Flow Research Report No. 444

NASA Contractor Report 182180

USERS MANUAL FOR AN AERODYNAMIC OPTIMIZATION SCHEME THAT UPDATES FLOW VARIABLES AND DESIGN PARAMETERS SIMULTANEOUSLY

Magdi H. Rizk
Flow Research, Inc.
Kent, Washington 98032

(NASA-CR-182180) USER'S MANUAL FOR AN AERODYNAMIC OPTIMIZATION SCHEME THAT UPDATES FLOW VARIABLES AND DESIGN PARAMETERS SIMULTANEOUSLY Flow Report No. 444 (Flow Research) 33 p CSCL 01A G3/02

N89-13399

Unclas
0174555

July 1988

Prepared for
Lewis Research Center
Under Contract NAS3-24855

NASA
National Aeronautics and
Space Administration

Flow Report No. 444

USER'S MANUAL FOR AN AERODYNAMIC OPTIMIZATION SCHEME THAT UPDATES FLOW VARIABLES AND DESIGN PARAMETERS SIMULTANEOUSLY

Magdi H. Rizk

July 1988

FLOW RESEARCH, INC.
Applied Mechanics Division
21414 68th Ave. So.
Kent, WA 98032

TABLE OF CONTENTS

	Page
INTRODUCTION	1
TECHNICAL BACKGROUND	2
OPTIMIZATION PROGRAM	10
Flow Chart	10
Program Input	10
Dictionary of Input Variables	10
Input Data Format	13
Input Data Example	14
COMMON Blocks	15
Dictionary of COMMON Block Variables	15
COMMON Block/Subroutine Cross Reference	19
Program Subroutines	20
Subroutine Cross Reference	20
Subroutine Descriptions	21
Program Output	26
APPENDIX: LIST OF SYMBOLS	28

INTRODUCTION

Solutions of constrained optimization problems minimize an objective function, E, subject to given constraints. In aerodynamic applications, the objective function and the constraint functions, f_i, i = 1, 2, . . ., depend on the flow field solution, $\underset{\sim}{g}$. The optimization scheme developed here is applicable to situations in which the flow governing equations are nonlinear equations that are solved iteratively.

Conventional optimization methods (e.g., the steepest descent method and the conjugate gradient method) are iterative procedures that require the evaluation of the objective function many times before the converged optimum solution is determined. Since E and f_i are dependent on the flow solution, $\underset{\sim}{g}$, in addition to the vector of design parameters, $\underset{\sim}{P}$, the flow governing equation must be solved each time E and f_i are evaluated. Therefore, the application of conventional optimization schemes to aerodynamic design problems[1-5] leads to two-cycle (inner-outer) iterative procedures. The inner iterative cycle solves the analysis problem for $\underset{\sim}{g}$ iteratively, while the outer cycle determines the the optimum $\underset{\sim}{P}$ iteratively. An alternative to this costly procedure is the approach based on the idea of updating the flow variable iterative solutions and the design parameter iterative solutions simultaneously.

In this manual we present a description of the optimization scheme. The formulation presented is that for optimizing the design of a propeller by maximizing its efficiency, subject to a given power constraint. The Euler equations are assumed to be the flow governing equations. An implicit approximate factorization scheme[6] is used to compute the flow field about the propeller.

TECHNICAL BACKGROUND

The propeller design problem is cast into an optimization formulation in which the optimum design parameter vector, $\underset{\sim}{P}^*$, is to be determined such that

$$E(\underset{\sim}{P}^*; \underset{\sim}{g}) = \min_{\underset{\sim}{P}} E(\underset{\sim}{P}; \underset{\sim}{g}) \tag{1}$$

subject to the constraint

$$f(\underset{\sim}{P}; \underset{\sim}{g}) = 0 \tag{2}$$

with the flow variable vector $\underset{\sim}{g}$ satisfying the flow governing equation

$$\underset{\sim}{D}(\underset{\sim}{g}) = 0 \tag{3}$$

subject to the boundary condition

$$\underset{\sim}{B}(\underset{\sim}{g}; \underset{\sim}{P}) = 0 \tag{4}$$

Our objective is to maximize the propeller efficiency, η. The objective function is therefore defined by

$$E = -\eta$$

The propeller power requirements are constrained to a specified value through the constraint function

$$f = C_p - C_{po}$$

Equation (3) is the system of Euler equations governing the flow field, and Equation (4) is the propeller solid wall boundary condition. The vector of design parameters, $\underset{\sim}{P}$, defines the propeller geometrical configuration.

The goal of the optimization scheme is to determine the values of the design parameters that minimize the objective function, E, subject to an equality constraint. A search must therefore be conducted in the design parameter space $\underset{\sim}{P}$ for the optimum solution, $\underset{\sim}{P}^*$. This optimization problem is most conveniently solved in the rotated design parameter space $\hat{\underset{\sim}{P}}$, with the $\hat{P}_1$ coordinate normal to the constraint surface and the $\hat{P}_\ell$ coordinates, where $\ell = 2, 3, \ldots, L$, parallel to the constraint surface. For fixed values of the components of $\hat{\underset{\sim}{P}}$, let

$$\underset{\sim}{g}^{n+1} = \underset{\sim}{\psi}(\underset{\sim}{g}^n; \hat{\underset{\sim}{P}}), \quad n = 0, 1, 2, \ldots \tag{5}$$

be the iterative solution for the analysis problem, where ψ denotes the solution obtained by applying the iterative scheme for solving the Euler equations once using $\tilde{g}^n$ as an initial guess. An implicit approximate factorization scheme is used here to solve the Euler equations. It is described in Reference 6. As for the analysis solution, obtaining the optimization solution requires the repeated application of Equation (5) to update the flow field. While $\hat{\tilde{P}}$ is held fixed in the former case, it is allowed to vary in the latter. The scheme used to update $\hat{\tilde{P}}$ follows.

The vector of design parameters $\hat{\tilde{P}}$ is updated every ΔN iterations. Therefore,

$$\hat{\tilde{P}}^{n+1} = \hat{\tilde{P}}^{n} + \delta\hat{\tilde{P}}^{n+1} \tag{6}$$

where

$$\delta\hat{\tilde{P}}^{n+1} = 0, \ (n+1)/\Delta N \neq 1, 2, 3, \ldots$$

In the iterative steps that satisfy the relation $(n+1)/\Delta N = 1, 2, 3, \ldots$, the incremental values for the design parameters are given by

$$\delta\hat{P}_1^{n+1} = -\frac{f^n}{|f^n|}\left[\min\left(C\,|f^n|, \delta P_{max}\right)\right] \tag{7}$$

$$\delta\hat{P}_\ell^{n+1} = \min\left(1, \frac{\delta P_{max}}{|\Delta\hat{P}_\ell^{n+1}|}\right)\Delta\hat{P}_\ell^{n+1}, \ \ell = 2, 3, \ldots, L \tag{8}$$

where

$$f^n = f(\hat{\tilde{P}}^n; \tilde{g}^n)$$

$$\Delta\hat{P}_\ell^{n+1} = \frac{1}{2}\left[c_1(\tau_\ell^{n+1} + 1) + c_2(\tau_\ell^{n+1} - 1)\right]\delta\hat{P}_\ell^{n+1-\Delta N} \tag{9}$$

$$\tau_\ell^{n+1} = -\frac{\Delta E_\ell^n\,\delta\hat{P}_\ell^{n+1-\Delta N}}{|\Delta E_\ell^n\,\delta\hat{P}_\ell^{n+1-\Delta N}|}$$

$$\Delta E_\ell^n = E(\hat{\tilde{P}}^n + \varepsilon\,\tilde{i}_\ell^n; \tilde{g}_\ell^n) - E(\hat{\tilde{P}}^n; \tilde{g}^n)$$

Here, ε is a small positive constant and $\tilde{i}_\ell^n$, $\ell = 1, 2, \ldots, L$, are the set of orthogonal unit vectors along the axes of the rotated coordinate system $\hat{P}_1^n, \hat{P}_2^n, \ldots, \hat{P}_L^n$. The solution $\tilde{g}_\ell^n$ is a solution in which the ℓ^{th} component of $\hat{\tilde{P}}$ is perturbed by ε.

The incremental displacement in the design parameter space, introduced so that the constraint may be satisfied, is taken in the direction normal to the constraint surface and is determined by the chord method in Equation (7). The constant δP_{max} sets an upper limit on the magnitude of this incremental displacement. The incremental displacements given by Equation (8) are introduced along the coordinate axes, which are parallel to the constraint surface with the purpose of reducing the value of the objective function. The sign of the incremental correction $\delta\hat{P}_{\ell}^{n+1}$, where $\delta\hat{P}_{\ell}^{n+1}$ is the ℓ^{th} component of the vector $\delta\hat{\underset{\sim}{P}}^{n+1}$, is chosen to be opposite to that of $\partial E/\partial\hat{P}_{\ell}^{n}$. The magnitude of the increment $\delta\hat{P}_{\ell}^{n+1}$ is given by

$$\left|\delta\hat{P}_{\ell}^{n+1}\right| = c\left|\delta\hat{P}_{\ell}^{n+1-\Delta N}\right|$$

with an upper limit given by δP_{max}, where $c > 0$. If the signs of $\delta\hat{P}_{\ell}^{n+1}$ and $\delta\hat{P}_{\ell}^{n+1-\Delta N}$ are in agreement, then the last two iterative solutions $\hat{P}_{\ell}^{n}$ and $\hat{P}_{\ell}^{n-\Delta N}$ fall to one side of the point along the $\hat{P}_{\ell}$ direction at which E is a minimum. In this case, c is set equal to the constant c_1, which is greater than 1. Increasing the magnitude of the step size in this manner accelerates the approach toward the point along the $\hat{P}_{\ell}$ direction at which E is a minimum. On the other hand, if the signs of $\delta\hat{P}_{\ell}^{n+1}$ and $\delta\hat{P}_{\ell}^{n+1-\Delta N}$ are not in agreement, then $\hat{P}_{\ell}^{n}$ and $\hat{P}_{\ell}^{n-\Delta N}$ fall on opposite sides of the point along the $\hat{P}_{\ell}$ direction at which E is a minimum. In this case, c is set equal to the constant c_2, which is less than 1. Decreasing the magnitude of the step size in this manner is necessary for convergence to the point along the $\hat{P}_{\ell}$ direction at which E is a minimum.

The updated components of the design parameter vector $\hat{\underset{\sim}{P}}^{n+1}$ are used to calculate the new flow iterative solution, $\underset{\sim}{g}^{n+1}$, given by

$$\underset{\sim}{g}^{n+1} = \underset{\sim}{\psi}(\underset{\sim}{g}^{n};\ \hat{\underset{\sim}{P}}^{n+1}) \tag{10}$$

and the perturbed solutions $\underset{\sim}{g}_{\ell}^{n+1}$, $\ell = 1, 2, \ldots, L$, given by

$$\underset{\sim}{g}_{\ell}^{n+1} = \underset{\sim}{\psi}(\underset{\sim}{g}_{\ell}^{n};\ \hat{\underset{\sim}{P}}^{n+1} + \varepsilon\ \hat{\underset{\sim}{i}}_{\ell}^{n+1}) \tag{11}$$

While the optimization procedure is most suitably conducted in terms of the transformed parameters $\hat{P}_{\ell}$, $\ell = 1, 2, \ldots, L$, the flow solution is computed in terms of the physical design parameters P_{ℓ}, $\ell = 1, 2, \ldots, L$.

In order to express the transformed design parameters in Equations (10) and (11) in terms of the original design parameters, it is necessary to use the transformation equation, which relates these two sets of parameters. This equation is

$$\underset{\sim}{P}^{n+1} = T^{n+1}\, \hat{\underset{\sim}{P}}^{n+1}$$

where the orthogonal transformation matrix T^{n+1} is given by

$$T^{n+1} = [\underset{\sim}{i}_1^{n+1}\ \underset{\sim}{i}_2^{n+1}\ \ldots\ \underset{\sim}{i}_L^{n+1}]$$

The unit vector $\underset{\sim}{i}_1^{n+1}$ is normal to the constraint surface at $\hat{\underset{\sim}{P}} = \hat{\underset{\sim}{P}}^n$ and is given by

$$\underset{\sim}{i}_1^{n+1} = \frac{\nabla f(\hat{\underset{\sim}{P}}^n;\ \underset{\sim}{g}^n)}{|\nabla f(\hat{\underset{\sim}{P}}^n;\ \underset{\sim}{g}^n)|} \tag{12}$$

where an estimate for $\hat{G}_\ell^n$, the ℓ^{th} component of ∇f, is given by

$$\hat{G}_\ell^n = \frac{f(\hat{\underset{\sim}{P}}^n + \varepsilon\ \hat{\underset{\sim}{i}}_\ell^n;\ \underset{\sim}{g}_\ell^n) - f(\hat{\underset{\sim}{P}}^n;\ \underset{\sim}{g}^n)}{\varepsilon} \tag{13}$$

The Gram-Schmidt orthogonalization process, which uses a set of L linearly independent vectors to construct a set of L orthonormal vectors, is used to construct the unit vectors $\underset{\sim}{i}_\ell^{n+1}$, $\ell = 2, 3, \ldots, L$, along the rotated axes $\hat{P}_\ell^{n+1}$, $\ell = 2, 3, \ldots, L$. The following equation is used for this purpose:

$$\underset{\sim}{i}_\ell^{n+1} = \frac{\underset{\sim}{I}_\ell^{n+1}}{|\underset{\sim}{I}_\ell^{n+1}|}, \quad \ell = 2, 3, \ldots, L$$

where

$$\underset{\sim}{I}_\ell^{n+1} = \underset{\sim}{i}_\ell^n - \sum_{k=1}^{\ell-1} (\underset{\sim}{i}_\ell^n \cdot \underset{\sim}{i}_k^{n+1})\, \underset{\sim}{i}_k^{n+1}$$

In the initial iterative step, the vectors $\underset{\sim}{i}_\ell$ are given by $\underset{\sim}{i}_\ell^1 = \underset{\sim}{e}_\ell$, $\ell = 1, 2, \ldots, L$, where $\underset{\sim}{e}_\ell$, $\ell = 1, 2, \ldots, L$, are the set of orthogonal unit vectors along the axes of the coordinate system $P_1, P_2, \ldots, P_L$.

While the flow variable vector $\underset{\sim}{g}$ is updated each iterative step, the coordinate system in the design parameter space is rotated every ΔN iterations. The unit vectors $\underset{\sim}{i}_\ell$, like the vector of design parameters $\underset{\sim}{P}$, are updated only in the iterative steps that satisfy the relation $(n+1)/\Delta N = 1, 2, 3, \ldots$

The optimization scheme described above requires that L+1 iterative problems be solved in parallel. In addition to the main solution, L perturbed solutions are computed in which each of the design parameters in the transformed space $\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_L$ is perturbed. The computational costs and the computer memory requirements are therefore proportional to L+1. A modification to this scheme, which requires that only L iterative solutions be obtained, is now introduced. In the modified procedure, the perturbation solution associated with the perturbed design parameter in the direction of the $\hat{P}_1$ axis, normal to the constraint surface, is not computed. This solution was used in Equation (13) to compute $\hat{G}_1^n$, which is required for the calculation of the vector $\underset{\sim}{i}_1^{n+1}$, which determines the direction normal to the constraint surface in Equation (12). In the absence of this solution, a new procedure for rotating the design parameter space must be defined. The procedure is first explained for the case of a two-design-parameter problem, and then it is extended to the general multi-design-parameter problem.

Figure 1 shows the design parameter space for a two-design-parameter problem. In the figure, the constraint function values f_o^n, f_1^n, f_2^n are defined as follows:

$$f_o^n = f(\hat{\underset{\sim}{p}}^n, \underset{\sim}{g}^n)$$

$$f_1^n = f(\hat{\underset{\sim}{p}}^n + \varepsilon\, \hat{\underset{\sim}{i}}_1^n, \underset{\sim}{g}_1^n)$$

$$f_2^n = f(\hat{\underset{\sim}{p}}^n + \varepsilon\, \hat{\underset{\sim}{i}}_2^n, \underset{\sim}{g}_2^n)$$

In the modified procedure, the chord method, used in Equation (7) to satisfy the constraint condition, is used to rotate the design parameter space. The rotation angle $\delta\theta_M^{n+1}$ given by

$$\delta\theta_M^{n+1} = \tan^{-1}\left[\frac{C(f_2^n - f_o^n)}{\varepsilon}\right] \tag{14}$$

is used to rotate the coordinate system, where the subscript M indicates that the modified scheme is used. The angle $\delta\theta_M^{n+1}$ is now compared to the corresponding rotation angle $\delta\theta^{n+1}$ used in the original scheme and given by

$$\delta\theta^{n+1} = \tan^{-1}\left[\frac{f_2^n - f_o^n}{\varepsilon}\;\frac{\varepsilon}{f_1^n - f_o^n}\right] \tag{15}$$

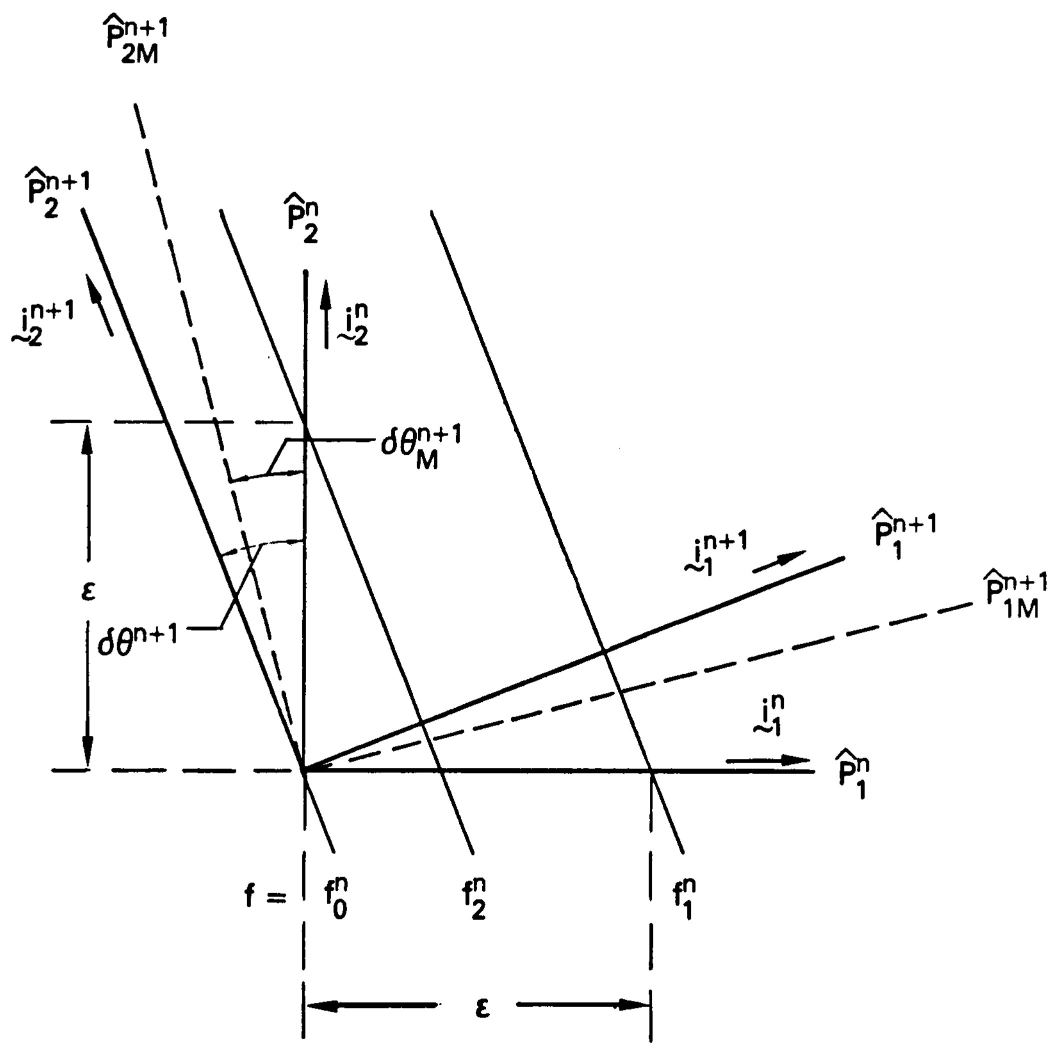

Figure 1. Two-Dimensional Design Parameter Space

This comparison shows that the term $f_1^n - f_o^n$ in the original scheme is replaced by ε/C in the modified scheme. Therefore, the modified scheme may be viewed as the original scheme with the exception that the exact value for $\underset{\sim}{i}_1$ is replaced by an approximate estimate in which the gradient of f in the direction of the $\hat{\underset{\sim}{P}}_1$ axis, $\hat{G}_1$, is not calculated but is estimated using the same proportionality constant used in the chord method of Equation (7). Thus,

$$\hat{G}_1 = \frac{1}{C} \tag{16}$$

This is applicable for both the two-design-parameter problem and the general multi-design-parameter problem.

In the optimization scheme developed here, corrective increments are applied to the design parameter solutions every few iterations of updating the flow solutions. For convergence to occur, the signs of the increments must be chosen correctly to allow the iterative solution to approach the desired solution. The magnitudes of the increments are dependent on the computational constants c_1, c_2, and C. Because the design parameters are updated frequently during the iterative process, we are not concerned with determining the incremental step sizes that lead to the highest short-term convergence rate. In fact, this may be difficult to define, since the flow variable solutions are continuously changing during the iterative process. Our aim is to achieve design parameter convergence over a long term defined by the number of iterations required for the flow solution convergence. A wide range of incremental step sizes should produce the desired convergence properties over many iterations, even though convergence properties over a few iterations may differ. These comments apply to both of the schemes described above for determining the design parameter space rotation. The direct procedure for determining the design parameter space rotation in the original scheme is replaced by an iterative procedure in the modified scheme. Since this rotation is updated frequently during the iterative process, this replacement should have no substantial effect on the overall convergence of the solution.

A potential problem exists when the modified scheme is used for rotating the design parameter axes. This problem is now discussed and suggestions for overcoming it are then presented.

In the first $\Delta N-1$ iterative steps of solving the problem, the coordinate system in the design parameter space coincides with the original unrotated design parameter space $P_1, P_2, \ldots, P_L$. At the ΔN^{th} iterative step, a new rotated coordinate system is determined. When Equation (13) for determining $\hat{G}_1^{\Delta N-1}$ is used, we are guaranteed that the vector $\underset{\sim}{i}_1^{\Delta N}$ points in the direction in which the constraint function increases. Consequently, the use of Equation (7) will cause the iterative solution to approach the constraint surface. When Equation (13) is replaced by Equation (16) for determining $\hat{G}_1^{\Delta N-1}$, there is a possibility that the computed vector $\underset{\sim}{i}_1^{\Delta N}$ points in the direction in which the constraint function decreases. In this case, the assumption that C is positive is wrong, and using it will cause the solution to diverge. This occurs if the vector $\underset{\sim}{e}_1$ is nearly in the direction of $-\nabla f^{\Delta N-1}$. That is, if the quantity

$$- \frac{\nabla f^{\Delta N-1}}{|\nabla f^{\Delta N-1}|} \cdot \underset{\sim}{e}_1$$

is close to unity. The probability of this occurring is approximately 1:4 in a two-design-parameter problem and is reduced further as the number of design parameters increases. There are two suggested approaches for overcoming this problem. In the first approach, the initial few iterations are performed using the original scheme for determining $\hat{G}_1^n$ by Equation (13) in order to determine the correct initial directions for the $\hat{P}_1$ axis. This may then be updated using the modified scheme, Equation (16), in the rest of the computation. Realizing that the probability for the potential problem to occur is small, the second approach uses the modified scheme from the beginning of the computation. If divergence does occur, then the constraint function is redefined to be equal to the negative of the original constraint function and the problem is solved again.

OPTIMIZATION PROGRAM

The optimization scheme described above is applicable to general aerodynamic problems, and can be used in conjunction with different analysis codes. The scheme was tested by applying it to the problem of optimizing advanced propeller designs.[7] In these tests it was used in conjunction with code NASPROP-E which is an analysis code that computes the flow around an advanced propeller. A description of the scheme used in solving the flow equations around the propeller is given in Reference 6.

In the following sections we will focus on describing the elements of the optimization program. It will be necessary to make some reference to code NASPROP-E; however, this reference will be minimized. Names of subroutines or COMMON blocks belonging to NASPROP-E will be bracketed to indicate that they do not belong to the optimization program and that they are replaceable by other subroutines or COMMON blocks when the optimization scheme is used in conjunction with other analysis codes. A manual describing the main elements of code NASPROP-E is given in Reference 8.

Flow Chart

In Figure 2, a brief flow chart describing the overall flow of the program is presented.

Program Input

The input required by the optimization program is described in this section. A dictionary of the input variables is provided, followed by a description of the input data format and an example of a set of input data.

Dictionary of Input Variables

The variables which are input to the optimization program are described in the following list.

BARM1(L) ,L=1,NPARM

Minimum allowable value for the Lth design parameter, in rotated space, during the iterative process.

BARM2(L) ,L=1,NPARM

Maximum allowable value for the Lth design parameter, in rotated space, during the iterative process.

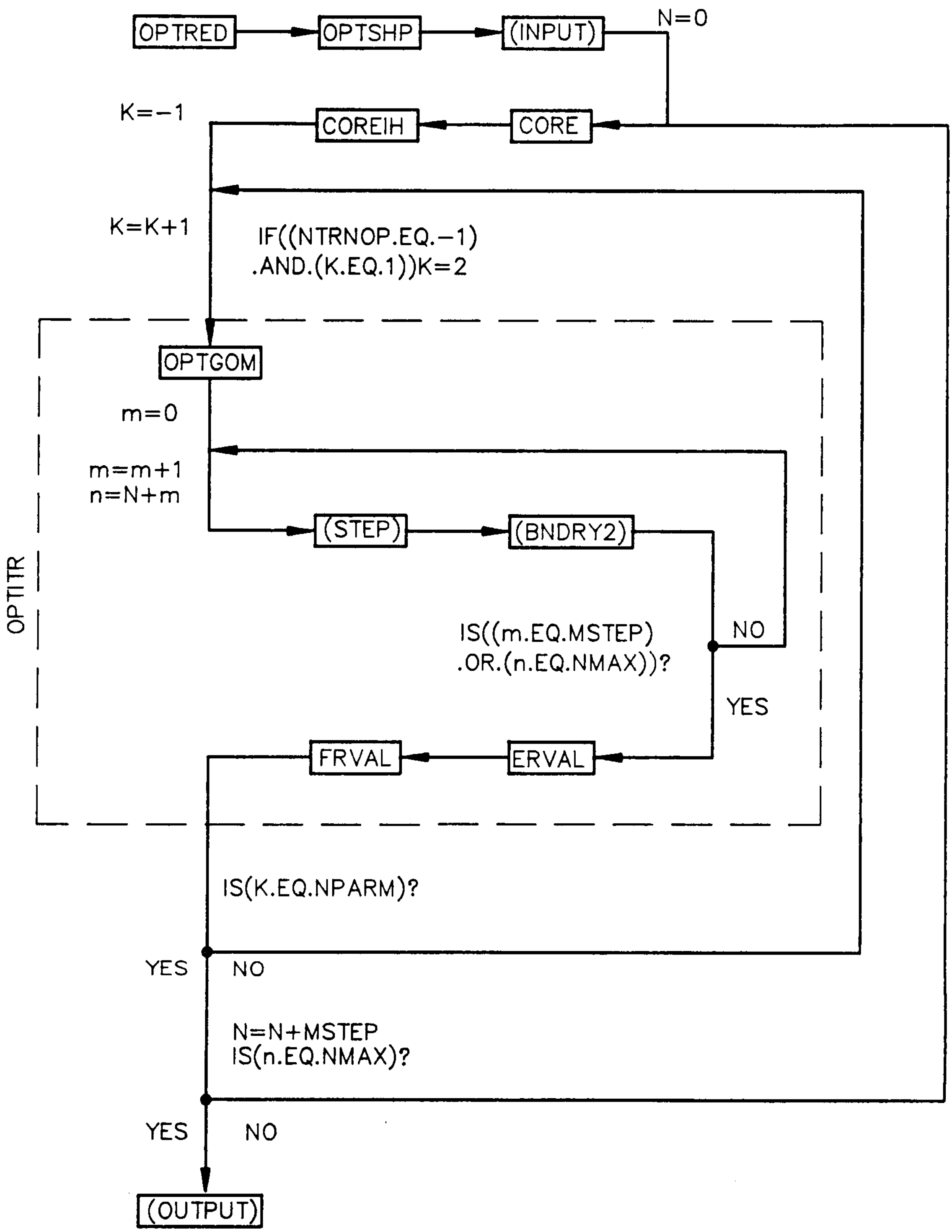

Figure 2. Program Flow Chart

C1	Incrementing factor for the optimization scheme.
C2	Decrementing factor for the optimization scheme.
CAA	Proportionality constant for the chord method.
CNSTR	Value of constraint function.
DELPRM	Small positive incremental value used to perturb the design parameters.
DPARM(L)	,L=2,NPARM Initial incremental step size for the Lth design parameter.
DPARMX(L)	,L=1,NPARM Maximum allowable incremental step size for the Lth design parameter in rotated space.
DSIN(M)	,M=1,NDSIN Initial values for the geometrical parameters. These geometrical parameters are coefficients of the shape functions defined in subroutine OPTSHP.
ISLCT(L)	,L=1,NPARM The Lth component of the design parameter vector is equal to the ISLCT(L) component of the vector of geometrical parameters.
IRSC	This is a parameter which identifies the type of restart data.
=0	The restart data was saved from a regular analysis run.
=1	The restart data was saved from an optimization run.
ISVC	This is a parameter that identifies the type of data being saved.
=0	The present run is a regular analysis run.
=1	The present run is an optimization run.
MSTEP	Number of iterative steps at which the design parameters are periodically updated.
NDSIN	Number of geometrical parameters used to perturb the aerodynamic shape from the baseline configuration. A number, NPARM, of the geometrical parameters are allowed to vary. These are the design parameters. The rest of the geometrical parameters are fixed. NDSIN must have a value of 1 or greater.
NHBL12 =1	The hub shape is optimized
=2	The blade shape is optimized
NPARM	Number of design parameters
=0	Regular analysis problem
=1	Constraint problem
>1	Constrained optimization problem

NPRNT Data associated with the iterative history of the solution is printed every NPRNT iterative steps.

NTRNOP =0 Coordinates will not be rotated. NPARM problems are solved in parallel. This option should be used only if the constraint surface is a planar surface and the first design parameter coordinate is normal to that surface and increases in the direction of increasing constraint function. In general, this option is not useful.

=-1 Coordinates will be rotated. NPARM problems are solved in parallel. $\hat{G}_1$ is determined by Equation (16).

=1 Coordiates will be rotated. NPARM + 1 problems are solved in parallel. $\hat{G}_1$ is determined by Equation (13).

RRRANG Parameter used in the definition of the shape functions given in subroutine OPTSHP.

RRRC Parameter used in the definition of the shape functions given in subroutine OPTSHP.

Input Data Format

The input data for the optimization program is read from subroutine OPTRED. The following reading sequence is used with the format given below.

```
      READ (55,100)
      READ (55,200) NPARM, MSTEP, NTRNOP, NDSIN, NPRNT, IRSC, ISVC
      READ (55,100)
      READ (55,200) (ISLCT (I), I=1, MMPARM)
      READ (55,100)
      READ (55,300) (DSIN (I), I=1, NDSIN)
      READ (55,100)
      DO 1 N=1, MMPARM
      READ (55,300) DPARM(N), DPARMX(N), BARM1(N), BARM2(N)
    1 CONTINUE
      READ (55,100)
      READ (55,300) CNSTR
      READ (55,100)
      READ (55,300) CAA, C1,C2, DELPRM
      READ (55,100)
      READ (55,300) RRRC, RRRANG
```

```
      READ (55,100)
      READ (55,200) NHBL12
  100 FORMAT (1X)
  200 FORMAT (8I10)
  300 FORMAT (8F10.0)
```

Note: MMPARM = 1 if NPARM = 0 and MMPARM = NPARM otherwise. When NPARM = 0, the variables ISLCT(1), DPARM(1), DPARMX(1), BARM1(1), and BARM2(1) must be specified in the input data, even though they are not used in this particular case. The variable DPARM(1) is always not used; however, a value must be specified for it in the input data.

An additional set of data is read from the subroutines of the analysis code used in conjunction with the optimization program. This data includes the baseline description of the aerodynamic configuration, flow conditions (for example, the free stream Mach number value), computational parameters associated with the analysis code, and the desired number of iterations, NMAX. For the propeller problem, this set of data is decribed in Reference 8.

Input Data Example

The following is an example of a set of input data:

NPARM	MSTEP	NTRNOP	NDSIN	NPRNT	IRSC	ISVC
2	40	-1	3	20	0	0
ISLCT(1)	ISLCT(2)					
1	3					
DSIN(1)	DSIN(2)	DSIN(3)				
2.0	0.0	2.0				
DPARM(I)	DPARMX(I)	BARM1(I)	BARM2(I)			
0.0	1.0	-50.0	50.0			
0.5	1.0	-20.0	20.0			
CNSTR						
1.7						
CAA	C1	C2	DELPRM			
3.0	1.2	0.6	0.0001			
RRRC	RRRANG					
0.25	0.5					
NHBL12						
2						

COMMON Blocks

The COMMON blocks used in the optimization program are given below and the variables appearing in these blocks are defined. This is followed by a COMMON block/subroutine cross reference.

Dictionary of COMMON Block Variables

COMMON BLOCK PARM1

COMMON/PARM1/NP, NB, MM, M1, M2, RADI

The variables belonging to this COMMON block are variables used in code NASPROP-E.

COMMON BLOCK PARM4

COMMON/PARM4/NPRNT

NPRNT	Output and plotting data is written out every NRPNT iterative steps.

COMMON BLOCK PARM5

COMMON/PARM5/DBETO(I,2), DBTINC(I,L), DSIN(L), NDSIN, ISLCT(L), RNO(II), DRNINC(II,L)

All dimensions, I, appearing in this COMMON block should have a value of NB or greater, where NB is the number of planes in the radial direction defining the blade geometry. All dimensions, II, appearing in this COMMON block should have a value of NN or greater, where NN is the number of points in the axial direction defining the nacelle geometry. All dimensions, L, appearing in the COMMON block should have a value of NDSIN or greater.

DBETO(J,2)	Input or baseline blade twist at each radial station relative to the 75% span station.
DBTINC(J,K)	Perturbation blade twist at each radial station, corresponding to the Kth geometrical shape function.
DRNINC(J,K)	Perturbation nacelle radius at each axial station, corresponding to the Kth geometrical shape function.
DSIN(J)	Jth geometrical parameter.
NDSIN	Number of geometrical parameters.
ISLCT(J)	The Jth component of the design parameter vector is equal to the ISLCT(J) component of the vector of geometrical parameters.
RNO(J)	Input or baseline nacelle radius at each axial station.

COMMON BLOCK PARM6

COMMON/PARM6/RRR(I), RRRC, RRRANG

The dimension I in this COMMON block should have a value of NB or greater, where NB is the number of planes in the radial direction defining the blade geometry.

RRR(J)	Radius of radial stations on the blade.
RRRANG	Parameter used in the definition of the shape functions.
RRRC	Parameter used in the definition of the shape functions.

COMMON BLOCK PARM7

COMMON/PARM7/IRSC, ISVC, NNCNT

IRSC	Parameter defining the type of restart data.
ISVC	Parameter defining the type of data being saved.
NNCNT	Counter

COMMON BLOCK PARM40

COMMON/PARM40/NHBL12

NHBL12	=1	The hub shape is optimized
	=2	The blade shape is optimized

COMMON BLOCK PAR171

COMMON/PAR171/PARM(I), DPARM(I), EP(I), FP(I), BARM1(I), BARM2(I), DPARMX(I),
XNRM(I), TRAN(I,I), TRANT(I,I), TRANO(I,I), NPARM, MSTEP,
CAA1, CAA2, CAA, DELPRM, E0, F0, PARMOC(I), NTRNOP

All dimensions, I, appearing in this COMMON block should have a value of NPARM or greater.

BARM1(L)	,L=1, NPARM Minimum allowable value for the Lth design parameter, in rotated space, during the iterative process.
BARM2(L)	,L=1, NPARM Maximum allowable value for the Lth design parameter, in rotated space, during the iterative process.
CAA	Proportionality constant for the chord method.
CAA1	Incrementing factor for the optimization scheme.
CAA2	Decrementing factor for the optimization scheme.
DELPRM	Small positive incremental value used to perturb the design parameters.

DPARM(L)	,L=1, NPARM Incremental step size for the Lth design parameter in rotated space.
DPARMX(L)	,L=1, NPARM Maximum allowable incremental step size for the Lth design parameter in rotated space.
EO	Objective function corresponding to main solution.
EP(L)	,L=1, NPARM Objective function corresponding to the Lth perturbed design parameter in rotated space.
FO	Constraint function corresponding to the Lth perturbed design parameter in rotated space.
FP(L)	,L=1, NPARM Constraint function corresponding to the Lth perturbed design parameter in rotated space.
MSTEP	Number of iterative steps at which the design parameters are periodically updated.
NPARM	Number of design parameters.
NTRNOP	Parameter defining scheme options
PARM(L)	,L=1, NPARM Lth design parameter in rotated space.
PARMOC(L)	,L=1, NPARM Lth design parameter in original unrotated space.
TRAN	The first NPARM x NPARM elements of this two-dimensional array form the transformation matrix which operates on a vector with components defined relative to the rotated coordinate system of the present iteration to give the corresponding components relative to the rotated coordinate system of the previous iteration.
TRANT	The first NPARM x NPARM elements of this two-dimensional array form the transpose of the transformation matrix defined by the elements of TRAN.
TRANO	The first NPARM x NPARM elements of this two-dimensional array form the transformation matrix which operates on a vector with components defined relative to the present rotated coordinate system to give the corresponding components relative to the original unrotated coordinate systef.

COMMON BLOCK PAR172

COMMON/PAR172/CNSTR

CNSTR	Constraint function.

COMMON BLOCK PAR176

COMMON/PAR176/LLMIN, LLMAX

LLMIN	Number of initial iterative step.
LLMAX	Number of final iterative step.

COMMON BLOCKS (BASE), (COUNT), (CRIDC)

These COMMON blocks are among the NASPROP-E COMMON blocks.

COMMON BLOCK PWEF

COMMON/PWEF/CPI, ETA, II

CP1	Power
ETA	Efficiency
II	Counter giving the number of times output data has been written.

COMMON BLOCK TRAC

COMMON/TRAC/TIME(I), RESMX(I), RESAV(I), POW(I), EFF(I), DSNPR(I,J)

All dimensions, I, appearing in this COMMON block should have a value of KK or greater, where

KK = NMAX / NPRNT,

NMAX is the desired number of iterations, and the output data is written out once every NPRNT iterations. The dimension, J, appearing in the COMMON block should have a value of MMPARM or greater where MMPARM = max(1,NPARM).

TIME(K) ,K=1, KK

Number of iterative steps for the Kth data point.

RESMX(K) ,K=1, KK

Maximum residual at Kth data point.

RESAV(K) ,K=1, KK

Residual Euclidean norm at Kth data point.

POW(K) ,K=1, KK

Power coefficient at the Kth data point.

EFF(K) ,K=1, KK

Efficiency at the Kth data point.

DSNPR(K,L) ,K=1, KK ,L=1, NPARM

Iterative solution for the Lth design parameter at the Kth data point.

COMMON BLOCK VAR1

This COMMON block is used to save memory storage requirements. Variables contained in this COMMON block in subroutine (MAIN4) are associated with mesh generation. These variables are replaced with flow field solutions in all other subroutines which contain the COMMON block VAR1. In subroutine (MAIN4) the COMMON block appears as

COMMON/VAR1/GRIDX, GRIDY, GRIDZ, GRIDR, GRIDP

where GRIDX, GRIDY, GRIDZ, GRIDR, and GRIDP are arrays that contain mesh information. In all other subroutines containing COMMON block VAR1, it appears as

COMMON/VAR1/Q, S, QM1, SCRAT

where Q, S, and QM1 are arrays that contain flow field solutions, while the array SCRAT is not used.

COMMON Block/Subroutine Cross Reference

	main	CORE	COREIH	ERVAL	FRVAL	OPTGOM	OPTITR	OPTRED	OPTSHP	PARMS	SWTCH1	SWTCH2
PARM 1	X								X			
PARM4								X				
PARM5	X							X	X	X		
PARM6	X							X	X			
PARM7	X							X				
PARM40								X	X			
PAR171	X	X	X	X			X	X				
PAR172					X			X				
PAR176	X						X					
(BASE)						X						
(COUNT)							X					
(CRIDC)							X					
PWEF	X			X	X							
TRAC	X											
VAR1	X					X					X	X

Program Subroutines

Subroutine Cross Reference

For each subroutine, the following table lists the subroutines that it calls and the subroutines that call it.

Subroutine	Called By	Calls
Main Program		CORE, CORE1H, OPTITR, OPTRED, PARMS, SWTCH1, SWTCH2
CORE	Main Program	GRAM, XNORM
CORE1H	Main Program	TRNSFR
ERVAL	OPTITR	(FORCE)
FRVAL	OPTITR	(FORCE)
GRAM	CORE	
OPTGOM	OPTITR	(GRID), (INPUT), (METRIC), (PRMESH)
OPTITR	Main Program	ERVAL, FRVAL, OPTGOM, PARMS, SWTCH1, SWTCH2, TRNSFR, (BNDRY2), (STEP)
OPTRED	Main Program	
OPTSHP	MREAD	
PARMS	Main Program, OPTITR	
SWTCH1	Main Program, OPTITR	
SWTCH2	Main Program, OPTITR	
TRNSFR	CORE1H, OPTITR	
XNORM	CORE	
(BNDRY2)	OPTITR	
(FORCE)	ERVAL, FRVAL	
(GRID)	OPTGOM	
(INPUT)	OPTGOM	
(METRIC)	OPTGOM	
(PRMESH)	OPTGOM	
(STEP)	OPTITR	

Subroutine Descriptions

In this section, a brief description of each of the subroutines of the optimization program is given. Also described briefly are the subroutines of Program NASPROP-E that were referenced in this report.

SUBROUTINE CORE

This subroutine computes the transformation matrix TRAN and its transpose TRANT. These two matrices relate the components of a vector relative to the current rotated coordinate system to its components relative to the rotated coordinate system of the previous iteration.

SUBROUTINE COREIH

This subroutine computes the new iterative solution for the design parameters. All the dimensions of the variables appearing in the DIMENSION statement in this subroutine should be given a value equal to or greater than NPARM, the number of design parameters.

SUBROUTINE ERVAL

This subroutine computes the value of the objective function. The user is required to add neccessary COMMON blocks and necessary calls to other subroutines which allow the computation of the objective function.

Argument List

E (Output) Objective function

SUBROUTINE FRVAL

This subroutine computes the value of the constraint function. The user is required to add necessary COMMON blocks and necessary calls to other subroutines which allow the computation of the constraint function.

Argument List

F (Output) Constraint function

SUBROUTINE GRAM

This subroutine uses the Gram-Schmidt orthogonalization process to construct a set of orthonormal vectors from a set of linearly independent vectors.

Argument List

E (Input) Two-dimensional array. Set the first dimension equal to the first dimension of TRANT in COMMON block PAR171. Set the second dimension equal to 1. The first N columns of array E contain in their N first rows the components of the input set of linearly independent vectors.

ET (Output) Two-dimensional array. Set the first dimension equal to the first dimension of TRANT in COMMON block PAR171. Set the second dimension equal to 1. The first N columns of array ET contain in their N first rows the components of the output set of orthonormal vectors.

N (Input) Number of vectors.

NBIG (Input) The component number with the biggest magnitude in the first column of the input array E.

SUBROUTINE OPTGOM

This subroutine is called so that the computational mesh may be generated. Subroutine OPTGOM calls subroutine (PRMESH), which does the actual computation of the mesh. The storage area allocated for COMMON block VAR1 is used for mesh computations. Array QM1, appearing as a variable in COMMON block VAR1, stores the flow solution. For that reason, array QM1 is written to tape 60 before subroutine (PRMESH) is called. Following the call to subroutine (PRMESH), the flow solution is restored in array QM1 by reading it from tape 60.

Argument List

LLKK (Input)

=1 the mesh is computed

=2 mesh dependent variables are computed

SUBROUTINE OPTITR

This subroutine updates the main flow solution or the perturbed flow solution by performing MSTEP iterative steps for fixed values of the design parameters. The dimensions of the variables appearing in the DIMENSION statement in the subroutine should be given a value equal to or greater than NPARM, the number of design parameters.

Argument List

N	(Input)	
	=0	Main solution is updated
	≠0	Perturbed solution is updated
NCYCB	(Input)	Number of cycles for updating the design parameters.

SUBROUTINE OPTRED

This subroutine reads the input data associated with the optimization scheme. The data is read from tape 55.

SUBROUTINE OPTSHP

This subroutine defines the shape function used to perturb the baseline aerodynamic configuration. The shape functions are defined by the two-dimensional array DBTINC in the case of optimizing the blade angle distribution and by the two dimensional array DRNINC in the case of optimizing the hub shape. The first argument of these arrays defines a radial position along the blade while the second argument defines a particular shape function.

SUBROUTINE PARMS

This subroutine sets the design parameters, which are contained in the elements of the array PARM, equal to the geometrical parameters, contained in the elements of the array DSIN, if NPL1R2 = 1. If NPL1R2 = 2, the subroutine sets the geometrical parameters equal to the design parameters.

Argument List

NPL1R2		(Input)
	=1	Elements of PARM are set equal to elements of DSIN
	=2	Elements of DSIN are set equal to elements of PARM
PARM		(Input or Output)
		Vector of design parameters
NPARM		(Input)
		Number of design parameters

SUBROUTINE SWTCH1

The optimization scheme requires that a number of solutions for different flow problems be obtained in parallel. The argument N refers to the main solution if N=0, and to the Nth perturbed solution if N≠0. The iterative solutions for these different problems are stored in the array QM1. Prior to updating a particular solution, it is temporarily stored in array Q. The code then updates array Q, replacing the old solution by the solution at the new iterative step. This solution is again stored in array QM1. Subroutine SWTCH1 stores the solution corresponding to the Nth problem in array Q prior to updating it.

SUBROUTINE SWTCH2

This subroutine stores the solution corresponding to the Nth problem in array QM1 after it is updated.

SUBROUTINE TRNSFR

A transformation matrix relating two coordinate systems, (1) and (2), is used in this subroutine to compute the vector components relative to coordinate system (2) from the vector components relative to coordinate system (1).

Argument List

VIN	(Input) Vector whose components are defined relative to coordinate system 1.
VOUT	(Output) Vector whose components are defined relative to coordinate system 2.
T	(Input) Transformation matrix. The first dimension in this array should be given a value equal to the first dimension of the variable TRAN in COMMON block PAR171.
N	(Input) Dimension of the space under consideration.

SUBROUTINE XNORM

This subroutine computes the components of the unit vector normal to a constraint surface contour.

Argument List

- DP (Input) Small positive number used as a perturbing parameter.
- FPNP (Input) Vector whose elements contain the values of the constraint function at positions perturbed from the point of interest by DP along each of the coordinate directions.
- FONP (Input) Value of the constraint function at the point of interest.
- X (Output) Unit vector normal to the constraint surface.
- N (Input) Number of design parameters.
- CAA (Input) Proportionality constant used in the chord method.
- NTRNOP (Input) Parameter determining the procedure for computing the unit normal vector.

SUBROUTINE (BNDRY2)

This subroutine satisfies the flow boundary conditions.

SUBROUTINE (FORCE)

This subroutine computes the power and efficiency.

SUBROUTINE (GRID)

This subroutine reads in the computational mesh from TAPE11.

SUBROUTINE (INPUT)

This subroutine reads the input data for the aerodynamic analysis code.

SUBROUTINE (METRIC)

Quantities which are dependent on the computational mesh are computed in this subroutine.

SUBROUTINE (PRMESH)

This is the main subroutine for the mesh generation program. Its function is to call other subroutines which compute the mesh. The mesh data is written on TAPE11.

SUBROUTINE (STEP)

This is the subroutine which computes the flow solution at a new iterative step.

Program Output

An iterative history showing the iterative time step, the residual, the power, the efficiency, and the design parameters is printed out. Similar data is saved on TAPE40 for the purpose of producing line plots. This data is written in Subroutine (OUTPUT).

REFERENCES

1. Hicks, R. M., Murman, E. M., and Vanderplaats, G. N., "An Assessment of Airfoil Design by Numerical Optimization," NASA TM X-3092, July 1974.

2. Haney, H. P., Johnson, R. R., and Hicks, R. M., "Computational Optimization and Wind Tunnel Test of Transonic Wing Designs," Journal of Aircraft, Vol. 17, July 1980, pp. 457-463.

3. Hicks, R. M., "Transonic Wing Design Using Potential-Flow Codes - Successes and Failures," SAE Paper 810565, April 1981.

4. Cosentino, G. B., and Holst, T. L., "Numerical Optimization Design of Advanced Transonic Wing Configurations," AIAA Paper 85-0424, January 1985.

5. Davis, W., "TRO-2D: A Code for Rational Transonic Aero Optimization," AIAA Paper 85-0425, January 1985.

6. Yamamoto, O., Barton, J. M., and Bober, L. J., "Improved Euler Analysis of Advanced Turboprop Propeller Flows," AIAA Paper 86-1521, June 1986.

7. Rizk, M. H., "Optimizing Advanced Propeller Designs by Simultaneously Updating Flow Variables and Design Parameters," AIAA Paper 88-2532, June 1988.

8. Chaussee, D. S., and Kutler, P., "User's Manual for Three-Dimensional Analysis of Propeller Flow Fields," NASA CR-167959, January 1983.

6141R

APPENDIX
LIST OF SYMBOLS

c_1 = incrementing factor for optimization scheme [see Equation (9)]

c_2 = decrementing factor for optimization scheme [see Equation (9)]

C = positive constant for chord method [see Equation (7)]

C_p = power coefficient

C_{po} = desired power coefficient

$\underset{\sim}{e}_\ell$ = unit vector along the P_ℓ axis

E = objective function

f = constraint function

$\underset{\sim}{g}$ = solution of the flow governing equations

$\hat{G}_\ell$ = ℓ^{th} component of ∇f relative to rotated coordinate system

$\underset{\sim}{i}_\ell$ = unit vector along the $\hat{P}_\ell$ axis with components defined relative to the unrotated design parameter coordinate system

$\hat{\underset{\sim}{i}}_\ell$ = unit vector along the $\hat{P}_\ell$ axis with components defined relative to the rotated design parameter coordinate system

L = number of design parameters

$\underset{\sim}{P}$ = vector of design parameters

$\hat{\underset{\sim}{P}}$ = vector of design parameters relative to rotated coordinate system

P_ℓ = ℓ^{th} component of design parameter vector

$\hat{P}_\ell$ = ℓ^{th} component of design parameter vector relative to rotated coordinate system

$\delta\hat{\underset{\sim}{P}}$ = incremental vector used to update the vector of design parameters

δP_{max} = maximum incremental value allowed in updating the design parameters

ΔN = number of iterative steps at which $\underset{\sim}{P}$ is periodically updated

ε = small positive incremental value used to perturb the design parameters

η = efficiency

$\underset{\sim}{\psi}$ = flow iterative solution

Superscripts

n = iteration number

$*$ = optimum value

$\hat{}$ = rotated coordinate system

Subscripts

M = coordinate system rotated by the modified scheme

NASA

National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-182180		

4. Title and Subtitle	5. Report Date
USERS MANUAL FOR AN AERODYNAMIC OPTIMIZATION SCHEME THAT UPDATES FLOW VARIABLES AND DESIGN PARAMETERS SIMULTANEOUSLY	July 1988
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Magdi H. Rizk	Flow Report No. 444
	10. Work Unit No.
9. Performing Organization Name and Address	11. Contract or Grant No.
Flow Research, Inc. 21414 - 68th Avenue South Kent, WA 98032	NAS3-24855
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report Users Manual
National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This user's manual is presented for an aerodynamic optimization program that updates flow variables and design parameters simultaneously. The program was developed for solving constrained optimization problems in which the objective function and the constraint function are dependent on the solution of the nonlinear flow equations. The program was tested by applying it to the problem of optimizing propeller designs. Some reference to this particular application is therefore made in the manual. However, the optimization scheme is suitable for application to general aerodynamic design problems. A description of the approach used in the optimization scheme is first presented, followed by a description of the use of the program.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Aerodynamic design, Efficiency, Optimization, Propeller, Transonic Turboprop	Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No of pages	22. Price*
Unclassified	Unclassified	30	

NASA FORM 1626 OCT 86

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