATS                           MAR., 1978.
C     NAVAL SHIP R AND D CENTER.
C
C     F = F0 + DELF TIMES X + X-TRANSPOSE TIMES DEL2F TIMES X.
C     ARGUEMENTS.
C     NX   - NUMBER OF INDEPENDENT VARIABLES CONTAINED IN X.
C     X    - VECTOR OF DELTA VARIABLES X-XNOM.  DIMENSIONED X(NX)
C     NF   - NUMBER OF FUNCTIONS TO BE EVALUATED.
C     FNOM - NOMINAL FUNCTION VALUES ABOUT WHICH TAYLER SERIES EXPANSION
C     WAS DONE.
C     FNEW - NEW APPROXIMATED VALUES. - OUTPUT.  DIMENSIONED FNEW(NF)
C     B    - MATRIX OF TAYLER SERIES COEFFICIENTS.
C            B(I,J) CONTAINS DEL F, I=1,NX.
C            B(NX+I,J) CONTAINS DEL2 TERMS,  I = 1,NX*(NX+1)/2.
C            MINIMUM DIMENSIONS - B(M,NF).
C     NBR  - DIMENSIONED ROWS OF B.
C     M    - TOTAL NUMBER OF COEFFICIENTS CURRENTLY USED.
C
      DIMENSION X(1), FNOM(1), FNEW(1), B(NBR,1)
      DO 50 J=1,NF
C     ------------------------------------------------------------------
C                              CONSTANT TERM.
C     ------------------------------------------------------------------
      F=FNOM(J)
C     ------------------------------------------------------------------
C                          FIRST ORDER TERMS.
C     ------------------------------------------------------------------
      DO 10 I=1,NX
      IF (I.GT.M) GO TO 40
10    F=F+B(I,J)*X(I)
C     ------------------------------------------------------------------
C                          SECOND ORDER TERMS.
C     ------------------------------------------------------------------
C
C     ------------------------------------------------------------------
C                          DIAGONAL ELEMENTS.
C     ------------------------------------------------------------------
      II=NX
      DO 20 I=1,NX
      II=II+1
      IF (II.GT.M) GO TO 40
20    F=F+.5*B(II,J)*(X(I)**2)
C     ------------------------------------------------------------------
C                        OFF-DIAGONAL ELEMENTS.
C     ------------------------------------------------------------------
      IF (NX.LT.2) GO TO 40
      NXM1=NX-1
      DO 30 I=1,NXM1
      IP1=I+1
      XX=X(I)
      DO 30 K=IP1,NX
      II=II+1
      IF (II.GT.M) GO TO 40
30    F=F+B(II,J)*XX*X(K)
40    CONTINUE
50    FNEW(J)=F
      RETURN
      END
      SUBROUTINE COPE17 (NX,X,J,B,NBR,M,GRAD)
C     ******************************************************************
C     ROUTINE TO CALCULATE GRADIENT OF THE J-TH FUNCTION APPROXIMATED
C     BY TAYLER SERIES EXPANSION UP TO SECOND ORDER.
C     ******************************************************************
C     BY G. N. VANDERPLAATS                           APRIL, 1978.
C     NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C
C     F = F0 + DELF TIMES X + .5 X-TRANSPOSE TIMES DEL2F TIMES X.
C     AGRUEMENTS.
C     NX   - NUMBER OF INDEPENDENT VARIABLES CONTAINED IN X.
C     X    - VECTOR OF DELTA VARIABLES X-XNOM.  DIMENSIONED X(NX).
C     J    - FUNCTION FOR WHICH GRADIENT INFORMATION IS CALCULATED.
C     B    - MATRIX OF TAYLER SERIES COEFFICIENTS.
C            B(I,J) CONTAINS DELF, I = 1,NX.
C             B(NX+I,J) CONTAINS DEL2 TERMS, I = 1,NS*(NX+1)/2.
C             MINIMUM DIMENSIONS - B(M,NF).
C     NBR  - DIMENSIONED ROWS OF B.
C     M    - TOTAL NUMBER OF COEFFICIENTS CURRENTLY USED.
C     GRAD - GRADIENT OF J-TH FUNCTION.  OUTPUT.
C
      DIMENSION X(1), B(NBR,1), GRAD(1)
C     ------------------------------------------------------------------
C                          FIRST ORDER TERMS.
C     ------------------------------------------------------------------
      DO 10 I=1,NX
      IF (I.GT.M) GO TO 40
10    GRAD(I)=B(I,J)
C     ------------------------------------------------------------------
C                          SECOND ORDER TERMS.
C     ------------------------------------------------------------------
      II=NX
C     ------------------------------------------------------------------
C                          DIAGONAL ELEMENTS.
C     ------------------------------------------------------------------
      DO 20 I=1,NX
      II=II+1
      IF (II.GT.M) GO TO 40
20    GRAD(I)=GRAD(I)+B(II,J)*X(I)
C     ------------------------------------------------------------------
```

```
C                      OFF-DIAGONAL ELEMENTS.
C     -----------------------------------------------------------------
      IF (NX.LT.2) GO TO 40
      NXM1=NX-1
      DO 30 I=1,NXM1
      IP1=I+1
      DO 30 K=IP1,NX
      II=II+1
      IF (II.GT.M) GO TO 40
      GRAD(I)=GRAD(I)+B(II,J)*X(K)
30    GRAD(K)=GRAD(K)+B(II,J)*X(I)
40    CONTINUE
      RETURN
      END
      SUBROUTINE COPE18 (IOBJ,NDVTOT,NCONA,RA,IA,LOCR,LOCI,ARRAY)
C     *****************************************************************
C               ROUTINE TO PRINT OPTIMIZATION RESULTS
C     *****************************************************************
C     BY G. N. VANDERPLAATS                             MAR., 1979
C     NASA AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
      DIMENSION RA(1), IA(1), LOCR(1), LOCI(1), ARRAY(1)
C     OBJECTIVE FUNCTION AND DESIGN VARIABLES.
      WRITE (6,30) IOBJ,ARRAY(IOBJ)
      N2=LOCR(2)
      N3=LOCR(3)
      N5=LOCR(5)
      M2=LOCI(2)
      DO 10 I=1,NDVTOT
C     DESIGN VARIABLE NUMBER.
      IDV=IA(M2)
      N2=LOCR(2)+IDV-1
      N3=LOCR(3)+IDV-1
      M2=M2+1
C     GLOBAL LOCATION.
      IG=IA(I)
C     MULTIPLIER.
      AMULT=RA(N5)
      N5=N5+1
C     LOWER BOUND.
      BL=AMULT*RA(N2)
C     VALUE.
      XX=ARRAY(IG)
C     UPPER BOUND.
      BU=AMULT*RA(N3)
      WRITE (6,40) I,IDV,IG,BL,XX,BU
10    CONTINUE
      IF (NCONA.EQ.0) RETURN
C     CONSTRAINTS.
      WRITE (6,50)
      M3=LOCI(3)
      N6=LOCR(6)
      ID=1
      IF (RA(N6).LT.-1.0E+15.AND.RA(N6+2).GT.1.0E+15) ID=0
      DO 20 I=1,NCONA
C     GLOBAL LOCATION.
      IG=IA(M3)
      M3=M3+1
C     LOWER BOUND.
      BL=RA(N6)
C     VALUE.
      XX=ARRAY(IG)
C     UPPER BOUND.
      BU=RA(N6+2)
      N6=N6+4
C     IDENTIFICATION NUMBER.
      JD=ID
      IF (BL.GT.-1.0E+15) ID=ID+1
      IF (BU.LT.1.0E+15) ID=ID+1
      WRITE (6,60) JD,IG,BL,XX,BU
20    CONTINUE
      RETURN
C     -----------------------------------------------------------------
C                              FORMATS
C     -----------------------------------------------------------------
30    FORMAT (1H1,4X,20HOPTIMIZATION RESULTS///5X,18HOBJECTIVE FUNCTION/
     15X,15HGLOBAL LOCATION,I5,5X,14HFUNCTION VALUE,E12.5///5X,16HDESIGN
     2 VARIABLES//14X,5HD. V.,5X,6HGLOBAL,7X,5HLOWER,23X,5HUPPER/8X,2HID
     3,5X,3HNO.,5X,8HVAR. NO.,6X,5HBOUND,9X,5HVALUE,9X,5HBOUND)
40    FORMAT (I10,I7,I11,3X,3E14.5)
50    FORMAT (///5X,18HDESIGN CONSTRAINTS//15X,6HGLOBAL,7X,5HLOWER,23X,5
     1HUPPER/9X,2HID,4X,8HVAR. NO.,6X,5HBOUND,9X,5HVALUE,9X,5HBOUND)
60    FORMAT (I10,I9,3X,3E14.5)
      END
//
```

```
      SUBROUTINE CONMIN (X,VLB,VUB,G,SCAL,DF,A,S,G1,G2,B,C,ISC,IC,MS1,N1CONMIN 2
     1,N2,N3,N4,N5)                                                    CONMIN 3
      COMMON /CNMN1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAXCONMIN 4
     1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,ITCONMIN 5
     2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER                              CONMIN 6
      DIMENSION X(N1), VLB(N1), VUB(N1), G(N2), SCAL(N1), DF(N1), A(N1,NCONMIN 7
     13), S(N1), G1(N2), G2(N2), B(N3,N3), C(N4), ISC(N2), IC(N3), MS1(NCONMIN 8
     25)                                                               CONMIN 9
      COMMON /CONSAV/ DM1,DM2,DM3,DM4,DM5,DM6,DM7,DM8,DM9,DM10,DM11,DM12CONMIN10
     1,DCT,DCTL,PHI,ABOBJ,CTA,CTAM,CTBM,OBJ1,SLOPE,DX,DX1,FI,XI,DFTDF1,ACONMIN11
     2LP,FFF,A1,A2,A3,A4,F1,F2,F3,F4,CV1,CV2,CV3,CV4,APP,ALPCA,ALPFES,ALCONMIN12
     3PLN,ALPMIN,ALPNC,ALPSAV,ALPSID,ALPTOT,RSPACE,IDM1,IDM2,IDM3,JDIR,ICONMIN13
     4OBJ,KOBJ,KCOUNT,NCAL(2),NFEAS,MSCAL,NCOBJ,NVC,KOUNT,ICOUNT,IGOOD1,CONMIN14
     5IGOOD2,IGOOD3,IGOOD4,IBEST,III,NLNC,JGOTO,ISPACE(2)               CONMIN15
C     ROUTINE TO SOLVE CONSTRAINED OR UNCONSTRAINED FUNCTION            CONMIN16
C     MINIMIZATION.                                                     CONMIN17
C     BY G. N. VANDERPLAATS                          APRIL, 1972.       CONMIN18
C     * * * * * * * * * * *   JUNE, 1979 VERSION   * * * * * * * * * * *CONMIN19
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.                  CONMIN20
C     REFERENCE;  CONMIN - A FORTRAN PROGRAM FOR CONSTRAINED FUNCTION   CONMIN21
C         MINIMIZATION:  USER'S MANUAL,  BY G. N. VANDERPLAATS,         CONMIN22
C         NASA TM X-62,282, AUGUST, 1973.                               CONMIN23
C     STORAGE REQUIREMENTS:                                             CONMIN24
C         PROGRAM - 7000 DECIMAL WORDS (CDC COMPUTER)                   CONMIN25
C         ARRAYS  - APPROX. 2*(NDV**2)+26*NDV+4*NCON,                   CONMIN26
C                WHERE N3 = NDV+2.                                      CONMIN27
C     RE-SCALE VARIABLES IF REQUIRED.                                   CONMIN28
      IF (NSCAL.EQ.0.OR.IGOTO.EQ.0) GO TO 20                            CONMIN29
      DO 10 I=1,NDV                                                     CONMIN30
10    X(I)=C(I)                                                         CONMIN31
20    CONTINUE                                                          CONMIN32
C     CONSTANTS.                                                        CONMIN33
      NDV1=NDV+1                                                        CONMIN34
      NDV2=NDV+2                                                        CONMIN35
      IF (IGOTO.EQ.0) GO TO 40                                          CONMIN36
C     ------------------------------------------------------------------CONMIN37
C                   CHECK FOR UNBOUNDED SOLUTION                        CONMIN38
C     ------------------------------------------------------------------CONMIN39
C     STOP IF OBJ IS LESS THAN -1.0E+40                                 CONMIN40
      IF (OBJ.GT.-1.0E+40) GO TO 30                                     CONMIN41
      WRITE (6,980)                                                     CONMIN42
      GO TO 810                                                         CONMIN43
30    CONTINUE                                                          CONMIN44
      GO TO (160,390,380,670,690),IGOTO                                 CONMIN45
C     ------------------------------------------------------------------CONMIN46
C                    SAVE INPUT CONTROL PARAMETERS                      CONMIN47
C     ------------------------------------------------------------------CONMIN48
40    CONTINUE                                                          CONMIN49
      IF (IPRINT.GT.0) WRITE (6,1220)                                   CONMIN50
      IF (LINOBJ.EQ.0.OR.(NCON.GT.0.OR.NSIDE.GT.0)) GO TO 50            CONMIN51
C     TOTALLY UNCONSTRAINED FUNCTION WITH LINEAR OBJECTIVE.             CONMIN52
C     SOLUTION IS UNBOUNDED.                                            CONMIN53
      WRITE (6,970) LINOBJ,NCON,NSIDE                                   CONMIN54
      RETURN                                                            CONMIN55
50    CONTINUE                                                          CONMIN56
      IDM1=ITRM                                                         CONMIN57
      IDM2=ITMAX                                                        CONMIN58
      IDM3=ICNDIR                                                       CONMIN59
      DM1=DELFUN                                                        CONMIN60
      DM2=DABFUN                                                        CONMIN61
      DM3=CT                                                            CONMIN62
      DM4=CTMIN                                                         CONMIN63
      DM5=CTL                                                           CONMIN64
      DM6=CTLMIN                                                        CONMIN65
      DM7=THETA                                                         CONMIN66
      DM8=PHI                                                           CONMIN67
      DM9=FDCH                                                          CONMIN68
      DM10=FDCHM                                                        CONMIN69
      DM11=ABOBJ1                                                       CONMIN70
      DM12=ALPHAX                                                       CONMIN71
C     ------------------------------------------------------------------CONMIN72
C                                DEFAULTS                               CONMIN73
C     ------------------------------------------------------------------CONMIN74
      IF (ITRM.LE.0) ITRM=3                                             CONMIN75
      IF (ITMAX.LE.0) ITMAX=20                                          CONMIN76
      NDV1=NDV+1                                                        CONMIN77
      IF (ICNDIR.EQ.0) ICNDIR=NDV1                                      CONMIN78
      IF (DELFUN.LE.0.) DELFUN=.0001                                    CONMIN79
      CT=-ABS(CT)                                                       CONMIN80
      IF (CT.GE.0.) CT=-.1                                              CONMIN81
      CTMIN=ABS(CTMIN)                                                  CONMIN82
      IF (CTMIN.LE.0.) CTMIN=.004                                       CONMIN83
      CTL=-ABS(CTL)                                                     CONMIN84
      IF (CTL.GE.0.) CTL=-0.01                                          CONMIN85
      CTLMIN=ABS(CTLMIN)                                                CONMIN86
      IF (CTLMIN.LE.0.) CTLMIN=.001                                     CONMIN87
      IF (THETA.LE.0.) THETA=1.                                         CONMIN88
      IF (ABOBJ1.LE.0.) ABOBJ1=.1                                       CONMIN89
      IF (ALPHAX.LE.0.) ALPHAX=.1                                       CONMIN90
      IF (FDCH.LE.0.) FDCH=.01                                          CONMIN91
      IF (FDCHM.LE.0.) FDCHM=.01                                        CONMIN92
C     ------------------------------------------------------------------CONMIN93
C                     INITIALIZE INTERNAL PARAMETERS                    CONMIN94
C     ------------------------------------------------------------------CONMIN95
      INFOG=0                                                           CONMIN96
      ITER=0                                                            CONMIN97
      JDIR=0                                                            CONMIN98
      IOBJ=0                                                            CONMIN99
      KOBJ=0                                                            CONMI100
      NDV2=NDV+2                                                        CONMI101
      KCOUNT=0                                                          CONMI102
      NCAL(1)=0                                                         CONMI103
      NCAL(2)=0                                                         CONMI104
      NAC=0                                                             CONMI105
      NFEAS=0                                                           CONMI106
      MSCAL=NSCAL                                                       CONMI107
      CT1=ITRM                                                          CONMI108
      CT1=1./CT1                                                        CONMI109
      DCT=(CTMIN/ABS(CT))**CT1                                          CONMI110
      DCTL=(CTLMIN/ABS(CTL))**CT1                                       CONMI111
      PHI=5.                                                            CONMI112
      ABOBJ=ABOBJ1                                                      CONMI113
      NCOBJ=0                                                           CONMI114
      CTAM=ABS(CTMIN)                                                   CONMI115
      CTBM=ABS(CTLMIN)                                                  CONMI116
C     CALCULATE NUMBER OF LINEAR CONSTRAINTS, NLNC.                     CONMI117
      NLNC=0                                                            CONMI118
      IF (NCON.EQ.0) GO TO 70                                           CONMI119
      DO 60 I=1,NCON                                                    CONMI120
      IF (ISC(I).GT.0) NLNC=NLNC+1                                      CONMI121
```

```
60    CONTINUE
70    CONTINUE
C     ------------------------------------------------------------------
C          CHECK TO BE SURE THAT SIDE CONSTRAINTS ARE SATISFIED
C     ------------------------------------------------------------------
      IF (NSIDE.EQ.0) GO TO 110
      DO 100 I=1,NDV
      IF (VLB(I).LE.VUB(I)) GO TO 80
      XX=.5*(VLB(I)+VUB(I))
      X(I)=XX
      VLB(I)=XX
      VUB(I)=XX
      WRITE (6,1120) I
80    CONTINUE
      XX=X(I)-VLB(I)
      IF (XX.GE.0.) GO TO 90
C     LOWER BOUND VIOLATED.
      WRITE (6,1130) X(I),VLB(I),I
      X(I)=VLB(I)
      GO TO 100
90    CONTINUE
      XX=VUB(I)-X(I)
      IF (XX.GE.0.) GO TO 100
      WRITE (6,1140) X(I),VUB(I),I
      X(I)=VUB(I)
100   CONTINUE
110   CONTINUE
C     ------------------------------------------------------------------
C                       INITIALIZE SCALING VECTOR, SCAL
C     ------------------------------------------------------------------
      IF (NSCAL.EQ.0) GO TO 150
      IF (NSCAL.LT.0) GO TO 130
      DO 120 I=1,NDV
120   SCAL(I)=1.
      GO TO 150
130   CONTINUE
      DO 140 I=1,NDV
      SI=ABS(SCAL(I))
      IF (SI.LT.1.0E-20) SI=1.0E-5
      SCAL(I)=SI
      SI=1./SI
      X(I)=X(I)*SI
      IF (NSIDE.EQ.0) GO TO 140
      VLB(I)=VLB(I)*SI
      VUB(I)=VUB(I)*SI
140   CONTINUE
150   CONTINUE
C     ------------------------------------------------------------------
C     ***** CALCULATE INITIAL FUNCTION AND CONSTRAINT VALUES  *****
C     ------------------------------------------------------------------
      INFO=1
      NCAL(1)=1
      IGOTO=1
      GO TO 950
160   CONTINUE
      OBJ1=OBJ
      IF (DABFUN.LE.0.) DABFUN=.001*ABS(OBJ)
      IF (DABFUN.LT.1.0E-10) DABFUN=1.0E-10
      IF (IPRINT.LE.0) GO TO 270
C     ------------------------------------------------------------------
C                      PRINT INITIAL DESIGN INFORMATION
C     ------------------------------------------------------------------
      IF (IPRINT.LE.1) GO TO 230
      IF (NSIDE.EQ.0.AND.NCON.EQ.0) WRITE (6,1290)
      IF (NSIDE.NE.0.OR.NCON.GT.0) WRITE (6,1230)
      WRITE (6,1240) IPRINT,NDV,ITMAX,NCON,NSIDE,ICNDIR,NSCAL,NFDG,LINOB
     1J,ITRM,N1,N2,N3,N4,N5
      WRITE (6,1260) CT,CTMIN,CTL,CTLMIN,THETA,PHI,DELFUN,DABFUN
      WRITE (6,1250) FDCH,FDCHM,ALPHAX,ABOBJ1
      IF (NSIDE.EQ.0) GO TO 190
      WRITE (6,1270)
      DO 170 I=1,NDV,6
      M1=MIN0(NDV,I+5)
170   WRITE (6,1010) I,(VLB(J),J=I,M1)
      WRITE (6,1280)
      DO 180 I=1,NDV,6
      M1=MIN0(NDV,I+5)
180   WRITE (6,1010) I,(VUB(J),J=I,M1)
190   CONTINUE
      IF (NSCAL.GE.0) GO TO 200
      WRITE (6,1300)
      WRITE (6,1460) (SCAL(I),I=1,NDV)
200   CONTINUE
      IF (NCON.EQ.0) GO TO 230
      IF (NLNC.EQ.0.OR.NLNC.EQ.NCON) GO TO 220
      WRITE (6,1020)
      DO 210 I=1,NCON,15
      M1=MIN0(NCON,I+14)
210   WRITE (6,1030) I,(ISC(J),J=I,M1)
      GO TO 230
220   IF (NLNC.EQ.NCON) WRITE (6,1040)
      IF (NLNC.EQ.0) WRITE (6,1050)
230   CONTINUE
      WRITE (6,1440) OBJ
      WRITE (6,1450)
      DO 240 I=1,NDV
      X1=1.
      IF (NSCAL.NE.0) X1=SCAL(I)
240   G1(I)=X(I)*X1
      DO 250 I=1,NDV,6
      M1=MIN0(NDV,I+5)
250   WRITE (6,1010) I,(G1(J),J=I,M1)
      IF (NCON.EQ.0) GO TO 270
      WRITE (6,1470)
      DO 260 I=1,NCON,6
      M1=MIN0(NCON,I+5)
260   WRITE (6,1010) I,(G(J),J=I,M1)
270   CONTINUE
      IF (IPRINT.GT.1) WRITE (6,1360)
C     ------------------------------------------------------------------
C     *******************  BEGIN MINIMIZATION  **********************
C     ------------------------------------------------------------------
280   CONTINUE
      ITER=ITER+1
      IF (ABOBJ1.LT..0001) ABOBJ1=.0001
      IF (ABOBJ1.GT..2) ABOBJ1=.2
      IF (ALPHAX.GT.1.) ALPHAX=1.
      IF (ALPHAX.LT..001) ALPHAX=.001
      IF (IPRINT.GT.2) WRITE (6,1310) ITER
      IF (IPRINT.GT.3.AND.NCON.GT.0) WRITE (6,1320) CT,CTL,PHI
```

```
      CTA=ABS(CT)                                                       CONMI242
      IF (NCOBJ.EQ.0) GO TO 340                                         CONMI243
C     ------------------------------------------------------------------CONMI244
C     NO MOVE ON LAST ITERATION.  DELETE CONSTRAINTS THAT ARE NO        CONMI245
C     LONGER ACTIVE.                                                    CONMI246
C     ------------------------------------------------------------------CONMI247
      NNAC=NAC                                                          CONMI248
      DO 290 I=1,NNAC                                                   CONMI249
      IF (IC(I).GT.NCON) NAC=NAC-1                                      CONMI250
290   CONTINUE                                                          CONMI251
      IF (NAC.LE.0) GO TO 420                                           CONMI252
      NNAC=NAC                                                          CONMI253
      DO 330 I=1,NNAC                                                   CONMI254
300   NIC=IC(I)                                                         CONMI255
      CT1=CT                                                            CONMI256
      IF (ISC(NIC).GT.0) CT1=CTL                                        CONMI257
      IF (G(NIC).GT.CT1) GO TO 330                                      CONMI258
      NAC=NAC-1                                                         CONMI259
      IF (I.GT.NAC) GO TO 420                                           CONMI260
      DO 320 K=I,NAC                                                    CONMI261
      II=K+1                                                            CONMI262
      DO 310 J=1,NDV2                                                   CONMI263
310   A(J,K)=A(J,II)                                                    CONMI264
320   IC(K)=IC(II)                                                      CONMI265
      GO TO 300                                                         CONMI266
330   CONTINUE                                                          CONMI267
      GO TO 420                                                         CONMI268
340   CONTINUE                                                          CONMI269
      IF (MSCAL.LT.NSCAL.OR.NSCAL.EQ.0) GO TO 360                       CONMI270
      IF (NSCAL.LT.0.AND.KCOUNT.LT.ICNDIR) GO TO 360                    CONMI271
      MSCAL=0                                                           CONMI272
      KCOUNT=0                                                          CONMI273
C     ------------------------------------------------------------------CONMI274
C                              SCALE VARIABLES                          CONMI275
C     ------------------------------------------------------------------CONMI276
      DO 350 I=1,NDV                                                    CONMI277
      SI=SCAL(I)                                                        CONMI278
      XI=SI*X(I)                                                        CONMI279
      SIB=SI                                                            CONMI280
      IF (NSCAL.GT.0) SI=ABS(XI)                                        CONMI281
      IF (SI.LT.1.0E-10) GO TO 350                                      CONMI282
      SCAL(I)=SI                                                        CONMI283
      SI=1./SI                                                          CONMI284
      X(I)=XI*SI                                                        CONMI285
      IF (NSIDE.EQ.0) GO TO 350                                         CONMI286
      VLB(I)=SIB*SI*VLB(I)                                              CONMI287
      VUB(I)=SIB*SI*VUB(I)                                              CONMI288
350   CONTINUE                                                          CONMI289
      IF (IPRINT.LT.4.OR.(NSCAL.LT.0.AND.ITER.GT.1)) GO TO 360          CONMI290
      WRITE (6,1330)                                                    CONMI291
      WRITE (6,1460) (SCAL(I),I=1,NDV)                                  CONMI292
360   CONTINUE                                                          CONMI293
      MSCAL=MSCAL+1                                                     CONMI294
      NAC=0                                                             CONMI295
C     ------------------------------------------------------------------CONMI296
C          OBTAIN GRADIENTS OF OBJECTIVE AND ACTIVE CONSTRAINTS         CONMI297
C     ------------------------------------------------------------------CONMI298
      INFO=2                                                            CONMI299
      NCAL(2)=NCAL(2)+1                                                 CONMI300
      IF (NFDG.NE.1) GO TO 370                                          CONMI301
      IGOTO=2                                                           CONMI302
      GO TO 950                                                         CONMI303
370   CONTINUE                                                          CONMI304
      JGOTO=0                                                           CONMI305
380   CONTINUE                                                          CONMI306
      CALL CNMN01 (JGOTO,X,DF,G,ISC,IC,A,G1,VLB,VUB,SCAL,C,NCAL,DX,DX1,FCONMI307
     1I,XI,III,N1,N2,N3,N4)                                             CONMI308
      IGOTO=3                                                           CONMI309
      IF (JGOTO.GT.0) GO TO 950                                         CONMI310
390   CONTINUE                                                          CONMI311
      INFO=1                                                            CONMI312
      IF (NAC.GE.N3) GO TO 810                                          CONMI313
      IF (NSCAL.EQ.0.OR.NFDG.EQ.0) GO TO 420                            CONMI314
C     ------------------------------------------------------------------CONMI315
C                              SCALE GRADIENTS                          CONMI316
C     ------------------------------------------------------------------CONMI317
C     SCALE GRADIENT OF OBJECTIVE FUNCTION.                             CONMI318
      DO 400 I=1,NDV                                                    CONMI319
400   DF(I)=DF(I)*SCAL(I)                                               CONMI320
      IF (NFDG.EQ.2.OR.NAC.EQ.0) GO TO 420                              CONMI321
C     SCALE GRADIENTS OF ACTIVE CONSTRAINTS.                            CONMI322
      DO 410 J=1,NDV                                                    CONMI323
      SCJ=SCAL(J)                                                       CONMI324
      DO 410 I=1,NAC                                                    CONMI325
410   A(J,I)=A(J,I)*SCJ                                                 CONMI326
420   CONTINUE                                                          CONMI327
      IF (IPRINT.LT.3.OR.NCON.EQ.0) GO TO 470                           CONMI328
C     ------------------------------------------------------------------CONMI329
C                                   PRINT                               CONMI330
C     ------------------------------------------------------------------CONMI331
C     PRINT ACTIVE AND VIOLATED CONSTRAINT NUMBERS.                     CONMI332
      M1=0                                                              CONMI333
      M2=N3                                                             CONMI334
      IF (NAC.EQ.0) GO TO 450                                           CONMI335
      DO 440 I=1,NAC                                                    CONMI336
      J=IC(I)                                                           CONMI337
      IF (J.GT.NCON) GO TO 440                                          CONMI338
      GI=G(J)                                                           CONMI339
      C1=CTAM                                                           CONMI340
      IF (ISC(J).GT.0) C1=CTBM                                          CONMI341
      GI=GI-C1                                                          CONMI342
      IF (GI.GT.0.) GO TO 430                                           CONMI343
C     ACTIVE CONSTRAINT.                                                CONMI344
      M1=M1+1                                                           CONMI345
      MS1(M1)=J                                                         CONMI346
      GO TO 440                                                         CONMI347
430   M2=M2+1                                                           CONMI348
C     VIOLATED CONSTRAINT.                                              CONMI349
      MS1(M2)=J                                                         CONMI350
440   CONTINUE                                                          CONMI351
450   M3=M2-N3                                                          CONMI352
      WRITE (6,1060) M1                                                 CONMI353
      IF (M1.EQ.0) GO TO 460                                            CONMI354
      WRITE (6,1070)                                                    CONMI355
      WRITE (6,1480) (MS1(I),I=1,M1)                                    CONMI356
460   WRITE (6,1080) M3                                                 CONMI357
      IF (M3.EQ.0) GO TO 470                                            CONMI358
      WRITE (6,1070)                                                    CONMI359
      M3=N3+1                                                           CONMI360
      WRITE (6,1480) (MS1(I),I=M3,M2)                                   CONMI361
```

```
470   CONTINUE
C     ------------------------------------------------------------------
C           CALCULATE GRADIENTS OF ACTIVE SIDE CONSTRAINTS
C     ------------------------------------------------------------------
      IF (NSIDE.EQ.0) GO TO 530
      MCN1=NCON
      M1=0
      DO 510 I=1,NDV
C     LOWER BOUND.
      XI=X(I)
      XID=VLB(I)
      X12=ABS(XID)
      IF (X12.LT.1.) X12=1.
      GI=(XID-XI)/X12
      IF (GI.LT.-1.0E-6) GO TO 490
      M1=M1+1
      MS1(M1)=-I
      NAC=NAC+1
      IF (NAC.GE.N3) GO TO 810
      MCN1=MCN1+1
      DO 480 J=1,NDV
480   A(J,NAC)=0.
      A(I,NAC)=-1.
      IC(NAC)=MCN1
      G(MCN1)=GI
      ISC(MCN1)=1
C     UPPER BOUND.
490   XID=VUB(I)
      X12=ABS(XID)
      IF (X12.LT.1.) X12=1.
      GI=(XI-XID)/X12
      IF (GI.LT.-1.0E-6) GO TO 510
      M1=M1+1
      MS1(M1)=I
      NAC=NAC+1
      IF (NAC.GE.N3) GO TO 810
      MCN1=MCN1+1
      DO 500 J=1,NDV
500   A(J,NAC)=0.
      A(I,NAC)=1.
      IC(NAC)=MCN1
      G(MCN1)=GI
      ISC(MCN1)=1
510   CONTINUE
C     ------------------------------------------------------------------
C                                 PRINT
C     ------------------------------------------------------------------
C     PRINT ACTIVE SIDE CONSTRAINT NUMBERS.
      IF (IPRINT.LT.3) GO TO 530
      WRITE (6,1090) M1
      IF (M1.EQ.0) GO TO 530
      WRITE (6,1100)
      WRITE(6,1480) (MS1(J),J=1,M1)
530   CONTINUE
C     PRINT GRADIENTS OF ACTIVE AND VIOLATED CONSTRAINTS.
      IF (IPRINT.LT.4) GO TO 570
      WRITE (6,1340)
      DO 540 I=1,NDV,6
      M1=MINO(NDV,I+5)
540   WRITE (6,1010) I,(DF(J),J=I,M1)
      IF (NAC.EQ.0) GO TO 570
      WRITE (6,1350)
      DO 560 I=1,NAC
      M1=IC(I)
      M2=M1-NCON
      M3=0
      IF (M2.GT.0) M3=IABS(MS1(M2))
      IF (M2.LE.0) WRITE (6,990) M1
      IF (M2.GT.0) WRITE (6,1000) M3
      DO 550 K=1,NDV,6
      M1=MINO(NDV,K+5)
550   WRITE (6,1010) K,(A(J,I),J=K,M1)
560   WRITE (6,1360)
570   CONTINUE
C     ------------------------------------------------------------------
C     ****************** DETERMINE SEARCH DIRECTION ******************
C     ------------------------------------------------------------------
      ALP=1.0E+20
      IF (NAC.GT.0) GO TO 580
C     ------------------------------------------------------------------
C                          UNCONSTRAINED FUNCTION
C     ------------------------------------------------------------------
C     FIND DIRECTION OF STEEPEST DESCENT OR CONJUGATE DIRECTION.
      NVC=0
      NFEAS=0
      KCOUNT=KCOUNT+1
C     IF KCOUNT.GT.ICNDIR  RESTART CONJUGATE DIRECTION ALGORITHM.
      IF (KCOUNT.GT.ICNDIR.OR.IOBJ.EQ.2) KCOUNT=1
      IF (KCOUNT.EQ.1) JDIR=0
C     IF JDIR = 0 FIND DIRECTION OF STEEPEST DESCENT.
      CALL CNMN02 (JDIR,SLOPE,DFTDF1,DF,S,N1)
      GO TO 630
580   CONTINUE
C     ------------------------------------------------------------------
C                           CONSTRAINED FUNCTION
C     ------------------------------------------------------------------
C     FIND USABLE-FEASIBLE DIRECTION.
      KCOUNT=0
      JDIR=0
      PHI=10.*PHI
      IF (PHI.GT.1000.) PHI=1000.
C     CALCULATE DIRECTION, S.
      CALL CNMN05 (G,DF,A,S,B,C,SLOPE,PHI,ISC,IC,MS1,NVC,N1,N2,N3,N4,N5)
      IF (IPRINT.LT.3) GO TO 600
      WRITE (6,1370)
      DO 590 I=1,NAC,6
      M1=MINO(NAC,I+5)
590   WRITE (6,1010) I,(A(NDV1,J),J=I,M1)
      WRITE (6,1210) S(NDV1)
600   CONTINUE
C     ------------------------------------------------------------------
C     ****************** ONE-DIMENSIONAL SEARCH ***********************
C     ------------------------------------------------------------------
      IF (S(NDV1).LT.1.0E-6.AND.NVC.EQ.0) GO TO 710
C     ------------------------------------------------------------------
C                  FIND ALPHA TO OBTAIN A FEASIBLE DESIGN
C     ------------------------------------------------------------------
      IF (NVC.EQ.0) GO TO 630
      ALP=-1.
      DO 620 I=1,NAC
```

```
      NCI=IC(I)
      C1=G(NCI)
      CTC=CTAM
      IF (ISC(NCI).GT.0) CTC=CTBM
      IF (C1.LE.CTC) GO TO 620
      ALP1=0.
      DO 610 J=1,NDV
  610 ALP1=ALP1+S(J)*A(J,I)
      ALP1=ALP1*A(NDV2,I)
      IF (ABS(ALP1).LT.1.0E-20) GO TO 620
      ALP1=-C1/ALP1
      IF (ALP1.GT.ALP) ALP=ALP1
  620 CONTINUE
  630 CONTINUE
C     ------------------------------------------------------------------
C                     LIMIT CHANCE TO ABOBJ1*OBJ
C     ------------------------------------------------------------------
      ALP1=1.0E+20
      SI=ABS(OBJ)
      IF (SI.LT..01) SI=.01
      IF (ABS(SLOPE).GT.1.0E-20) ALP1=ABOBJ1*SI/SLOPE
      ALP1=ABS(ALP1)
      IF (NVC.GT.0) ALP1=10.*ALP1
      IF (ALP1.LT.ALP) ALP=ALP1
C     ------------------------------------------------------------------
C               .   LIMIT CHANGE IN VARIABLE TO ALPHAX
C     ------------------------------------------------------------------
      ALP11=1.0E+20
      DO 640 I=1,NDV
      SI=ABS(S(I))
      XI=ABS(X(I))
      IF (SI.LT.1.0E-10.OR.XI.LT.0.1) GO TO 640
      ALP1=ALPHAX*XI/SI
      IF (ALP1.LT.ALP11) ALP11=ALP1
  640 CONTINUE
      IF (NVC.GT.0) ALP11=10.*ALP11
      IF (ALP11.LT.ALP) ALP=ALP11
      IF (ALP.GT.1.0E+20) ALP=1.0E+20
      IF (ALP.LE.1.0E-20) ALP=1.0E-20
      IF (IPRINT.LT.3) GO TO 660
      WRITE (6,1380)
      DO 650 I=1,NDV,6
      M1=MIN0(NDV,I+5)
  650 WRITE (6,1010) I,(S(J),J=I,M1)
      WRITE (6,1110) SLOPE,ALP
  660 CONTINUE
      IF (NCON.GT.0.OR.NSIDE.GT.0) GO TO 680
C     ------------------------------------------------------------------
C           DO ONE-DIMENSIONAL SEARCH FOR UNCONSTRAINED FUNCTION
C     ------------------------------------------------------------------
      JGOTO=0
  670 CONTINUE
      CALL CNMN03 (X,S,SLOPE,ALP,FFF,A1,A2,A3,A4,F1,F2,F3,F4,APP,N1,NCAL
     1,KOUNT,JGOTO)
      IGOTO=4
      IF (JGOTO.GT.0) GO TO 950
      JDIR=1
C     PROCEED TO CONVERGENCE CHECK.
      GO TO 700
C     ------------------------------------------------------------------
C        SOLVE ONE-DIMENSIONAL SEARCH PROBLEM FOR CONSTRAINED FUNCTION
C     ------------------------------------------------------------------
  680 CONTINUE
      JGOTO=0
  690 CONTINUE
      CALL CNMN06 (X,VLB,VUB,G,SCAL,DF,S,G1,G2,CTAM,CTBM,SLOPE,ALP,A2,A3
     1,A4,F1,F2,F3,CV1,CV2,CV3,CV4,ALPCA,ALPFES,ALPLN,ALPMIN,ALPNC,ALPSA
     2V,ALPSID,ALPTOT,ISC,N1,NC,NCAL,NVC,ICOUNT,IGOOD1,IGOOD2,IGOOD3,IGO
     3OD4,IBEST,III,NLNC,JGOTO)
      IGOTO=5
      IF (JGOTO.GT.0) GO TO 950
      IF (NAC.EQ.0) JDIR=1
C     ------------------------------------------------------------------
C     ******************     UPDATE ALPHAX     ***********************
C     ------------------------------------------------------------------
  700 CONTINUE
  710 CONTINUE
      IF (ALP.GT.1.0E+19) ALP=0.
C     UPDATE ALPHAX TO BE AVERAGE OF MAXIMUM CHANGE IN X(I)
C     AND ALHPAX.
      ALP11=0.
      DO 720 I=1,NDV
      SI=ABS(S(I))
      XI=ABS(X(I))
      IF (XI.LT.1.0E-10) GO TO 720
      ALP1=ALP*SI/XI
      IF (ALP1.GT.ALP11) ALP11=ALP1
  720 CONTINUE
      ALP11=.5*(ALP11+ALPHAX)
      ALP12=5.*ALPHAX
      IF (ALP11.GT.ALP12) ALP11=ALP12
      ALPHAX=ALP11
      NCOBJ=NCOBJ+1
C     ABSOLUTE CHANGE IN OBJECTIVE.
      OBJD=OBJ1-OBJ
      OBJB=ABS(OBJD)
      IF (OBJB.LT.1.0E-10) OBJB=0.
      IF (NAC.EQ.0.OR.OBJB.GT.0.) NCOBJ=0
      IF (NCOBJ.GT.1) NCOBJ=0
C     ------------------------------------------------------------------
C                                  PRINT
C     ------------------------------------------------------------------
C     PRINT MOVE PARAMETER, NEW X-VECTOR AND CONSTRAINTS.
      IF (IPRINT.LT.3) GO TO 730
      WRITE (6,1390) ALP
  730 IF (IPRINT.LT.2) GO TO 800
      IF (OBJB.GT.0.) GO TO 740
      IF (IPRINT.EQ.2) WRITE (6,1400) ITER,OBJ
      IF (IPRINT.GT.2) WRITE (6,1410) OBJ
      GO TO 760
  740 IF (IPRINT.EQ.2) GO TO 750
      WRITE (6,1420) OBJ
      GO TO 760
  750 WRITE (6,1430) ITER,OBJ
  760 WRITE (6,1450)
      DO 770 I=1,NDV
      FF1=1.
      IF (NSCAL.NE.0) FF1=SCAL(I)
  770 G1(I)=FF1*X(I)
      DO 780 I=1,NDV,6
```

```
      M1=MIN0(NDV,I+5)
780   WRITE (6,1010) I,(G1(J),J=I,M1)
      IF (NCON.EQ.0) GO TO 800
      WRITE (6,1470)
      DO 790 I=1,NCON,6
      M1=MIN0(NCON,I+5)
790   WRITE (6,1010) I,(G(J),J=I,M1)
800   CONTINUE
C     ------------------------------------------------------------------
C                          CHECK FEASABILITY
C     ------------------------------------------------------------------
      IF(NCON.LE.0) GO TO 808
      DO 804 I=1,NCON
      C1=CTAM
      IF(ISC(I).GT.0) C1=CTBM
      IF(G(I).LE.C1) GO TO 804
      NFEAS=NFEAS+1
      GO TO 806
804   CONTINUE
      IF(NFEAS.GT.0) ABOBJ1=.05
      NFEAS=0
      PHI=5.
806   IF(NFEAS.GE.10) GO TO 810
808   CONTINUE
C     ------------------------------------------------------------------
C                          CHECK CONVERGENCE
C     ------------------------------------------------------------------
C     STOP IF ITER EQUALS ITMAX.
      IF (ITER.GE.ITMAX) GO TO 810
C     ------------------------------------------------------------------
C                     ABSOLUTE CHANGE IN OBJECTIVE
C     ------------------------------------------------------------------
      OBJB=ABS(OBJD)
      KOBJ=KOBJ+1
      IF (OBJB.GE.DABFUN.OR.NFEAS.GT.0) KOBJ=0
C     ------------------------------------------------------------------
C                     RELATIVE CHANGE IN OBJECTIVE
C     ------------------------------------------------------------------
      IF (ABS(OBJ1).GT.1.0E-10) OBJD=OBJD/ABS(OBJ1)
      ABOBJ1=.5*(ABS(ABOBJ)+ABS(OBJD))
      ABOBJ=ABS(OBJD)
      IOBJ=IOBJ+1
      IF(NFEAS.GT.0.OR.OBJD.GE.DELFUN) IOBJ=0
      IF (IOBJ.GE.ITRM.OR.KOBJ.GE.ITRM) GO TO 810
      OBJ1=OBJ
C     ------------------------------------------------------------------
C           REDUCE CT IF OBJECTIVE FUNCTION IS CHANGING SLOWLY
C     ------------------------------------------------------------------
      IF (IOBJ.LT.1.OR.NAC.EQ.0) GO TO 280
      CT=DCT*CT
      CTL=CTL*DCTL
      IF (ABS(CT).LT.CTMIN) CT=-CTMIN
      IF (ABS(CTL).LT.CTLMIN) CTL=-CTLMIN
      GO TO 280
810   CONTINUE
      IF (NAC.GE.N3) WRITE (6,1490)
C     ------------------------------------------------------------------
C     ****************  FINAL FUNCTION INFORMATION  *******************
C     ------------------------------------------------------------------
      IF (NSCAL.EQ.0) GO TO 830
C     UN-SCALE THE DESIGN VARIABLES.
      DO 820 I=1,NDV
      XI=SCAL(I)
      IF (NSIDE.EQ.0) GO TO 820
      VLB(I)=XI*VLB(I)
      VUB(I)=XI*VUB(I)
820   X(I)=XI*X(I)
C     ------------------------------------------------------------------
C                         PRINT FINAL RESULTS
C     ------------------------------------------------------------------
830   IF (IPRINT.EQ.0.OR.NAC.GE.N3) GO TO 940
      WRITE (6,1500)
      WRITE (6,1420) OBJ
      WRITE (6,1450)
      DO 840 I=1,NDV,6
      M1=MIN0(NDV,I+5)
840   WRITE (6,1010) I,(X(J),J=I,M1)
      IF (NCON.EQ.0) GO TO 900
      WRITE (6,1470)
      DO 850 I=1,NCON,6
      M1=MIN0(NCON,I+5)
850   WRITE (6,1010) I,(G(J),J=I,M1)
C     DETERMINE WHICH CONSTRAINTS ARE ACTIVE AND PRINT.
      NAC=0
      NVC=0
      DO 870 I=1,NCON
      CTA=CTAM
      IF (ISC(I).GT.0) CTA=CTBM
      GI=G(I)
      IF (GI.GT.CTA) GO TO 860
      IF (GI.LT.CT.AND.ISC(I).EQ.0) GO TO 870
      IF (GI.LT.CTL.AND.ISC(I).GT.0) GO TO 870
      NAC=NAC+1
      IC(NAC)=I
      GO TO 870
860   NVC=NVC+1
      MS1(NVC)=I
870   CONTINUE
      WRITE (6,1060) NAC
      IF (NAC.EQ.0) GO TO 880
      WRITE (6,1070)
      WRITE (6,1480) (IC(J),J=1,NAC)
880   WRITE (6,1080) NVC
      IF (NVC.EQ.0) GO TO 890
      WRITE (6,1070)
      WRITE (6,1480) (MS1(J),J=1,NVC)
890   CONTINUE
900   CONTINUE
      IF (NSIDE.EQ.0) GO TO 930
C     DETERMINE WHICH SIDE CONSTRAINTS ARE ACTIVE AND PRINT.
      NAC=0
      DO 920 I=1,NDV
      XI=X(I)
      XID=VLB(I)
      X12=ABS(XID)
      IF (X12.LT.1.) X12=1.
      GI=(XID-XI)/X12
      IF (GI.LT.-1.0E-6) GO TO 910
      NAC=NAC+1
      MS1(NAC)=-I
```

```
910   XID=VUB(I)
      X12=ABS(XID)
      IF (X12.LT.1.) X12=1.
      GI=(XI-XID)/X12
      IF (GI.LT.-1.0E-6) GO TO 920
      NAC=NAC+1
      MS1(NAC)=I
920   CONTINUE
      WRITE (6,1090) NAC
      IF (NAC.EQ.0) GO TO 930
      WRITE (6,1100)
      WRITE (6,1480) (MS1(J),J=1,NAC)
930   CONTINUE
      WRITE (6,1150)
      IF (ITER.GE.ITMAX) WRITE (6,1160)
      IF (NFEAS.GE.10) WRITE (6,1170)
      IF (IOBJ.GE.ITRM) WRITE (6,1180) ITRM
      IF (KOBJ.GE.ITRM) WRITE (6,1190) ITRM
      WRITE (6,1200) ITER
      WRITE (6,1510) NCAL(1)
      IF (NCON.GT.0) WRITE (6,1520) NCAL(1)
      IF (NFDG.NE.0) WRITE (6,1530) NCAL(2)
      IF (NCON.GT.0.AND.NFDG.EQ.1) WRITE (6,1540) NCAL(2)
C     ------------------------------------------------------------------
C                  RE-SET BASIC PARAMETERS TO INPUT VALUES
C     ------------------------------------------------------------------
940   ITRM=IDM1
      ITMAX=IDM2
      ICNDIR=IDM3
      DELFUN=DM1
      DABFUN=DM2
      CT=DM3
      CTMIN=DM4
      CTL=DM5
      CTLMIN=DM6
      THETA=DM7
      PHI=DM8
      FDCH=DM9
      FDCHM=DM10
      ABOBJ1=DM11
      ALPHAX=DM12
      IGOTO=0
950   CONTINUE
      IF (NSCAL.EQ.0.OR.IGOTO.EQ.0) RETURN
C     UN-SCALE VARIABLES.
      DO 960 I=1,NDV
      C(I)=X(I)
960   X(I)=X(I)*SCAL(I)
      RETURN
C     ------------------------------------------------------------------
C                                 FORMATS
C     ------------------------------------------------------------------
C
C
970   FORMAT (///5X,72HA COMPLETELY UNCONSTRAINED FUNCTION WITH A LINEAR
     1 OBJECTIVE IS SPECIFIED//10X,8HLINOBJ =,I5/10X,8HNCON    =,I5/10X,8
     2HNSIDE  =,I5//5X,35HCONTROL RETURNED TO CALLING PROGRAM)
980   FORMAT (///5X,56HCONMIN HAS ACHIEVED A SOLUTION OF OBJ LESS THAN -
     11.0E+40/5X,32HSOLUTION APPEARS TOABE UNBOUNDED/5X,26HOPTIMIZATION
     2IS TERMINATED)
990   FORMAT (5X,17HCONSTRAINT NUMBER,I5)
1000  FORMAT (5X,27HSIDE CONSTRAINT ON VARIABLE,I5)
1010  FORMAT (3X,I5,1H),2X,6E13.5)
1020  FORMAT (/5X,35HLINEAR CONSTRAINT IDENTIFIERS (ISC)/5X,36HNON-ZERO
     1INDICATES LINEAR CONSTRAINT)
1030  FORMAT (3X,I5,1H),2X,15I5)
1040  FORMAT (/5X,26HALL CONSTRAINTS ARE LINEAR)
1050  FORMAT (/5X,30HALL CONSTRAINTS ARE NON-LINEAR)
1060  FORMAT (/5X,9HTHERE ARE,I5,19H ACTIVE CONSTRAINTS)
1070  FORMAT (5X,22HCONSTRAINT NUMBERS ARE)
1080  FORMAT (/5X,9HTHERE ARE,I5,21H VIOLATED CONSTRAINTS)
1090  FORMAT (/5X,9HTHERE ARE,I5,24H ACTIVE SIDE CONSTRAINTS)
1100  FORMAT (5X,43HDECISION VARIABLES AT LOWER OR UPPER BOUNDS,30H (MIN
     1US INDICATES LOWER BOUND))
1110  FORMAT (/5X,22HONE-DIMENSIONAL SEARCH/5X,15HINITIAL SLOPE =,E12.4,
     12X,16HPROPOSED ALPHA =,E12.4)
1120  FORMAT (///5X,35H* * CONMIN DETECTS VLB(I).GT.VUB(I)/5X,57HFIX IS
     1SET X(I)=VLB(I)=VUB(I) = .5*(VLB(I)+VUB(I) FOR I =,I5)
1130  FORMAT (///5X,41H* * CONMIN DETECTS INITIAL X(I).LT.VLB(I)/5X,6HX(
     1I) =,E12.4,2X,8HVLB(I) =,E12.4/5X,35HX(I) IS SET EQUAL TO VLB(I) F
     2OR I =,I5)
1140  FORMAT (///5X,41H* * CONMIN DETECTS INITIAL X(I).GT.VUB(I)/5X,6HX(
     1I) =,E12.4,2X,8HVUB(I) =,E12.4/5X,35HX(I) IS SET EQUAL TO VUB(I) F
     2OR I =,I5)
1150  FORMAT (/5X,21HTERMINATION CRITERION)
1160  FORMAT (10X,17HITER EQUALS ITMAX)
1170  FORMAT (10X,62HTEN CONSECUTIVE ITERATIONS FAILED TO PRODUCE A FEAS
     1IBLE DESIGN)
1180  FORMAT (10X,43HABS(1-OBJ(I-1)/OBJ(I)) LESS THAN DELFUN FOR,I3,11H
     1ITERATIONS)
1190  FORMAT (10X,43HABS(OBJ(I)-OBJ(I-1))   LESS THAN DABFUN FOR,I3,11H
     1ITERATIONS)
1200  FORMAT (/5X,22HNUMBER OF ITERATIONS =,I5)
1210  FORMAT (/5X,28HCONSTRAINT PARAMETER, BETA =,E14.5)
1220  FORMAT (1H1,////12X,27(2H* )/12X,1H*,51X,1H*/12X,1H*,20X,11HC O N
     1M I N,20X,1H*/12X,1H*,51X,1H*/12X,1H*,15X,21H FORTRAN PROGRAM FOR
     2,15X,1H*/12X,1H*,51X,1H*/12X,1H*,9X,33HCONSTRAINED FUNCTION MINIMI
     3ZATION,9X,1H*/12X,1H*,51X,1H*/12X,27(2H* ))
1230  FORMAT (////5X,33HCONSTRAINED FUNCTION MINIMIZATION//5X,18HCONTROL
     1 PARAMETERS)
1240  FORMAT (/5X,60HIPRINT  NDV    ITMAX    NCON    NSIDE  ICNDIR   NSC
     1AL   NFDG/8I8//5X,12HLINOBJ  ITRM,5X,2HN1,6X,2HN2,6X,2HN3,6X,2HN4,
     26X,2HN5/8I8)
1250  FORMAT (/9X,4HFDCH,12X,5HFDCHM,11X,6HALPHAX,10X,6HABOBJ1/1X,4(2X,E
     114.5))
1260  FORMAT (/9X,2HCT,14X,5HCTMIN,11X,3HCTL,13X,6HCTLMIN/1X,4(2X,E14.5)
     1//9X,5HTHETA,11X,3HPHI,13X,6HDELFUN,10X,6HDABFUN/1X,4(2X,E14.5))
1270  FORMAT (/5X,40HLOWER BOUNDS ON DECISION VARIABLES (VLB))
1280  FORMAT (/5X,40HUPPER BOUNDS ON DECISION VARIABLES (VUB))
1290  FORMAT (////5X,35HUNCONSTRAINED FUNCTION MINIMIZATION//5X,18HCONTR
     1OL PARAMETERS)
1300  FORMAT (/5X,21HSCALING VECTOR (SCAL))
1310  FORMAT (////5X,22HBEGIN ITERATION NUMBER,I5)
1320  FORMAT (/5X,4HCT =,E14.5,5X,5HCTL =,E14.5,5X,5HPHI =,E14.5)
1330  FORMAT (/5X,25HNEW SCALING VECTOR (SCAL))
1340  FORMAT (/5X,15HGRADIENT OF OBJ)
1350  FORMAT (/5X,44HGRADIENTS OF ACTIVE AND VIOLATED CONSTRAINTS)
1360  FORMAT (1H )
1370  FORMAT (/5X,37HPUSH-OFF FACTORS, (THETA(I), I=1,NAC))
1380  FORMAT (/5X,27HSEARCH DIRECTION (S-VECTOR))
```

```
1390  FORMAT (/5X,18HCALCULATED ALPHA =,E14.5)
1400  FORMAT (////5X,6HITER =,I5,5X,5HOBJ =,E14.5,5X,16HNO CHANGE IN OBJ
     1)
1410  FORMAT (/5X,5HOBJ =,E15.6,5X,16HNO CHANGE ON OBJ)
1420  FORMAT (/5X,5HOBJ =,E15.6)
1430  FORMAT (////5X,6HITER =,I5,5X,5HOBJ =,E14.5)
1440  FORMAT (//5X,28HINITIAL FUNCTION INFORMATION//5X,5HOBJ =,E15.6)
1450  FORMAT (/5X,29HDECISION VARIABLES (X-VECTOR))
1460  FORMAT (3X,7E13.4)
1470  FORMAT (/5X,28HCONSTRAINT VALUES (G-VECTOR))
1480  FORMAT (5X,15I5)
1490  FORMAT (/5X,59HTHE NUMBER OF ACTIVE AND VIOLATED CONSTRAINTS EXCEE
     1DS N3-1./5X,66HDIMENSIONED SIZE OF MATRICES A AND B AND VECTOR IC
     2IS INSUFFICIENT/5X,61HOPTIMIZATION TERMINATED AND CONTROL RETURNED
     3 TO MAIN PROGRAM.)
1500  FORMAT (1H1,////4X,30HFINAL OPTIMIZATION INFORMATION)
1510  FORMAT (/5X,32HOBJECTIVE FUNCTION WAS EVALUATED,8X,I5,2X,5HTIMES)
1520  FORMAT (/5X,35HCONSTRAINT FUNCTIONS WERE EVALUATED,I10,2X,5HTIMES)
1530  FORMAT (/5X,36HGRADIENT OF OBJECTIVE WAS CALCULATED,I9,2X,5HTIMES)
1540  FORMAT (/5X,40HGRADIENTS OF CONSTRAINTS WERE CALCULATED,I5,2X,5HTI
     1MES)
      END
      SUBROUTINE CNMN01 (JGOTO,X,DF,G,ISC,IC,A,G1,VLB,VUB,SCAL,C,NCAL,DX
     1,DX1,FI,XI,III,N1,N2,N3,N4)
      COMMON /CNMN1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAX
     1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,IT
     2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER
      DIMENSION X(N1), DF(N1), G(N2), ISC(N2), IC(N3), A(N1,N3), G1(N2),
     1 VLB(N1), VUB(N1), SCAL(N1), NCAL(2), C(N4)
C     ROUTINE TO CALCULATE GRADIENT INFORMATION BY FINITE DIFFERENCE.
C     BY G. N. VANDERPLAATS                          JUNE, 1972.
C     NASA-AMES RESEARCH CENTER,  MOFFETT FIELD, CALIF.
      IF (JGOTO.EQ.1) GO TO 10
      IF (JGOTO.EQ.2) GO TO 70
      INFOG=0
      INF=INFO
      NAC=0
      IF (LINOBJ.NE.0.AND.ITER.GT.1) GO TO 10
C     ------------------------------------------------------------------
C                      GRADIENT OF LINEAR OBJECTIVE
C     ------------------------------------------------------------------
      IF (NFDG.EQ.2) JGOTO=1
      IF (NFDG.EQ.2) RETURN
10    CONTINUE
      JGOTO=0
      IF (NFDG.EQ.2.AND.NCON.EQ.0) RETURN
      IF (NCON.EQ.0) GO TO 40
C     ------------------------------------------------------------------
C       * * * DETERMINE WHICH CONSTRAINTS ARE ACTIVE OR VIOLATED * * *
C     ------------------------------------------------------------------
      DO 20 I=1,NCON
      IF (G(I).LT.CT) GO TO 20
      IF (ISC(I).GT.0.AND.G(I).LT.CTL) GO TO 20
      NAC=NAC+1
      IF (NAC.GE.N3) RETURN
      IC(NAC)=I
20    CONTINUE
      IF (NFDG.EQ.2.AND.NAC.EQ.0) RETURN
      IF ((LINOBJ.GT.0.AND.ITER.GT.1).AND.NAC.EQ.0) RETURN
C     ------------------------------------------------------------------
C                  STOPE VALUES OF CONSTRAINTS IN G1
C     ------------------------------------------------------------------
      DO 30 I=1,NCON
30    G1(I)=G(I)
40    CONTINUE
      JGOTO=0
      IF (NAC.EQ.0.AND.NFDG.EQ.2) RETURN
C     ------------------------------------------------------------------
C                         CALCULATE GRADIENTS
C     ------------------------------------------------------------------
      INFOG=1
      INFO=1
      FI=OBJ
      III=0
50    III=III+1
      XI=X(III)
      DX=FDCH*XI
      DX=ABS(DX)
      FDCH1=FDCHM
      IF (NSCAL.NE.0) FDCH1=FDCHM/SCAL(III)
      IF (DX.LT.FDCH1) DX=FDCH1
      X1=XI+DX
      IF (NSIDE.EQ.0) GO TO 60
      IF (X1.GT.VUB(III)) DX=-DX
60    DX1=1./DX
      X(III)=XI+DX
      NCAL(1)=NCAL(1)+1
C     ------------------------------------------------------------------
C                         FUNCTION EVALUATION
C     ------------------------------------------------------------------
      JGOTO=2
      RETURN
70    CONTINUE
      X(III)=XI
      IF (NFDG.EQ.0) DF(III)=DX1*(OBJ-FI)
      IF (NAC.EQ.0) GO TO 90
C     ------------------------------------------------------------------
C           DETERMINE GRADIENT COMPONENTS OF ACTIVE CONSTRAINTS
C     ------------------------------------------------------------------
      DO 80 J=1,NAC
      I1=IC(J)
80    A(III,J)=DX1*(G(I1)-G1(I1))
90    CONTINUE
      IF (III.LT.NDV) GO TO 50
      INFOG=0
      INFO=INF
      JGOTO=0
      OBJ=FI
      IF (NCON.EQ.0) RETURN
C     ------------------------------------------------------------------
C            STORE CURRENT CONSTRAINT VALUES BACK IN G-VECTOR
C     ------------------------------------------------------------------
      DO 100 I=1,NCON
100   G(I)=G1(I)
      RETURN
      END
      SUBROUTINE CNMN02 (NCALC,SLOPE,DFTDF1,DF,S,N1)
      COMMON /CNMN1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAX
     1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,IT
     2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER
```

```
      DIMENSION DF(N1), S(N1)
C     ROUTINE TO DETERMINE CONJUGATE DIRECTION VECTOR OR DIRECTION
C     OF STEEPEST DESCENT FOR UNCONSTRAINED FUNCTION MINIMIZATION.
C     BY G. N. VANDERPLAATS                    APRIL, 1972.
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.
C     NCALC = CALCULATION CONTROL.
C         NCALC = 0,     S = STEEPEST DESCENT.
C         NCALC = 1,     S = CONJUGATE DIRECTION.
C     CONJUGATE DIRECTION IS FOUND BY FLETCHER-REEVES ALGORITHM.
C     ------------------------------------------------------------------
C                  CALCULATE NORM OF GRADIENT VECTOR
C     ------------------------------------------------------------------
      DFTDF=0.
      DO 10 I=1,NDV
      DFI=DF(I)
10    DFTDF=DFTDF+DFI*DFI
C     ------------------------------------------------------------------
C     **********                FIND DIRECTION S              **********
C     ------------------------------------------------------------------
      IF (NCALC.NE.1) GO TO 30
      IF (DFTDF1.LT.1.0E-20) GO TO 30
C     ------------------------------------------------------------------
C                FIND FLETCHER-REEVES CONJUGATE DIRECTION
C     ------------------------------------------------------------------
      BETA=DFTDF/DFTDF1
      SLOPE=0.
      DO 20 I=1,NDV
      DFI=DF(I)
      SI=BETA*S(I)-DFI
      SLOPE=SLOPE+SI*DFI
20    S(I)=SI
      GO TO 50
30    CONTINUE
      NCALC=0
C     ------------------------------------------------------------------
C                 CALCULATE DIRECTION OF STEEPEST DESCENT
C     ------------------------------------------------------------------
      DO 40 I=1,NDV
40    S(I)=-DF(I)
      SLOPE=-DFTDF
50    CONTINUE
C     ------------------------------------------------------------------
C                 NORMALIZE S TO MAX ABS VALUE OF UNITY
C     ------------------------------------------------------------------
      S1=0.
      DO 60 I=1,NDV
      S2=ABS(S(I))
      IF (S2.GT.S1) S1=S2
60    CONTINUE
      IF (S1.LT.1.0E-20) S1=1.0E-20
      S1=1./S1
      DFTDF1=DFTDF*S1
      DO 70 I=1,NDV
70    S(I)=S1*S(I)
      SLOPE=S1*SLOPE
      RETURN
      END
      SUBROUTINE CNMN03 (X,S,SLOPE,ALP,FFF,A1,A2,A3,A4,F1,F2,F3,F4,APP,N
     11,NCAL,KOUNT,JGOTO)
      COMMON /CNMN1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAX
     1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,IT
     2RM,ICNDIR,IGOTO,NAC,INFO,INFCG,ITER
      DIMENSION X(N1), S(N1), NCAL(2)
C     ROUTINE TO SOLVE ONE-DIMENSIONAL SEARCH IN UNCONSTRAINED
C     MINIMIZATION USING 2-POINT QUADRATIC INTERPOLATION, 3-POINT
C     CUBIC INTERPOLATION AND 4-POINT CUBIC INTERPOLATION.
C     BY G. N. VANDERPLAATS                         APRIL, 1972.
C     NASA-AMES RESEARCH CENTER,  MOFFETT FIELD, CALIF.
C     ALP = PROPOSED MOVE PARAMETER.
C     SLOPE = INITIAL FUNCTION SLOPE = S-TRANSPOSE TIMES DF.
C     SLOPE MUST BE NEGATIVE.
C     OBJ = INITIAL FUNCTION VALUE.
      ZRO=0.
      IF (JGOTO.EQ.0) GO TO 10
      GO TO (50,80,110,140,180,220,270),JGOTO
C     ------------------------------------------------------------------
C                    INITIAL INFORMATION  (ALPHA=0)
C     ------------------------------------------------------------------
10    IF (SLOPE.LT.0.) GO TO 20
      ALP=0.
      RETURN
20    CONTINUE
      IF (IPRINT.GT.4) WRITE (6,360)
      FFF=OBJ
      AP1=0.
      A1=0.
      F1=OBJ
      A2=ALP
      A3=0.
      F3=0.
      AP=A2
      KOUNT=0
C     ------------------------------------------------------------------
C            MOVE A DISTANCE AP*S AND UPDATE FUNCTION VALUE
C     ------------------------------------------------------------------
30    CONTINUE
      KOUNT=KOUNT+1
      DO 40 I=1,NDV
40    X(I)=X(I)+AP*S(I)
      IF (IPRINT.GT.4) WRITE (6,370) AP
      IF (IPRINT.GT.4) WRITE (6,380) (X(I),I=1,NDV)
      NCAL(1)=NCAL(1)+1
      JGOTO=1
      RETURN
50    CONTINUE
      F2=OBJ
      IF (IPRINT.GT.4) WRITE (6,390) F2
      IF (F2.LT.F1) GO TO 120
C     ------------------------------------------------------------------
C                      CHECK FOR ILL-CONDITIONING
C     ------------------------------------------------------------------
      IF (KOUNT.GT.5) GO TO 60
      FF=2.*ABS(F1)
      IF (F2.LT.FF) GO TO 90
      FF=5.*ABS(F1)
      IF (F2.LT.FF) GO TO 60
      A2=.5*A2
      AP=-A2
      ALP=A2
      GO TO 30
```

```
60    F3=F2
      A3=A2
      A2=.5*A2
C     -----------------------------------------------------------------
C                 UPDATE DESIGN VECTOR AND FUNCTION VALUE
C     -----------------------------------------------------------------
      AP=A2-ALP
      ALP=A2
      DO 70 I=1,NDV
70    X(I)=X(I)+AP*S(I)
      IF (IPRINT.GT.4) WRITE (6,370) A2
      IF (IPRINT.GT.4) WRITE (6,380) (X(I),I=1,NDV)
      NCAL(1)=NCAL(1)+1
      JGOTO=2
      RETURN
80    CONTINUE
      F2=OBJ
      IF (IPRINT.GT.4) WRITE (6,390) F2
C     PROCEED TO CUBIC INTERPOLATION.
      GO TO 160
90    CONTINUE
C     -----------------------------------------------------------------
C     **********        2-POINT QUADRATIC INTERPOLATION       **********
C     -----------------------------------------------------------------
      JJ=1
      II=1
      CALL CNMN04 (II,APP,ZRO,A1,F1,SLOPE,A2,F2,ZRO,ZRO,ZRO,ZRO)
      IF (APP.LT.ZRO.OR.APP.GT.A2) GO TO 120
      F3=F2
      A3=A2
      A2=APP
      JJ=0
C     -----------------------------------------------------------------
C                 UPDATE DESIGN VECTOR AND FUNCTION VALUE
C     -----------------------------------------------------------------
      AP=A2-ALP
      ALP=A2
      DO 100 I=1,NDV
100   X(I)=X(I)+AP*S(I)
      IF (IPRINT.GT.4) WRITE (6,370) A2
      IF (IPRINT.GT.4) WRITE (6,380) (X(I),I=1,NDV)
      NCAL(1)=NCAL(1)+1
      JGOTO=3
      RETURN
110   CONTINUE
      F2=OBJ
      IF (IPRINT.GT.4) WRITE (6,390) F2
      GO TO 150
120   A3=2.*A2
C     -----------------------------------------------------------------
C                 UPDATE DESIGN VECTOR AND FUNCTION VALUE
C     -----------------------------------------------------------------
      AP=A3-ALP
      ALP=A3
      DO 130 I=1,NDV
130   X(I)=X(I)+AP*S(I)
      IF (IPRINT.GT.4) WRITE (6,370) A3
      IF (IPRINT.GT.4) WRITE (6,380) (X(I),I=1,NDV)
      NCAL(1)=NCAL(1)+1
      JGOTO=4
      RETURN
140   CONTINUE
      F3=OBJ
      IF (IPRINT.GT.4) WRITE (6,390) F3
150   CONTINUE
      IF (F3.LT.F2) GO TO 190
160   CONTINUE
C     -----------------------------------------------------------------
C     **********        3-POINT CUBIC INTERPOLATION       **********
C     -----------------------------------------------------------------
      II=3
      CALL CNMN04 (II,APP,ZRO,A1,F1,SLOPE,A2,F2,A3,F3,ZRO,ZRO)
      IF (APP.LT.ZRO.OR.APP.GT.A3) GO TO 190
C     -----------------------------------------------------------------
C     UPDATE DESIGN VECTOR AND FUNCTION VALUE.
C     -----------------------------------------------------------------
      AP1=APP
      AP=APP-ALP
      ALP=APP
      DO 170 I=1,NDV
170   X(I)=X(I)+AP*S(I)
      IF (IPRINT.GT.4) WRITE (6,370) ALP
      IF (IPRINT.GT.4) WRITE (6,380) (X(I),I=1,NDV)
      NCAL(1)=NCAL(1)+1
      JGOTO=5
      RETURN
180   CONTINUE
      IF (IPRINT.GT.4) WRITE (6,390) OBJ
C     -----------------------------------------------------------------
C                           CHECK CONVERGENCE
C     -----------------------------------------------------------------
      AA=1.-APP/A2
      AB2=ABS(F2)
      AB3=ABS(OBJ)
      AB=AB2
      IF (AB3.GT.AB) AB=AB3
      IF (AB.LT.1.0E-15) AB=1.0E-15
      AB=(AB2-AB3)/AB
      IF (ABS(AB).LT.1.0E-15.AND.ABS(AA).LT..001) GO TO 330
      A4=A3
      F4=F3
      A3=APP
      F3=OBJ
      IF (A3.GT.A2) GO TO 230
      A3=A2
      F3=F2
      A2=APP
      F2=OBJ
      GO TO 230
190   CONTINUE
C     -----------------------------------------------------------------
C     **********        4-POINT CUBIC INTERPOLATION       **********
C     -----------------------------------------------------------------
200   CONTINUE
      A4=2.*A3
C     UPDATE DESIGN VECTOR AND FUNCTION VALUE.
      AP=A4-ALP
      ALP=A4
      DO 210 I=1,NDV
210   X(I)=X(I)+AP*S(I)
```

```
      IF (IPRINT.GT.4) WRITE (6,370) ALP
      IF (IPRINT.GT.4) WRITE (6,380) (X(I),I=1,NDV)
      NCAL(1)=NCAL(1)+1
      JGOTO=6
      RETURN
220   CONTINUE
      F4=OBJ
      IF (IPRINT.GT.4) WRITE (6,390) F4
      IF (F4.GT.F3) GO TO 230
      A1=A2
      F1=F2
      A2=A3
      F2=F3
      A3=A4
      F3=F4
      GO TO 200
230   CONTINUE
      II=4
      CALL CNMN04 (II,APP,A1,A1,F1,SLOPE,A2,F2,A3,F3,A4,F4)
      IF (APP.GT.A1) GO TO 250
      AP=A1-ALP
      ALP=A1
      OBJ=F1
      DO 240 I=1,NDV
240   X(I)=X(I)+AP*S(I)
      GO TO 280
250   CONTINUE
C     ------------------------------------------------------------------
C                 UPDATE DESIGN VECTOR AND FUNCTION VALUE
C     ------------------------------------------------------------------
      AP=APP-ALP
      ALP=APP
      DO 260 I=1,NDV
260   X(I)=X(I)+AP*S(I)
      IF (IPRINT.GT.4) WRITE (6,370) ALP
      IF (IPRINT.GT.4) WRITE (6,380) (X(I),I=1,NDV)
      NCAL(1)=NCAL(1)+1
      JGOTO=7
      RETURN
270   CONTINUE
      IF (IPRINT.GT.4) WRITE (6,390) OBJ
280   CONTINUE
C     ------------------------------------------------------------------
C                     CHECK FOR ILL-CONDITIONING
C     ------------------------------------------------------------------
      IF (OBJ.GT.F2.OR.OBJ.GT.F3) GO TO 290
      IF (OBJ.LE.F1) GO TO 330
      AP=A1-ALP
      ALP=A1
      OBJ=F1
      GO TO 310
290   CONTINUE
      IF (F2.LT.F3) GO TO 300
      OBJ=F3
      AP=A3-ALP
      ALP=A3
      GO TO 310
300   OBJ=F2
      AP=A2-ALP
      ALP=A2
310   CONTINUE
C     ------------------------------------------------------------------
C                       UPDATE DESIGN VECTOR
C     ------------------------------------------------------------------
      DO 320 I=1,NDV
320   X(I)=X(I)+AP*S(I)
330   CONTINUE
C     ------------------------------------------------------------------
C                     CHECK FOR MULTIPLE MINIMA
C     ------------------------------------------------------------------
      IF (OBJ.LE.FFF) GO TO 350
C     INITIAL FUNCTION IS MINIMUM.
      DO 340 I=1,NDV
340   X(I)=X(I)-ALP*S(I)
      ALP=0.
      OBJ=FFF
350   CONTINUE
      JGOTO=0
      RETURN
C     ------------------------------------------------------------------
C                               FORMATS
C     ------------------------------------------------------------------
C
C
360   FORMAT (/////5X,60H* * UNCONSTRAINED ONE-DIMENSIONAL SEARCH INFO
     1RMATION * * *)
370   FORMAT (/5X,7HALPHA =,E14.5/5X,8HX-VECTOR)
380   FORMAT (5X,6E13.5)
390   FORMAT (/5X,5HOBJ =,E14.5)
      END
      SUBROUTINE CNMN04 (II,XBAR,EPS,X1,Y1,SLOPE,X2,Y2,X3,Y3,X4,Y4)
C     ROUTINE TO FIND FIRST XBAR.GE.EPS CORRESPONDING TO A MINIMUM
C     OF A ONE-DIMENSIONAL REAL FUNCTION BY POLYNOMIEL INTERPOLATION.
C     BY G. N. VANDERPLAATS                          APRIL, 1972.
C     NASA-AMES RESEARCH CENTER,  MOFFETT FIELD, CALIF.
C
C     II = CALCULATION CONTROL.
C          1:  2-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, SLOPE,
C              X2 AND Y2.
C          2:  3-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, X2, Y2,
C              X3 AND Y3.
C          3:  3-POINT CUBIC INTERPOLATION, GIVEN X1, Y1, SLOPE, X2, Y2,
C              X3 AND Y3.
C          4:  4-POINT CUBIC INTERPOLATION, GIVEN X1, Y1, X2, Y2, X3,
C              Y3, X4 AND Y4.
C     EPS MAY BE NEGATIVE.
C     IF REQUIRED MINIMUM ON Y DOES NOT EXITS, OR THE FUNCTION IS
C     ILL-CONDITIONED, XBAR = EPS-1.0 WILL BE RETURNED AS AN ERROR
C     INDICATOR.
C     IF DESIRED INTERPOLATION IS ILL-CONDITIONED, A LOWER ORDER
C     INTERPOLATION, CONSISTANT WITH INPUT DATA, WILL BE ATTEMPTED,
C     AND II WILL BE CHANGED ACCORDINGLY.
      XBAR1=EPS-1.
      XBAR=XBAR1
      X21=X2-X1
      IF (ABS(X21).LT.1.0E-20) RETURN
      NSLOP=MOD(II,2)
      GO TO (10,20,40,50),II
10    CONTINUE
C     ------------------------------------------------------------------
```

```
C                 II=1: 2-POINT QUADRATIC INTERPOLATION                 CONM1322
C     ------------------------------------------------------------------CONM1323
      II=1                                                              CONM1324
      DX=X1-X2                                                          CONM1325
      IF (ABS(DX).LT.1.0E-20) RETURN                                    CONM1326
      AA=(SLOPE+(Y2-Y1)/DX)/DX                                          CONM1327
      IF (AA.LT.1.0E-20) RETURN                                         CONM1328
      BB=SLOPE-2.*AA*X1                                                 CONM1329
      XBAR=-.5*BB/AA                                                    CONM1330
      IF (XBAR.LT.EPS) XBAR=XBAR1                                       CONM1331
      RETURN                                                            CONM1332
20    CONTINUE                                                          CONM1333
C     ------------------------------------------------------------------CONM1334
C                 II=2: 3-POINT QUADRATIC INTERPOLATION                 CONM1335
C     ------------------------------------------------------------------CONM1336
      II=2                                                              CONM1337
      X21=X2-X1                                                         CONM1338
      X31=X3-X1                                                         CONM1339
      X32=X3-X2                                                         CONM1340
      QQ=X21*X31*X32                                                    CONM1341
      IF (ABS(QQ).LT.1.0E-20) RETURN                                    CONM1342
      AA=(Y1*X32-Y2*X31+Y3*X21)/QQ                                      CONM1343
      IF (AA.LT.1.0E-20) GO TO 30                                       CONM1344
      BB=(Y2-Y1)/X21-AA*(X1+X2)                                         CONM1345
      XBAR=-.5*BB/AA                                                    CONM1346
      IF (XBAR.LT.EPS) XBAR=XBAR1                                       CONM1347
      RETURN                                                            CONM1348
30    CONTINUE                                                          CONM1349
      IF (NSLOP.EQ.0) RETURN                                            CONM1350
      GO TO 10                                                          CONM1351
40    CONTINUE                                                          CONM1352
C     ------------------------------------------------------------------CONM1353
C                   II=3: 3-POINT CUBIC INTERPOLATION                   CONM1354
C     ------------------------------------------------------------------CONM1355
      II=3                                                              CONM1356
      X21=X2-X1                                                         CONM1357
      X31=X3-X1                                                         CONM1358
      X32=X3-X2                                                         CONM1359
      QQ=X21*X31*X32                                                    CONM1360
      IF (ABS(QQ).LT.1.0E-20) RETURN                                    CONM1361
      X11=X1*X1                                                         CONM1362
      DNOM=X2*X2*X31-X11*X32-X3*X3*X21                                  CONM1363
      IF (ABS(DNOM).LT.1.0E-20) GO TO 20                                CONM1364
      AA=((X31*X31*(Y2-Y1)-X21*X21*(Y3-Y1))/(X31*X21)-SLOPE*X32)/DNOM   CONM1365
      IF (ABS(AA).LT.1.0E-20) GO TO 20                                  CONM1366
      BB=((Y2-Y1)/X21-SLOPE-AA*(X2*X2+X1*X2-2.*X11))/X21                CONM1367
      CC=SLOPE-3.*AA*X11-2.*BB*X1                                       CONM1368
      BAC=BB*BB-3.*AA*CC                                                CONM1369
      IF (BAC.LT.0.) GO TO 20                                           CONM1370
      BAC=SQRT(BAC)                                                     CONM1371
      XBAR=(BAC-BB)/(3.*AA)                                             CONM1372
      IF (XBAR.LT.EPS) XBAR=EPS                                         CONM1373
      RETURN                                                            CONM1374
50    CONTINUE                                                          CONM1375
C     ------------------------------------------------------------------CONM1376
C                   II=4: 4-POINT CUBIC INTERPOLATION                   CONM1377
C     ------------------------------------------------------------------CONM1378
      X21=X2-X1                                                         CONM1379
      X31=X3-X1                                                         CONM1380
      X41=X4-X1                                                         CONM1381
      X32=X3-X2                                                         CONM1382
      X42=X4-X2                                                         CONM1383
      X11=X1*X1                                                         CONM1384
      X22=X2*X2                                                         CONM1385
      X33=X3*X3                                                         CONM1386
      X44=X4*X4                                                         CONM1387
      X111=X1*X11                                                       CONM1388
      X222=X2*X22                                                       CONM1389
      Q2=X31*X21*X32                                                    CONM1390
      IF (ABS(Q2).LT.1.0E-30) RETURN                                    CONM1391
      Q1=X111*X32-X222*X31+X3*X33*X21                                   CONM1392
      Q4=X111*X42-X222*X41+X4*X44*X21                                   CONM1393
      Q5=X41*X21*X42                                                    CONM1394
      DNOM=Q2*Q4-Q1*Q5                                                  CONM1395
      IF (ABS(DNOM).LT.1.0E-30) GO TO 60                                CONM1396
      Q3=Y3*X21-Y2*X31+Y1*X32                                           CONM1397
      Q6=Y4*X21-Y2*X41+Y1*X42                                           CONM1398
      AA=(Q2*Q6-Q3*Q5)/DNOM                                             CONM1399
      BB=(Q3-Q1*AA)/Q2                                                  CONM1400
      CC=(Y2-Y1-AA*(X222-X111))/X21-BB*(X1+X2)                          CONM1401
      BAC=BB*BB-3.*AA*CC                                                CONM1402
      IF (ABS(AA).LT.1.0E-20.OR.BAC.LT.0.) GO TO 60                     CONM1403
      BAC=SQRT(BAC)                                                     CONM1404
      XBAR=(BAC-BB)/(3.*AA)                                             CONM1405
      IF (XBAR.LT.EPS) XBAR=XBAR1                                       CONM1406
      RETURN                                                            CONM1407
60    CONTINUE                                                          CONM1408
      IF (NSLOP.EQ.1) GO TO 40                                          CONM1409
      GO TO 20                                                          CONM1410
      END                                                               CONM1411
      SUBROUTINE CNMN05 (G,DF,A,S,B,C,SLOPE,PHI,ISC,IC,MS1,NVC,N1,N2,N3,CONM1412
     1N4,N5)                                                            CONM1413
      COMMON /CNMN1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAXCONM1414
     1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,ITCONM1415
     2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER                               CONM1416
      DIMENSION DF(N1), G(N2), ISC(N2), IC(N3), A(N1,N3), S(N1), C(N4), CONM1417
     1MS1(N5), B(N3,N3)                                                 CONM1418
C     ROUTINE TO SOLVE DIRECTION FINDING PROBLEM IN MODIFIED METHOD OF  CONM1419
C     FEASIBLE DIRECTIONS.                                              CONM1420
C     BY G. N. VANDERPLAATS                         MAY, 1972.          CONM1421
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.                  CONM1422
C     NORM OF S VECTOR USED HERE IS S-TRANSPOSE TIMES S.LE.1.           CONM1423
C     IF NVC = 0 FIND DIRECTION BY ZOUTENDIJK'S METHOD.  OTHERWISE      CONM1424
C     FIND MODIFIED DIRECTION.                                          CONM1425
C     ------------------------------------------------------------------CONM1426
C     ***  NORMALIZE GRADIENTS, CALCULATE THETA'S AND DETERMINE NVC  ***CONM1427
C     ------------------------------------------------------------------CONM1428
      NDV1=NDV+1                                                        CONM1429
      NDV2=NDV+2                                                        CONM1430
      NAC1=NAC+1                                                        CONM1431
      NVC=0                                                             CONM1432
      THMAX=0.                                                          CONM1433
      CTA=ABS(CT)                                                       CONM1434
      CT1=1./CTA                                                        CONM1435
      CTAM=ABS(CTMIN)                                                   CONM1436
      CTB=ABS(CTL)                                                      CONM1437
      CT2=1./CTB                                                        CONM1438
      CTBM=ABS(CTLMIN)                                                  CONM1439
      A1=1.                                                             CONM1440
      DO 40 I=1,NAC                                                     CONM1441
```

```
C     CALCULATE THETA
      NCI=IC(I)
      NCJ=1
      IF (NCI.LE.NCON) NCJ=ISC(NCI)
      C1=G(NCI)
      CTD=CT1
      CTC=CTAM
      IF (NCJ.LE.0) GO TO 10
      CTC=CTBM
      CTD=CT2
10    IF (C1.GT.CTC) NVC=NVC+1
      THT=0.
      GG=1.+CTD*C1
      IF (NCJ.EQ.0.OR.C1.GT.CTC) THT=THETA*GG*GG
      IF (THT.GT.50.) THT=50.
      IF (THT.GT.THMAX) THMAX=THT
      A(NDV1,I)=THT
C     ------------------------------------------------------------------
C                   NORMALIZE GRADIENTS OF CONSTRAINTS
C     ------------------------------------------------------------------
      A(NDV2,I)=1.
      IF (NCI.GT.NCON) GO TO 40
      A1=0.
      DO 20 J=1,NDV
      A1=A1+A(J,I)**2
20    CONTINUE
      IF (A1.LT.1.0E-20) A1=1.0E-20
      A1=SQRT(A1)
      A(NDV2,I)=A1
      A1=1./A1
      DO 30 J=1,NDV
30    A(J,I)=A1*A(J,I)
40    CONTINUE
C     ------------------------------------------------------------------
C     NORMALIZE GRADIENT OF OBJECTIVE FUNCTION AND STORE IN NAC+1
C     COLUMN OF A
C     ------------------------------------------------------------------
      A1=0.
      DO 50 I=1,NDV
      A1=A1+DF(I)**2
50    CONTINUE
      IF (A1.LT.1.0E-20) A1=1.0E-20
      A1=SQRT(A1)
      A1=1./A1
      DO 60 I=1,NDV
60    A(I,NAC1)=A1*DF(I)
C     BUILD C VECTOR.
      IF (NVC.GT.0) GO TO 80
C     ------------------------------------------------------------------
C                  BUILD C FOR CLASSICAL METHOD
C     ------------------------------------------------------------------
      NDB=NAC1
      A(NDV1,NDB)=1.
      DO 70 I=1,NDB
70    C(I)=-A(NDV1,I)
      GO TO 110
80    CONTINUE
C     ------------------------------------------------------------------
C                   BUILD C FOR MODIFIED METHOD
C     ------------------------------------------------------------------
      NDB=NAC
      A(NDV1,NAC1)=-PHI
C     ------------------------------------------------------------------
C         SCALE THETA'S SO THAT MAXIMUM THETA IS UNITY
C     ------------------------------------------------------------------
      IF (THMAX.GT.0.00001) THMAX=1./THMAX
      DO 90 I=1,NDB
      A(NDV1,I)=A(NDV1,I)*THMAX
90    CONTINUE
      DO 100 I=1,NDB
      C(I)=0.
      DO 100 J=1,NDV1
100   C(I)=C(I)+A(J,I)*A(J,NAC1)
110   CONTINUE
C     ------------------------------------------------------------------
C                        BUILD B MATRIX
C     ------------------------------------------------------------------
      DO 120 I=1,NDB
      DO 120 J=1,NDB
      B(I,J)=0.
      DO 120 K=1,NDV1
120   B(I,J)=B(I,J)-A(K,I)*A(K,J)
C     ------------------------------------------------------------------
C                    SOLVE SPECIAL L. P. PROBLEM
C     ------------------------------------------------------------------
      CALL CNMN08 (NDB,NER,C,MS1,B,N3,N4,N5)
      IF (IPRINT.GT.1.AND.NER.GT.0) WRITE (6,180)
C     CALCULATE RESULTING DIRECTION VECTOR, S.
      SLOPE=0.
C     ------------------------------------------------------------------
C                  USABLE-FEASIBLE DIRECTION
C     ------------------------------------------------------------------
      DO 140 I=1,NDV
      S1=0.
      IF (NVC.GT.0) S1=-A(I,NAC1)
      DO 130 J=1,NDB
130   S1=S1-A(I,J)*C(J)
      SLOPE=SLOPE+S1*DF(I)
140   S(I)=S1
      S(NDV1)=1.
      IF (NVC.GT.0) S(NDV1)=-A(NDV1,NAC1)
      DO 150 J=1,NDB
150   S(NDV1)=S(NDV1)-A(NDV1,J)*C(J)
C     ------------------------------------------------------------------
C                 NORMALIZE S TO MAX ABS OF UNITY
C     ------------------------------------------------------------------
      S1=0.
      DO 160 I=1,NDV
      A1=ABS(S(I))
      IF (A1.GT.S1) S1=A1
160   CONTINUE
      IF (S1.LT.1.0E-10) RETURN
      S1=1./S1
      DO 170 I=1,NDV
170   S(I)=S1*S(I)
      SLOPE=S1*SLOPE
      S(NDV1)=S1*S(NDV1)
      RETURN
C     ------------------------------------------------------------------
C                             FORMATS
```

```
C     ----------------------------------------------------------------CONM1562
C                                                                     CONM1563
C                                                                     CONM1564
180   FORMAT (//5X,46H* * DIRECTION FINDING PROCESS DID NOT CONVERGE/5X,CONM1565
     129H* * S-VECTOR MAY NOT BE VALID)                               CONM1566
      END                                                             CONM1567
      SUBROUTINE CNMN06 (X,VLB,VUB,G,SCAL,DF,S,G1,G2,CTAM,CTBM,SLOPE,ALPCONM1568
     1,A2,A3,A4,F1,F2,F3,CV1,CV2,CV3,CV4,ALPCA,ALPFES,ALPLN,ALPMIN,ALPNCCONM1569
     2,ALPSAV,ALPSID,ALPTOT,ISC,N1,N2,NCAL,NVC,ICOUNT,IGOOD1,IGOOD2,IGOOCONM1570
     3D3,IGOOD4,IBEST,III,NLNC,JGOTO)                                 CONM1571
      COMMON /CNMN1/ DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,ALPHAXCONM1572
     1,ABOBJ1,THETA,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,LINOBJ,ITMAX,ITCONM1573
     2RM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER                             CONM1574
      DIMENSION X(N1), VLB(N1), VUB(N1), G(N2), SCAL(N1), DF(N1), S(N1),CONM1575
     1 G1(N2), G2(N2), ISC(N2), NCAL(2)                               CONM1576
C     ROUTINE TO SOLVE ONE-DIMENSIONAL SEARCH PROBLEM FOR CONSTRAINED CONM1577
C     FUNCTION MINIMIZATION.                                          CONM1578
C     BY G. N. VANDERPLAATS                          AUG., 1974.      CONM1579
C     NASA-AMES RESEARCH CENTER, MOFFETT FIELD, CALIF.                CONM1580
C     OBJ = INITIAL AND FINAL FUNCTION VALUE.                         CONM1581
C     ALP = MOVE PARAMETER.                                           CONM1582
C     SLOPE = INITIAL SLOPE.                                          CONM1583
C                                                                     CONM1584
C     ALPSID = MOVE TO SIDE CONSTRAINT.                               CONM1585
C     ALPFES = MOVE TO FEASIBLE REGION.                               CONM1586
C     ALPNC = MOVE TO NEW NON-LINEAR CONSTRAINT.                      CONM1587
C     ALPLN = MOVE TO LINEAR CONSTRAINT.                              CONM1588
C     ALPCA = MOVE TO RE-ENCOUNTER CURRENTLY ACTIVE CONSTRAINT.       CONM1589
C     ALPMIN = MOVE TO MINIMIZE FUNCTION.                             CONM1590
C     ALPTOT = TOTAL MOVE PARAMETER.                                  CONM1591
      ZRO=0.                                                          CONM1592
      IF (JGOTO.EQ.0) GO TO 10                                        CONM1593
      GO TO (140,310,520),JGOTO                                       CONM1594
10    IF (IPRINT.GE.5) WRITE (6,730)                                  CONM1595
      ALPSAV=ALP                                                      CONM1596
      ICOUNT=0                                                        CONM1597
      ALPTOT=0.                                                       CONM1598
C     TOLERANCES.                                                     CONM1599
      CTAM=ABS(CTMIN)                                                 CONM1600
      CTBM=ABS(CTLMIN)                                                CONM1601
C     PROPOSED MOVE.                                                  CONM1602
20    CONTINUE                                                        CONM1603
C     ----------------------------------------------------------------CONM1604
C     *****  BEGIN SEARCH OR IMPOSE SIDE CONSTRAINT MODIFICATION  ***** CONM1605
C     ----------------------------------------------------------------CONM1606
      A2=ALPSAV                                                       CONM1607
      ICOUNT=ICOUNT+1                                                 CONM1608
      ALPSID=1.0E+20                                                  CONM1609
C     INITIAL ALPHA AND OBJ.                                          CONM1610
      ALP=0.                                                          CONM1611
      F1=OBJ                                                          CONM1612
      KSID=0                                                          CONM1613
      IF (NSIDE.EQ.0) GO TO 70                                        CONM1614
C     ----------------------------------------------------------------CONM1615
C     FIND MOVE TO SIDE CONSTRAINT AND INSURE AGAINST VIOLATION OF    CONM1616
C     SIDE CONSTRAINTS                                                CONM1617
C     ----------------------------------------------------------------CONM1618
      DO 60 I=1,NDV                                                   CONM1619
      SI=S(I)                                                         CONM1620
      IF (ABS(SI).GT.1.0E-20) GO TO 30                                CONM1621
C     ITH COMPONENT OF S IS SMALL.  SET TO ZERO.                      CONM1622
      S(I)=0.                                                         CONM1623
      SLOPE=SLOPE-SI*DF(I)                                            CONM1624
      GO TO 60                                                        CONM1625
30    CONTINUE                                                        CONM1626
      XI=X(I)                                                         CONM1627
      SI=1./SI                                                        CONM1628
      IF (SI.GT.0.) GO TO 40                                          CONM1629
C     LOWER BOUND.                                                    CONM1630
      XI2=VLB(I)                                                      CONM1631
      XI1=ABS(XI2)                                                    CONM1632
      IF (XI1.LT.1.) XI1=1.                                           CONM1633
C     CONSTRAINT VALUE.                                               CONM1634
      GI=(XI2-XI)/XI1                                                 CONM1635
      IF (GI.GT.-1.0E-6) GO TO 50                                     CONM1636
C     PROPOSED MOVE TO LOWER BOUND.                                   CONM1637
      ALPA=(XI2-XI)*SI                                                CONM1638
      IF (ALPA.LT.ALPSID) ALPSID=ALPA                                 CONM1639
      GO TO 60                                                        CONM1640
40    CONTINUE                                                        CONM1641
C     UPPER BOUND.                                                    CONM1642
      XI2=VUB(I)                                                      CONM1643
      XI1=ABS(XI2)                                                    CONM1644
      IF (XI1.LT.1.) XI1=1.                                           CONM1645
C     CONSTRAINT VALUE.                                               CONM1646
      GI=(XI-XI2)/XI1                                                 CONM1647
      IF (GI.GT.-1.0E-6) GO TO 50                                     CONM1648
C     PROPOSED MOVE TO UPPER BOUND.                                   CONM1649
      ALPA=(XI2-XI)*SI                                                CONM1650
      IF (ALPA.LT.ALPSID) ALPSID=ALPA                                 CONM1651
      GO TO 60                                                        CONM1652
50    CONTINUE                                                        CONM1653
C     MOVE WILL VIOLATE SIDE CONSTRAINT.  SET S(I)=0.                 CONM1654
      SLOPE=SLOPE-S(I)*DF(I)                                          CONM1655
      S(I)=0.                                                         CONM1656
      KSID=KSID+1                                                     CONM1657
60    CONTINUE                                                        CONM1658
C     ALPSID IS UPPER BOUND ON ALPHA.                                 CONM1659
      IF (A2.GT.ALPSID) A2=ALPSID                                     CONM1660
70    CONTINUE                                                        CONM1661
C     ----------------------------------------------------------------CONM1662
C                  CHECK ILL-CONDITIONING                             CONM1663
C     ----------------------------------------------------------------CONM1664
      IF (KSID.EQ.NDV.OR.ICOUNT.GT.10) GO TO 710                      CONM1665
      IF (NVC.EQ.0.AND.SLOPE.GT.0.) GO TO 710                         CONM1666
      ALPFES=-1.                                                      CONM1667
      ALPMIN=-1.                                                      CONM1668
      ALPLN=1.1*ALPSID                                                CONM1669
      ALPNC=ALPSID                                                    CONM1670
      ALPCA=ALPSID                                                    CONM1671
      IF (NCON.EQ.0) GO TO 90                                         CONM1672
C     STORE CONSTRAINT VALUES IN G1.                                  CONM1673
      DO 80 I=1,NCON                                                  CONM1674
      G1(I)=G(I)                                                      CONM1675
80    CONTINUE                                                        CONM1676
90    CONTINUE                                                        CONM1677
C     ----------------------------------------------------------------CONM1678
C                     MOVE A DISTANCE A2*S                            CONM1679
C     ----------------------------------------------------------------CONM1680
      ALPTOT=ALPTOT+A2                                                CONM1681
```

```
      DO 100 I=1,NDV
      X(I)=X(I)+A2*S(I)
100   CONTINUE
      IF (IPRINT.LT.5) GO TO 130
      WRITE (6,740) A2
      IF (NSCAL.EQ.0) GO TO 120
      DO 110 I=1,NDV
110   G(I)=SCAL(I)*X(I)
      WRITE (6,750) (G(I),I=1,NDV)
      GO TO 130
120   WRITE (6,750) (X(I),I=1,NDV)
C     ------------------------------------------------------------------
C                  UPDATE FUNCTION AND CONSTRAINT VALUES
C     ------------------------------------------------------------------
130   NCAL(1)=NCAL(1)+1
      JGOTO=1
      RETURN
140   CONTINUE
      F2=OBJ
      IF (IPRINT.GE.5) WRITE (6,760) F2
      IF (IPRINT.LT.5.OR.NCON.EQ.0) GO TO 150
      WRITE (6,770)
      WRITE (6,750) (G(I),I=1,NCON)
150   CONTINUE
C     ------------------------------------------------------------------
C               IDENTIFY ACCAPTABILITY OF DESIGNS F1 AND F2
C     ------------------------------------------------------------------
C     IGOOD = 0 IS ACCAPTABLE.
C     CV = MAXIMUM CONSTRAINT VIOLATION.
      IGOOD1=0
      IGOOD2=0
      CV1=0.
      CV2=0.
      NVC1=0
      IF (NCON.EQ.0) GO TO 170
      DO 160 I=1,NCON
      CC=CTAM
      IF (ISC(I).GT.0) CC=CTBM
      C1=G1(I)-CC
      C2=G(I)-CC
      IF (C2.GT.0.) NVC1=NVC1+1
      IF (C1.GT.CV1) CV1=C1
      IF (C2.GT.CV2) CV2=C2
160   CONTINUE
      IF (CV1.GT.0.) IGOOD1=1
      IF (CV2.GT.0.) IGOOD2=1
170   CONTINUE
      ALP=A2
      OBJ=F2
C     ------------------------------------------------------------------
C     IF F2 VIOLATES FEWER CONSTRAINTS THAN F1 BUT STILL HAS CONSTRAINT
C     VIOLATIONS RETURN
C     ------------------------------------------------------------------
      IF (NVC1.LT.NVC.AND.NVC1.GT.0) GO TO 710
C     ------------------------------------------------------------------
C                IDENTIFY BEST OF DESIGNS F1 ANF F2
C     ------------------------------------------------------------------
C     IBEST CORRESPONDS TO MINIMUM VALUE DESIGN.
C     IF CONSTRAINTS ARE VIOLATED, IBEST CORRESPONDS TO MINIMUM
C     CONSTRAINT VIOLATION.
      IF (IGOOD1.EQ.0.AND.IGOOD2.EQ.0) GO TO 180
C     VIOLATED CONSTRAINTS.  PICK MINIMUM VIOLATION.
      IBEST=1
      IF (CV1.GE.CV2) IBEST=2
      GO TO 190
180   CONTINUE
C     NO CONSTRAINT VIOLATION.  PICK MINIMUM F.
      IBEST=1
      IF (F2.LE.F1) IBEST=2
190   CONTINUE
      II=1
      IF (NCON.EQ.0) GO TO 230
C     ------------------------------------------------------------------
C     *****              2 - POINT INTERPOLATION                   *****
C     ------------------------------------------------------------------
      III=0
200   III=III+1
      C1=G1(III)
      C2=G(III)
      IF (ISC(III).EQ.0) GO TO 210
C     ------------------------------------------------------------------
C                         LINEAR CONSTRAINT
C     ------------------------------------------------------------------
      IF (C1.GE.1.0E-5.AND.C1.LE.CTBM) GO TO 220
      CALL CNMN07 (II,ALP,ZRO,ZRO,C1,A2,C2,ZRO,ZRO)
      IF (ALP.LE.0.) GO TO 220
      IF (C1.GT.CTBM.AND.ALP.GT.ALPFES) ALPFES=ALP
      IF (C1.LT.CTL.AND.ALP.LT.ALPLN) ALPLN=ALP
      GO TO 220
210   CONTINUE
C     ------------------------------------------------------------------
C                       NON-LINEAR CONSTRAINT
C     ------------------------------------------------------------------
      IF (C1.GE.1.0E-5.AND.C1.LE.CTAM) GO TO 220
      CALL CNMN07 (II,ALP,ZRO,ZRO,C1,A2,C2,ZRO,ZRO)
      IF (ALP.LE.0.) GO TO 220
      IF (C1.GT.CTAM.AND.ALP.GT.ALPFES) ALPFES=ALP
      IF (C1.LT.CT.AND.ALP.LT.ALPNC) ALPNC=ALP
220   CONTINUE
      IF (III.LT.NCON) GO TO 200
230   CONTINUE
      IF (LINOBJ.GT.0.OR.SLOPE.GE.0.) GO TO 240
C     CALCULATE ALPHA TO MINIMIZE FUNCTION.
      CALL CNMN04 (II,ALPMIN,ZRO,ZRO,F1,SLOPE,A2,F2,ZRO,ZRO,ZRO,ZRO)
240   CONTINUE
C     ------------------------------------------------------------------
C                           PROPOSED MOVE
C     ------------------------------------------------------------------
C     MOVE AT LEAST FAR ENOUGH TO OVERCOME CONSTRAINT VIOLATIONS.
      A3=ALPFES
C     MOVE TO MINIMIZE FUNCTION.
      IF (ALPMIN.GT.A3) A3=ALPMIN
C     IF A3.LE.0, SET A3 = ALPSID.
      IF (A3.LE.0.) A3=ALPSID
C     LIMIT MOVE TO NEW CONSTRAINT ENCOUNTER.
      IF (A3.GT.ALPNC) A3=ALPNC
      IF (A3.GT.ALPLN) A3=ALPLN
C     MAKE A3 NON-ZERO.
      IF (A3.LE.1.0E-20) A3=1.0E-20
C     IF A3=A2=ALPSID AND F2 IS BEST, GO INVOKE SIDE CONSTRAINT
```

```
C     MODIFICATION.
      ALPB=1.-A2/A3
      ALPA=1.-ALPSID/A3
      JBEST=0
      IF (ABS(ALPB).LT.1.0E-10.AND.ABS(ALPA).LT.1.0E-10) JBEST=1
      IF (JBEST.EQ.1.AND.IBEST.EQ.2) GO TO 20
C     SIDE CONSTRAINT CHECK NOT SATISFIED.
      IF (NCON.EQ.0) GO TO 260
C     STORE CONSTRAINT VALUES IN G2.
      DO 250 I=1,NCON
      G2(I)=G(I)
250   CONTINUE
260   CONTINUE
C     IF A3=A2, SET A3=.9*A2.
      IF (ABS(ALPB).LT.1.0E-10) A3=.9*A2
C     MOVE AT LEAST .01*A2.
      IF (A3.LT.(.01*A2)) A3=.01*A2
C     LIMIT MOVE TO 5.*A2.
      IF (A3.GT.(5.*A2)) A3=5.*A2
C     LIMIT MOVE TO ALPSID.
      IF (A3.GT.ALPSID) A3=ALPSID
C     MOVE A DISTANCE A3*S.
      ALP=A3-A2
      ALPTOT=ALPTOT+ALP
      DO 270 I=1,NDV
      X(I)=X(I)+ALP*S(I)
270   CONTINUE
      IF (IPRINT.LT.5) GO TO 300
      WRITE (6,780)
      WRITE (6,740) A3
      IF (NSCAL.EQ.0) GO TO 290
      DO 280 I=1,NDV
280   G(I)=SCAL(I)*X(I)
      WRITE (6,750) (G(I),I=1,NDV)
      GO TO 300
290   WRITE (6,750) (X(I),I=1,NDV)
300   CONTINUE
C     ------------------------------------------------------------------
C               UPDATE FUNCTION AND CONSTRAINT VALUES
C     ------------------------------------------------------------------
      NCAL(1)=NCAL(1)+1
      JGOTO=2
      RETURN
310   CONTINUE
      F3=OBJ
      IF (IPRINT.GE.5) WRITE (6,760) F3
      IF (IPRINT.LT.5.OR.NCON.EQ.0) GO TO 320
      WRITE (6,770)
      WRITE (6,750) (G(I),I=1,NCON)
320   CONTINUE
C     ------------------------------------------------------------------
C       CALCULATE MAXIMUM CONSTRAINT VIOLATION AND PICK BEST DESIGN
C     ------------------------------------------------------------------
      CV3=0.
      IGOOD3=0
      NVC1=0
      IF (NCON.EQ.0) GO TO 340
      DO 330 I=1,NCON
      CC=CTAM
      IF (ISC(I).GT.0) CC=CTBM
      C1=G(I)-CC
      IF (C1.GT.CV3) CV3=C1
      IF (C1.GT.0.) NVC1=NVC1+1
330   CONTINUE
      IF (CV3.GT.0.) IGOOD3=1
340   CONTINUE
C     DETERMINE BEST DESIGN.
      IF (IBEST.EQ.2) GO TO 360
C     CHOOSE BETWEEN F1 AND F3.
      IF (IGOOD1.EQ.0.AND.IGOOD3.EQ.0) GO TO 350
      IF (CV1.GE.CV3) IBEST=3
      GO TO 380
350   IF (F3.LE.F1) IBEST=3
      GO TO 380
360   CONTINUE
C     CHOOSE BETWEEN F2 AND F3.
      IF (IGOOD2.EQ.0.AND.IGOOD3.EQ.0) GO TO 370
      IF (CV2.GE.CV3) IBEST=3
      GO TO 380
370   IF (F3.LE.F2) IBEST=3
380   CONTINUE
      ALP=A3
      OBJ=F3
C     IF F3 VIOLATES FEWER CONSTRAINTS THAN F1 RETURN.
      IF (NVC1.LT.NVC) GO TO 710
C     IF OBJECTIVE AND ALL CONSTRAINTS ARE LINEAR, RETURN.
      IF (LINOBJ.NE.0.AND.NLNC.EQ.NCON) GO TO 710
C     IF A3 = ALPLN AND F3 IS BOTH GOOD AND BEST RETURN.
      ALPB=1.-ALPLN/A3
      IF ((ABS(ALPB).LT.1.0E-20.AND.IBEST.EQ.3).AND.(IGOOD3.EQ.0)) GO TO
     1 710
C     IF A3 = ALPSID AND F3 IS BEST, GO INVOKE SIDE CONSTRAINT
C     MODIFICATION.
      ALPA=1.-ALPSID/A3
      IF (ABS(ALPA).LT.1.0E-20.AND.IBEST.EQ.3) GO TO 20
C     ------------------------------------------------------------------
C     **********            3 - POINT INTERPOLATION            *********
C     ------------------------------------------------------------------
      ALPNC=ALPSID
      ALPCA=ALPSID
      ALPFES=-1.
      ALPMIN=-1.
      IF (NCON.EQ.0) GO TO 440
      III=0
390   III=III+1
      C1=G1(III)
      C2=G2(III)
      C3=G(III)
      IF (ISC(III).EQ.0) GO TO 400
C     ------------------------------------------------------------------
C     LINEAR CONSTRAINT.  FIND ALPFES ONLY.  ALPLN SAME AS BEFORE.
C     ------------------------------------------------------------------
      IF (C1.LE.CTBM) GO TO 430
      II=1
      CALL CNMN07 (II,ALP,ZRO,ZRO,C1,A3,C3,ZRO,ZRO)
      IF (ALP.GT.ALPFES) ALPFES=ALP
      GO TO 430
400   CONTINUE
C     ------------------------------------------------------------------
C                        NON-LINEAR CONSTRAINT
```

```
C     ------------------------------------------------------------------
      II=2
      CALL CNMN07 (II,ALP,ZRO,ZRO,C1,A2,C2,A3,C3)
      IF (ALP.LE.ZRO) GO TO 430
      IF (C1.GE.CT.AND.C1.LE.0.) GO TO 410
      IF (C1.GT.CTAM.OR.C1.LT.0.) GO TO 420
C     ALP IS MINIMUM MOVE.  UPDATE FOR NEXT CONSTRAINT ENCOUNTER.
410   ALPA=ALP
      CALL CNMN07 (II,ALP,ALPA,ZRO,C1,A2,C2,A3,C3)
      IF (ALP.LT.ALPCA.AND.ALP.GE.ALPA) ALPCA=ALP
      GO TO 430
420   CONTINUE
      IF (ALP.GT.ALPFES.AND.C1.GT.CTAM) ALPFES=ALP
      IF (ALP.LT.ALPNC.AND.C1.LT.0.) ALPNC=ALP
430   CONTINUE
      IF (III.LT.NCON) GO TO 390
440   CONTINUE
      IF (LINOBJ.GT.0.OR.SLOPE.GT.0.) GO TO 450
C     ------------------------------------------------------------------
C              CALCULATE ALPHA TO MINIMIZE FUNCTION
C     ------------------------------------------------------------------
      II=3
      IF (A2.GT.A3.AND.(IGOOD2.EQ.0.AND.IBEST.EQ.2)) II=2
      CALL CNMN04 (II,ALPMIN,ZRO,ZRO,F1,SLOPE,A2,F2,A3,F3,ZRO,ZRO)
450   CONTINUE
C     ------------------------------------------------------------------
C                          PROPOSED MOVE
C     ------------------------------------------------------------------
C     MOVE AT LEAST ENOUGH TO OVERCOME CONSTRAINT VIOLATIONS.
      A4=ALPFES
C     MOVE TO MINIMIZE FUNCTION.
      IF (ALPMIN.GT.A4) A4=ALPMIN
C     IF A4.LE.0, SET A4 = ALPSID.
      IF (A4.LE.0.) A4=ALPSID
C     LIMIT MOVE TO NEW CONSTRAINT ENCOUNTER.
      IF (A4.GT.ALPLN) A4=ALPLN
      IF (A4.GT.ALPNC) A4=ALPNC
C     LIMIT MOVE TO RE-ENCOUNTER CURRENTLY ACTIVE CONSTRAINT.
      IF (A4.GT.ALPCA) A4=ALPCA
C     LIMIT A4 TO 5.*A3.
      IF (A4.GT.(5.*A3)) A4=5.*A3
C     UPDATE DESIGN.
      IF (IBEST.NE.3.OR.NCON.EQ.0) GO TO 470
C     STORE CONSTRAINT VALUES IN G2.  F3 IS BEST.  F2 IS NOT.
      DO 460 I=1,NCON
      G2(I)=G(I)
460   CONTINUE
470   CONTINUE
C     IF A4=A3 AND IGOOD1=0 AND IGOOD3=1, SET A4=.9*A3.
      ALP=A4-A3
      IF ((IGOOD1.EQ.0.AND.IGOOD3.EQ.1).AND.ABS(ALP).LT.1.0E-20) A4=.9*A
     13
C     ------------------------------------------------------------------
C                    MOVE A DISTANCE A4*S
C     ------------------------------------------------------------------
      ALP=A4-A3
      ALPTOT=ALPTOT+ALP
      DO 480 I=1,NDV
      X(I)=X(I)+ALP*S(I)
480   CONTINUE
      IF (IPRINT.LT.5) GO TO 510
      WRITE (6,720)
      WRITE (6,740) A4
      IF (NSCAL.EQ.0) GO TO 500
      DO 490 I=1,NDV
490   G(I)=SCAL(I)*X(I)
      WRITE (6,750) (G(I),I=1,NDV)
      GO TO 510
500   WRITE (6,750) (X(I),I=1,NDV)
510   CONTINUE
C     ------------------------------------------------------------------
C              UPDATE FUNCTION AND CONSTRAINT VALUES
C     ------------------------------------------------------------------
      NCAL(1)=NCAL(1)+1
      JGOTO=3
      RETURN
520   CONTINUE
      F4=OBJ
      IF (IPRINT.GE.5) WRITE (6,760) F4
      IF (IPRINT.LT.5.OR.NCON.EQ.0) GO TO 530
      WRITE (6,770)
      WRITE (6,750) (G(I),I=1,NCON)
530   CONTINUE
C     DETERMINE ACCAPTABILITY OF F4.
      IGOOD4=0
      CV4=0.
      IF (NCON.EQ.0) GO TO 550
      DO 540 I=1,NCON
      CC=CTAM
      IF (ISC(I).GT.0) CC=CTBM
      C1=G(I)-CC
      IF (C1.GT.CV4) CV4=C1
540   CONTINUE
      IF (CV4.GT.0.) IGOOD4=1
550   CONTINUE
      ALP=A4
      OBJ=F4
C     ------------------------------------------------------------------
C                       DETERMINE BEST DESIGN
C     ------------------------------------------------------------------
      GO TO (560,610,660),IBEST
560   CONTINUE
C     CHOOSE BETWEEN F1 AND F4.
      IF (IGOOD1.EQ.0.AND.IGOOD4.EQ.0) GO TO 570
      IF (CV1.GT.CV4) GO TO 710
      GO TO 580
570   CONTINUE
      IF (F4.LE.F1) GO TO 710
580   CONTINUE
C     F1 IS BEST.
      ALPTOT=ALPTOT-A4
      OBJ=F1
      DO 590 I=1,NDV
      X(I)=X(I)-A4*S(I)
590   CONTINUE
      IF (NCON.EQ.0) GO TO 710
      DO 600 I=1,NCON
      G(I)=G1(I)
600   CONTINUE
      GO TO 710
```

```
  610 CONTINUE
C     CHOOSE BETWEEN F2 AND F4.
      IF (IGOOD2.EQ.0.AND.IGOOD4.EQ.0) GO TO 620
      IF (CV2.GT.CV4) GO TO 710
      GO TO 630
  620 CONTINUE
      IF (F4.LE.F2) GO TO 710
  630 CONTINUE
C     F2 IS BEST.
      OBJ=F2
      A2=A4-A2
      ALPTOT=ALPTOT-A2
      DO 640 I=1,NDV
      X(I)=X(I)-A2*S(I)
  640 CONTINUE
      IF (NCON.EQ.0) GO TO 710
      DO 650 I=1,NCON
      G(I)=G2(I)
  650 CONTINUE
      GO TO 710
  660 CONTINUE
C     CHOOSE BETWEEN F3 AND F4.
      IF (IGOOD3.EQ.0.AND.IGOOD4.EQ.0) GO TO 670
      IF (CV3.GT.CV4) GO TO 710
      GO TO 680
  670 CONTINUE
      IF (F4.LE.F3) GO TO 710
  680 CONTINUE
C     F3 IS BEST.
      OBJ=F3
      A3=A4-A3
      ALPTOT=ALPTOT-A3
      DO 690 I=1,NDV
      X(I)=X(I)-A3*S(I)
  690 CONTINUE
      IF (NCON.EQ.0) GO TO 710
      DO 700 I=1,NCON
      G(I)=G2(I)
  700 CONTINUE
  710 CONTINUE
      ALP=ALPTOT
      IF (IPRINT.GE.5) WRITE (6,790)
      JGOTO=0
      RETURN
C     ------------------------------------------------------------------
C                                  FORMATS
C     ------------------------------------------------------------------
C
C
  720 FORMAT (/5X,25HTHREE-POINT INTERPOLATION)
  730 FORMAT (/////58H* * CONSTRAINED ONE-DIMENSIONAL SEARCH INFORMATI
     1ON * * *)
  740 FORMAT (//5X,15HPROPOSED DESIGN/5X,7HALPHA =,E12.5/5X,8HX-VECTOR)
  750 FORMAT (1X,8E12.4)
  760 FORMAT (/5X,5HOBJ =,E13.5)
  770 FORMAT (/5X,17HCONSTRAINT VALUES)
  780 FORMAT (/5X,23HTWO-POINT INTERPOLATION)
  790 FORMAT (/5X,35H* * * END OF ONE-DIMENSIONAL SEARCH)
      END
      SUBROUTINE CNMN07 (II,XBAR,EPS,X1,Y1,X2,Y2,X3,Y3)
```

```
C     ROUTINE TO FIND FIRST XBAR.GE.EPS CORRESPONDING TO A REAL ZERO
C     OF A ONE-DIMENSIONAL FUNCTION BY POLYNOMIEL INTERPOLATION.
C     BY G. N. VANDERPLAATS                         APRIL, 1972.
C     NASA-AMES RESEARCH CENTER,  MOFFETT FIELD, CALIF.
C     II = CALCULATION CONTROL.
C          1:  2-POINT LINEAR INTERPOLATION, GIVEN X1, Y1, X2 AND Y2.
C          2:  3-POINT QUADRATIC INTERPOLATION, GIVEN X1, Y1, X2, Y2,
C              X3 AND Y3.
C     EPS MAY BE NEGATIVE.
C     IF REQUIRED ZERO ON Y DOES NOT EXITS, OR THE FUNCTION IS
C     ILL-CONDITIONED, XBAR = EPS-1.0 WILL BE RETURNED AS AN ERROR
C     INDICATOR.
C     IF DESIRED INTERPOLATION IS ILL-CONDITIONED, A LOWER ORDER
C     INTERPOLATION, CONSISTANT WITH INPUT DATA, WILL BE ATTEMPTED AND
C     II WILL BE CHANGED ACCORDINGLY.
      XBAR1=EPS-1.
      XBAR=XBAR1
      JJ=0
      X21=X2-X1
      IF (ABS(X21).LT.1.0E-20) RETURN
      IF (II.EQ.2) GO TO 30
C
   10 CONTINUE
C     ------------------------------------------------------------------
C                   II=1: 2-POINT LINEAR INTERPOLATION
C     ------------------------------------------------------------------
      II=1
      YY=Y1*Y2
      IF (JJ.EQ.0.OR.YY.LT.0.) GO TO 20
C     INTERPOLATE BETWEEN X2 AND X3.
      DY=Y3-Y2
      IF (ABS(DY).LT.1.0E-20) GO TO 20
      XBAR=X2+Y2*(X2-X3)/DY
      IF (XBAR.LT.EPS) XBAR=XBAR1
      RETURN
   20 DY=Y2-Y1
C     INTERPOLATE BETWEEN X1 AND X2.
      IF (ABS(DY).LT.1.0E-20) RETURN
      XBAR=X1+Y1*(X1-X2)/DY
      IF (XBAR.LT.EPS) XBAR=XBAR1
      RETURN
   30 CONTINUE
C     ------------------------------------------------------------------
C                  II=2: 3-POINT QUADRATIC INTERPOLATION
C     ------------------------------------------------------------------
      JJ=1
      X31=X3-X1
      X32=X3-X2
      QQ=X21*X31*X32
      IF (ABS(QQ).LT.1.0E-20) RETURN
      AA=(Y1*X32-Y2*X31+Y3*X21)/QQ
      IF (ABS(AA).LT.1.0E-20) GO TO 10
      BB=(Y2-Y1)/X21-AA*(X1+X2)
      CC=Y1-X1*(AA*X1+BB)
      BAC=BB*BB-4.*AA*CC
      IF (BAC.LT.0.) GO TO 10
      BAC=SQRT(BAC)
      AA=.5/AA
      XBAR=AA*(BAC-BB)
      XB2=-AA*(BAC+BB)
```

```
      IF (XBAR.LT.EPS) XBAR=XB2
      IF (XB2.LT.XBAR.AND.XB2.GT.EPS) XBAR=XB2
      IF (XBAR.LT.EPS) XBAR=XBAR1
      RETURN
      END
      SUBROUTINE CNMN08 (NDB,NER,C,MS1,B,N3,N4,N5)
      DIMENSION C(N4), B(N3,N3), MS1(N5)
C     ROUTINE TO SOLVE SPECIAL LINEAR PROBLEM FOR IMPOSING S-TRANSPOSE
C     TIMES S.LE.1 BOUNDS IN THE MODIFIED METHOD OF FEASIBLE DIRECTIONS.
C     BY G. N. VANDERPLAATS                         APRIL, 1972.
C     NASA-AMES RESEARCH CENTER,  MOFFETT FIELD, CALIF.
C     REF.  'STRUCTURAL OPTIMIZATION BY METHODS OF FEASIBLE DIRECTIONS',
C     G. N. VANDERPLAATS AND F. MOSES, JOURNAL OF COMPUTERS
C     AND STRUCTURES, VOL 3, PP 739-755, 1973.
C     FORM OF L. P. IS BX=C WHERE 1ST NDB COMPONENTS OF X CONTAIN VECTOR
C     U AND LAST NDB COMPONENTS CONTAIN VECTOR V.  CONSTRAINTS ARE
C     U.GE.0, V.GE.0, AND U-TRANSPOSE TIMES V = 0.
C     NER = ERROR FLAG.  IF NER.NE.0 ON RETURN, PROCESS HAS NOT
C     CONVERGED IN 5*NDB ITERATIONS.
C     VECTOR MS1 IDENTIFIES THE SET OF BASIC VARIABLES.
C     ------------------------------------------------------------------
C     CHOOSE INITIAL BASIC VARIABLES AS V, AND INITIALIZE VECTOR MS1
C     ------------------------------------------------------------------
      NER=1
      M2=2*NDB
C     CALCULATE CBMIN AND EPS AND INITIALIZE MS1.
      EPS=-1.0E+10
      CBMIN=0.
      DO 10 I=1,NDB
      BI=B(I,I)
      CBMAX=0.
      IF (BI.LT.-1.0E-6) CBMAX=C(I)/BI
      IF (BI.GT.EPS) EPS=BI
      IF (CBMAX.GT.CBMIN) CBMIN=CBMAX
10    MS1(I)=0
      EPS=.0001*EPS
      IF (EPS.LT.-1.0E-10) EPS=-1.0E-10
      IF (EPS.GT.-.0001) EPS=-.0001
      CBMIN=CBMIN*1.0E-6
      IF (CBMIN.LT.1.0E-10) CBMIN=1.0E-10
      ITER1=0
      NMAX=5*NDB
C     ------------------------------------------------------------------
C     **********             BEGIN NEW ITERATION              **********
C     ------------------------------------------------------------------
20    ITER1=ITER1+1
      IF (ITER1.GT.NMAX) RETURN
C     FIND MAX. C(I)/B(I,I) FOR I=1,NDB.
      CBMAX=.9*CBMIN
      ICHK=0
      DO 30 I=1,NDB
      C1=C(I)
      BI=B(I,I)
      IF (BI.GT.EPS.OR.C1.GT.0.) GO TO 30
      CB=C1/BI
      IF (CB.LE.CBMAX) GO TO 30
      ICHK=I
      CBMAX=CB
30    CONTINUE
      IF (CBMAX.LT.CBMIN) GO TO 70
      IF (ICHK.EQ.0) GO TO 70
C     UPDATE VECTOR MS1.
      JJ=ICHK
      IF (MS1(JJ).EQ.0) JJ=ICHK+NDB
      KK=JJ+NDB
      IF (KK.GT.M2) KK=JJ-NDB
      MS1(KK)=ICHK
      MS1(JJ)=0
C     ------------------------------------------------------------------
C                       PIVOT OF B(ICHK,ICHK)
C     ------------------------------------------------------------------
      BB=1./B(ICHK,ICHK)
      DO 40 J=1,NDB
40    B(ICHK,J)=BB*B(ICHK,J)
      C(ICHK)=CBMAX
      B(ICHK,ICHK)=BB
C     ELIMINATE COEFICIENTS ON VARIABLE ENTERING BASIS AND STORE
C     COEFICIENTS ON VARIABLE LEAVING BASIS IN THEIR PLACE.
      DO 60 I=1,NDB
      IF (I.EQ.ICHK) GO TO 60
      BB1=B(I,ICHK)
      B(I,ICHK)=0.
      DO 50 J=1,NDB
50    B(I,J)=B(I,J)-BB1*B(ICHK,J)
      C(I)=C(I)-BB1*CBMAX
60    CONTINUE
      GO TO 20
70    CONTINUE
      NER=0
C     ------------------------------------------------------------------
C     STORE ONLY COMPONENTS OF U-VECTOR IN 'C'.  USE B(I,1) FOR
C     TEMPORARY STORAGE
C     ------------------------------------------------------------------
      DO 80 I=1,NDB
      B(I,1)=C(I)
80    CONTINUE
      DO 90 I=1,NDB
      C(I)=0.
      J=MS1(I)
      IF (J.GT.0) C(I)=B(J,1)
      IF (C(I).LT.0.) C(I)=0.
90    CONTINUE
      RETURN
      END
```

APPENDIX C

LIST OF SYMBOLS

A_i	design variable coefficient of profile shape function; Equation (14)
C	blade chord, meters
i	invariant point index; Equation (5); also, index for surface shape functions; Equation (14)
k	dummy index; Equation (11)
M	number of independent flow or geometrical variables to be perturbed; Equation (1)
n	total number of shock points and high-gradient maxima points; Equation (13)
N	total number of invariant points, equal to n + 2; Equation (5)
q_j	*jth* arbitrary geometric or flow parameter to be perturbed; Equation (8)
q_{c_j}	calibration flow value of q_j; Equation (7)
q_{o_j}	base flow value of q_j; Equation (7)
Q	approximate flow solution for arbitrary flow quantity; Equation (1)
Q_{c_j}	calibration flow solution for value q_{c_j} of arbitrary parameter; Equation (3)
Q_o	base flow solution for values q_{o_j} of arbitrary parameters; Equation (1)
Q_{1_j}	*jth* perturbation solution per unit change of perturbed parameter q_j; Equation (3)
s	strained x coordinate; Equation (2)
x	nondimensional blade-fixed orthogonal coordinate; Equation (1), normalized by C
$\bar{x}_j$	nondimensional blade-fixed orthogonal coordinate related to *jth* calibration solution; Equation (3)

x_1 straining function associated with x coordinate; Equation (2)

x_{1_i} straining function associated with *ith* invariant point; Equation (5)

δx_o unit displacement in x direction associated with *ith* invariant point; Equations (6) and (10)

ε_j desired perturbaiton change of *jth* geometric or flow parameter; Equation (8)

$\bar{\varepsilon}_j$ perturbation change of *jth* geometric or flow parameter between base and calibration flows; Equation (7)

Subscripts

i denotes quantities associated with *ith* invariant point

j denotes perturbation quantities

Superscripts

o denotes base flow quantities

c denotes quantities associated with calibration flows

REFERENCES

1. Stahara, S. S., Chaussee, D. S., and Spreiter, J. R.: Perturbation Solutions for Transonic Flow on the Blade-to-Blade Surface of Compressor Blade Rows. NASA CR-2941, Jan. 1978.

2. Stahara, S. S., Elliott, J. P., and Spreiter, J. R.: A Rapid Perturbation Procedure for Determining Nonlinear Flow Solutions: Application to Transonic Turbomachinery Flows. NASA CR-3425, May 1981.

3. Stahara, S. S., Elliott, J. P., and Spreiter, J. R.: Development of a Multiple-Parameter Nonlinear Perturbation Procedure for Transonic Turbomachinery Flows: Preliminary Application to Design/Optimization Problems. NASA CR-3657, Jan. 1983.

4. Van Dyke, M.: Perturbation Methods in Fluid Mechanics. The Parabolic Press, California, 1975.

5. Nixon, D.: Perturbations in Two- and Three-Dimensional Transonic Flows. AIAA J., Vol. 16, Jul. 1978, pp. 699-709.

6. Nixon, D.: Design of Transonic Airfoil Sections Using a Similarity Theory. NASA TM 78521, Oct. 1978.

7. Stahara, S. S., Crisalli, A. J., and Spreiter, J. R.: Evaluation of a Strained Coordinate Perturbation Procedure: Nonlinear Subsonic and Transonic Flows. AIAA Paper No. 80-0339, Jan. 1980.

8. Nixon, D.: Perturbation Methods in Transonic Flow. AIAA Paper No. 80-1367, Jul. 1980.

9. Stahara, S. S., Elliott, J. P., and Spreiter, J. R.: A Rapid Method for the Approximate Determination of Nonlinear Solutions: Application to Aerodynamic Flows. ICAS Paper No. 80-7.5, Oct. 1980.

10. Nixon, D. and McIntosh, S. C., Jr.: Further Observations on the Strained Coordinate Method for Transonic Flows. AIAA J., Vol. 18, Dec. 1980, pp. 1540-1541.

11. Stahara, S. S.: The Rapid Approximate Determination of Nonlinear Solutions: Application to Aerodynamic Flows and Design/Optimization Problems. Transonic Aerodynamics. (Ed. D. Nixon) Vol. 81, Progress in Astronautics and Aeronautics. AIAA, N.Y., 1982, pp. 637-659.

12. Vanderplaats, G. N.: CONMIN – A Fortran Program for Constrained Function Minimization, User's Manual. NASA TMX-62,282, Aug. 1973.

13. Madsen, L. E. and Vanderplaats, G. N.: COPES – A Fortran Control Program for Engineering Synthesis. NPS 69-81-003, Mar. 1982.

14. Hicks, R. M. and Vanderplaats, G. N.: Application of Numerical Optimization to the Design of Supercritical Airfoils Without Drag-Creep. SAE Paper 770440, 1977.

15. Katsanis, T.: Fortran Program for Calculating Transonic Velocities on a Blade-to-Blade Stream Surface of a Turbomachine. NASA TN D-5427, Sep. 1969.

16. Crouse, J. E.: Computer Program for Definition of Transonic Axial-Flow Compressor Blade Rows. NASA TN D-7345, Feb. 1974.

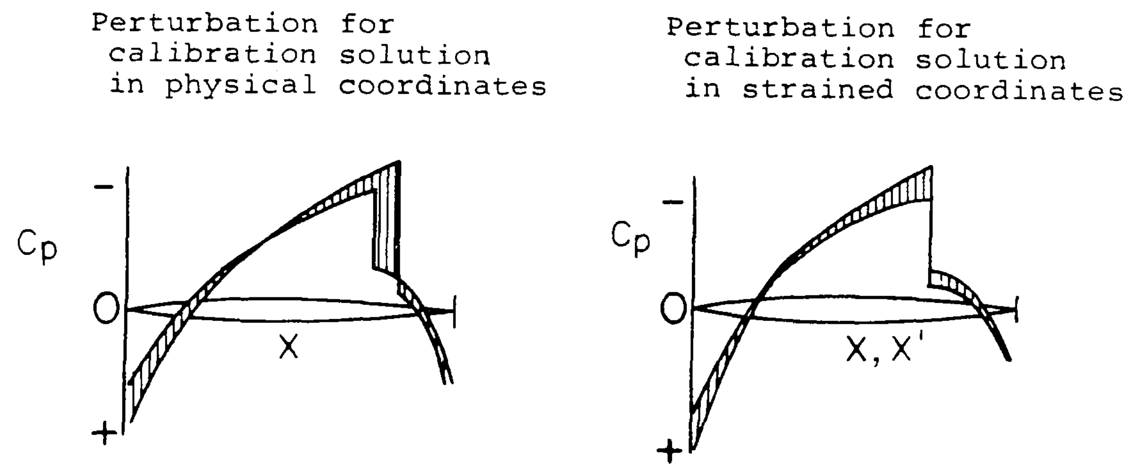
17. Katsanis, T. and McNally, W. D.: Fortran Program for Calculating Velocities and Streamlines on a Blade-to-Blade Stream Surface of a Tandem Blade Turbomachine. NASA TND-5044, 1969.

18. Kennelly, R. T., Jr.: Improved Method for Transonic Airfoil Design-by-Optimization. AIAA Paper No. 83-1864, July 1983.

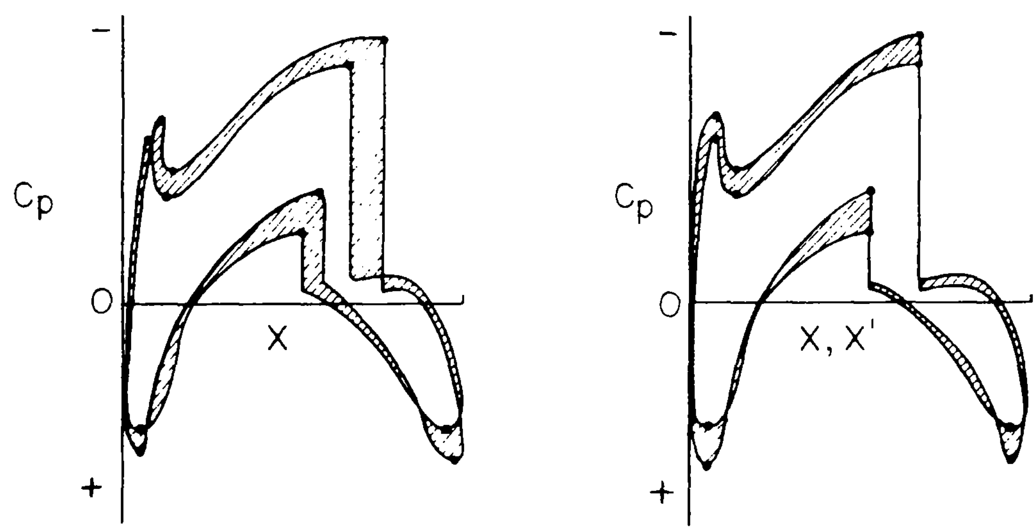
TABLE 1

COMPARISON OF FINAL DESIGN VARIABLES AND OBJECTIVE FUNCTION WHEN EMPLOYING FULL NONLINEAR TSONIC SOLUTIONS OR PERTURBATION METHOD FOR DIFFERENT CHOICES OF CALIBRATION SOLUTION MATRIX FOR SIX DESIGN VARIABLE SUBCRITICAL OPTIMIZATION CASE STUDY USING MAXIMUM SUCTION SURFACE VELOCITY DIFFUSION OBJECTIVE

Design Variables	KOCR	T	ZM	P	TMX	THLE	Objective Function
			INITIAL				
Baseline	-10.0000	0.2500	0.4500	1.50000	0.0500	0.0050	1.8400
Upper Bound	0.0000	0.6000	0.5500	4.0000	0.1000	0.0120	
Lower Bound	-15.0000	0.2000	0.2000	0.5000	0.0300	0.0030	
			FINAL TSONIC SOLUTIONS ONLY RESULTS IOPT=1				
Final	-7.3854	0.2000	0.5500	0.9220	0.0300	0.0060	1.6764
			PERTURBATION SOLUTION RESULTS IOPT=2				
CASE 1							
Calibration	-7.0000	0.2000	0.5500	0.9400	0.0300	0.0060	
Final	-9.1904	0.2342	0.5555	0.9371	0.0358	0.0052	1.6918
CASE 2							
Calibration	-9.0000	0.2350	0.4950	1.6500	0.0550	0.0055	
Final	-8.8994	0.3098	0.5500	0.7463	0.0300	0.0050	1.6829
CASE 3							
Calibration	-11.0000	0.2250	0.4050	1.3500	0.0450	0.0045	
Final	-9.7099	0.3162	0.5500	1.1332	0.0323	0.0055	1.7304
CASE 4							
Calibration	-8.0000	0.2300	0.5000	1.0000	0.0350	0.0058	
Final	-8.8860	0.2814	0.5500	0.9023	0.0370	0.0051	1.6986
CASE 5							
Calibration	-7.0000	0.2000	0.5500	0.9000	0.0300	0.0070	
Final	-9.1846	0.2314	0.5500	0.9397	0.0363	0.0053	1.6909
CASE 6							
Calibration	-11.0000	0.6000	0.2000	4.0000	0.1000	0.0120	
Final	-7.1492	0.3166	0.5500	1.2202	0.0300	0.0046	1.7332
			IOPT=3				
Final	-8.0297	0.2000	0.5500	0.7356	0.0363	0.0039	1.6989



(a) Single shock



(b) Multiple shock and high-gradient locations.

Figure 1.- Illustration of perturbation solution for calibration solution in physical and strained coordinates

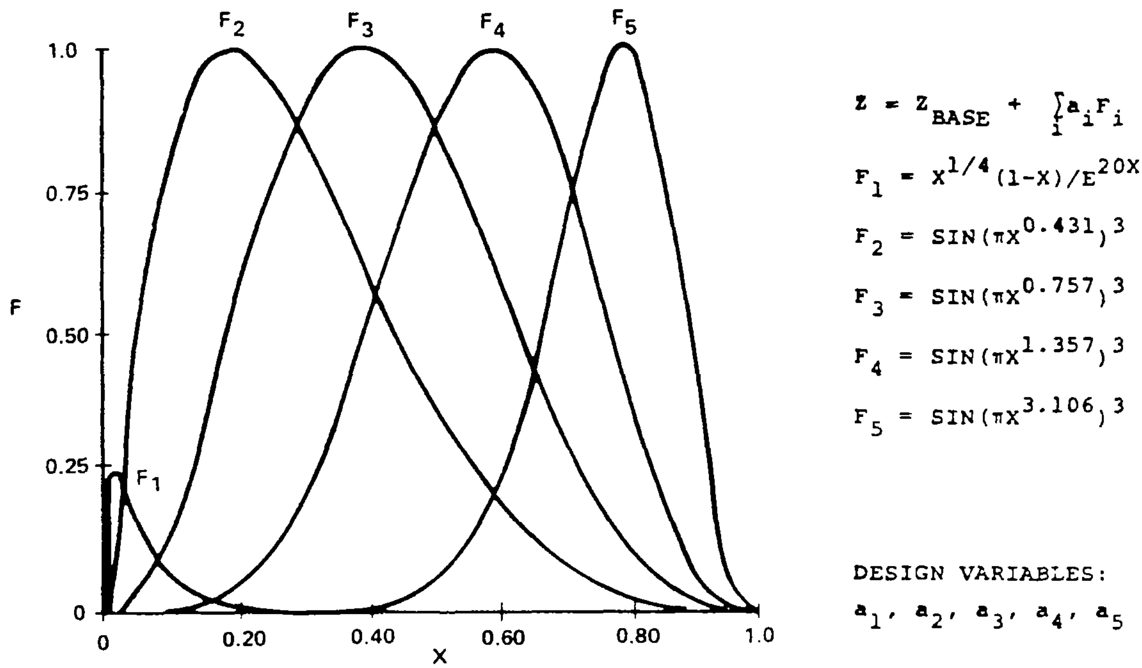


Figure 2.- Illustration of typical ordinate shape functions F_i employed in blade contour alteration optimization problems.

OBJECTIVE IS TO MINIMIZE:

$$\Sigma\left(C_{P_{Predicted}} - C_{P_{Desired}} \right)^2$$

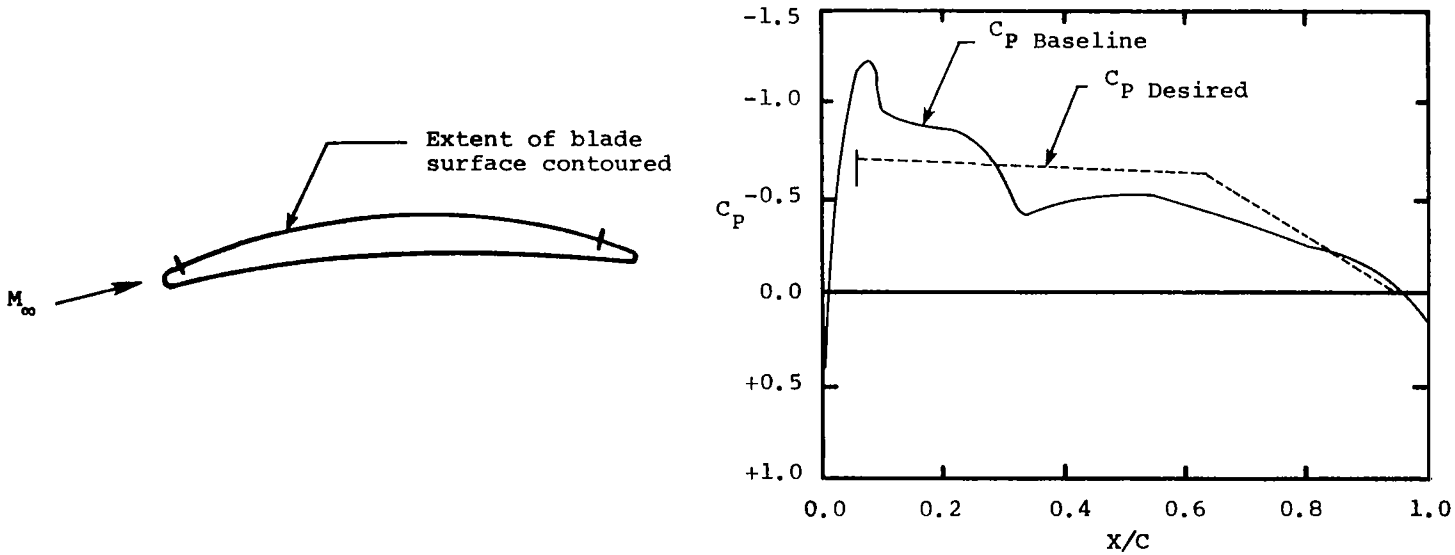


Figure 3.- Illustration of physical basis of optimization problem involving blade surface contouring to tailor the surface pressure distribution to a desired distribution.

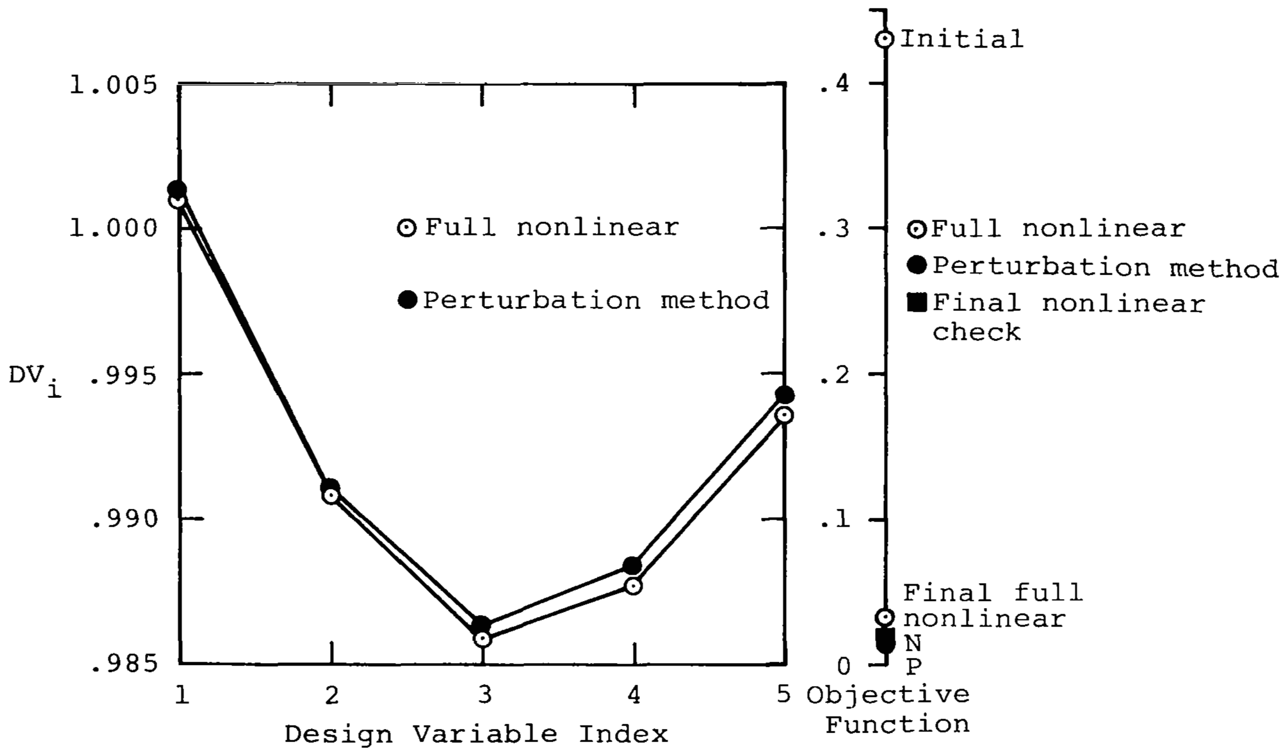


Figure 4.- Comparison of perturbation predicted and full nonlinear results for final design variables and objective function for 5 design variable supercritical case study with surface pressure tailoring objective.

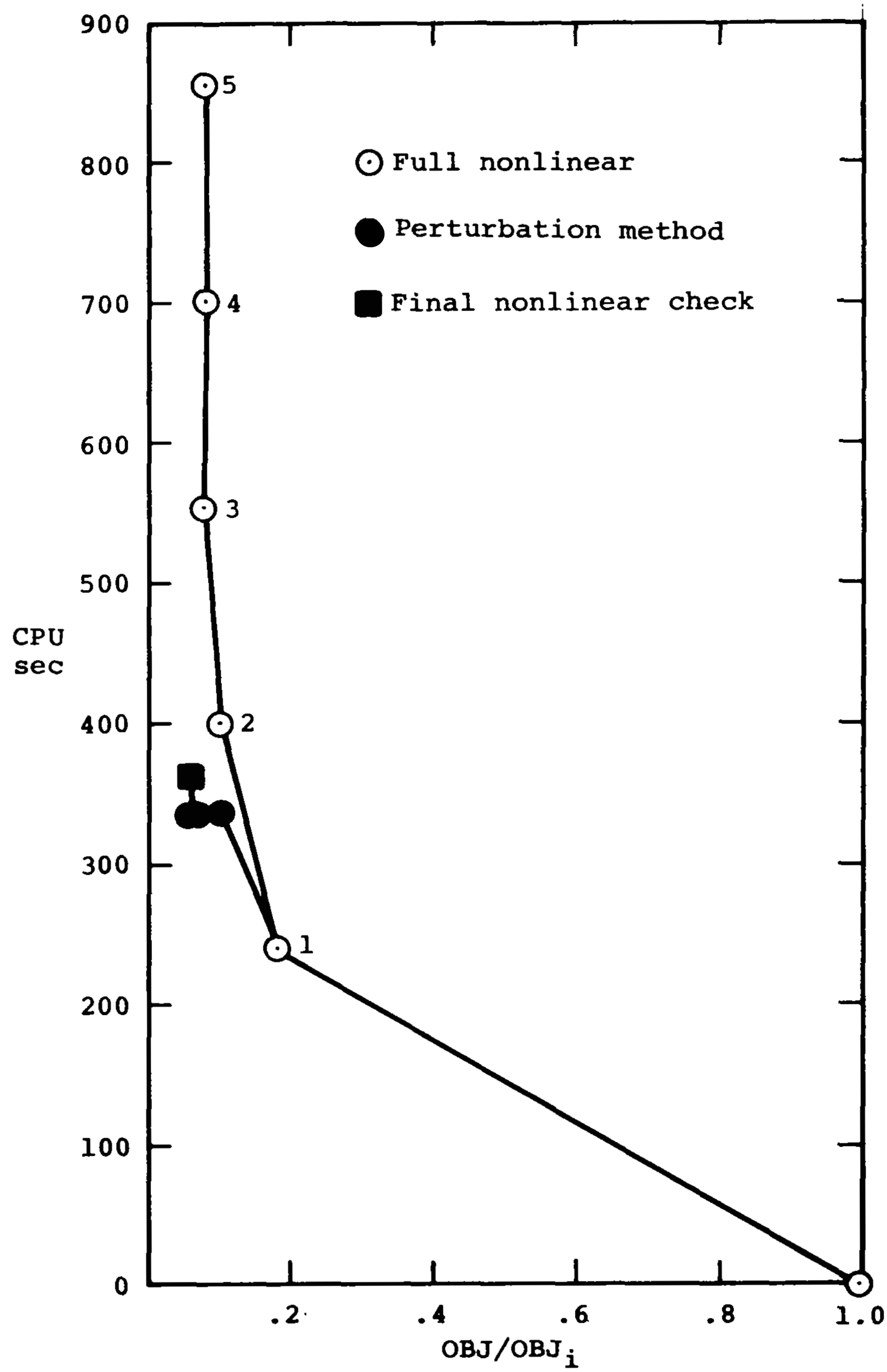


Figure 5.- Comparison of computational work and objective function reduction per optimization search cycle when employing perturbation method after first search cycle (●) or when using full nonlinear solutions (⊙) for a 5 design variable supercritical case study with surface pressure tailoring objective.

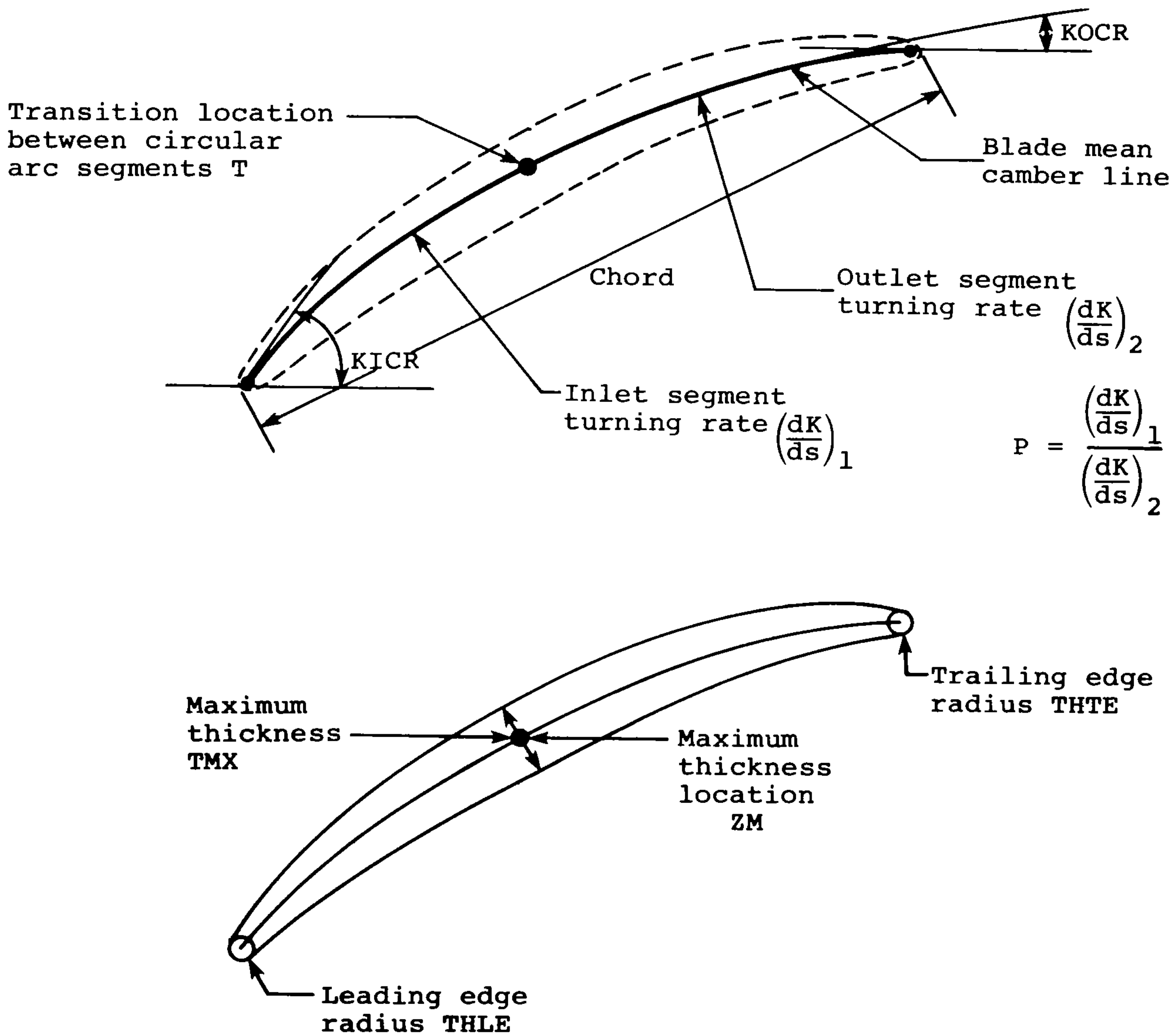


Figure 6.- NASA circular arc blade element layout parameters.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-3831		

4. Title and Subtitle	5. Report Date
Development of a Turbomachinery Design Optimization Procedure Using a Multiple-Parameter Nonlinear Perturbation Method	September 1984
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Stephen S. Stahara	NEAR TR 295
	10. Work Unit No.
9. Performing Organization Name and Address	11. Contract or Grant No.
Nielsen Engineering & Research, Inc. 510 Clyde Avenue Mountain View, California 94043	NAS3-20836
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code 505-31-0A (E-2181)

15. Supplementary Notes

Final report. Project Manager, Aaron Snyder, Fluid Mechanics and Instrumentation Division, NASA Lewis Research Center, Cleveland, Ohio 44135.

16. Abstract

An investigation was carried out to complete the preliminary development of a combined perturbation/optimization procedure and associated computational code for designing optimized blade-to-blade profiles of turbomachinery blades. The overall purpose of the procedures developed in this study is to provide demonstration of a rapid nonlinear perturbation method for minimizing the computational requirements associated with parametric design studies of turbomachinery flows. The method reported here combines the multiple-parameter nonlinear perturbation method, successfully developed in previous phases of this study, with the NASA TSONIC blade-to-blade turbomachinery flow solver, and the COPES-CONMIN optimization procedure into a user's code for designing optimized blade-to-blade surface profiles of turbomachinery blades. Results of several design applications and a documented version of the code together with a user's manual are provided.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Turbomachinery Blade optimization Perturbation method	LIMITED DISTRIBUTION Until September 1985 STAR Category 01

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price*
Unclassified	Unclassified	261	A12

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