NASA Contractor Report 182164

Structural Tailoring of Advanced Turboprops (STAT)

Programmer's Manual

K. Brown
UNITED TECHNOLOGIES CORPORATION
Pratt & Whitney
400 Main Street
East Hartford, CT 06108

March 1989

(NASA-CR-182164) STRUCTURAL TAILORING OF ADVANCED TURBOPROPS (STAT) PROGRAMMER'S MANUAL (Pratt and Whitney Aircraft) 53 p CSCL 21E
N89-20132
Unclas
G3/07 0198626

Prepared for
Lewis Research Center
Under Contract NAS3-23941

National Aeronautics and
Space Administration

PREFACE

This manual describes the programming aspects of the Structural Tailoring of Advanced Turboprops (STAT) system. The manual is divided into seven main sections, each of which describe module intent and interaction within the STAT program. Section 1 discusses the overall program flow and categorization of the modules. The ADSREAD subsystem, which provides a means for the analysis modules to access a description of the current blade design, is described in Section 2. The function of each high-level module is given in Section 3. The file units and common blocks are described in Sections 4 and 5, respectively. Section 6 describes the method in which refined analysis modules can be added to the program, and Section 7 defines the job control language for the CRAY COS version of STAT.

STRUCTURAL TAILORING OF ADVANCED TURBOPROPS (STAT)
PROGRAMMER'S MANUAL

Table of Contents

Section Page

1.0 INTRODUCTION

The Structural Tailoring of Advanced Turboprops (STAT) computer program was developed to perform numerical optimizations on highly swept propfan blades. The optimization procedure seeks to minimize an objective function, defined as either direct operating cost or aeroelastic differences between a blade and its scaled model, by tuning internal and external geometry variables that must satisfy realistic blade design constraints.

The STAT analyses include an aerodynamic efficiency evaluation, a finite element stress and vibration analysis, an acoustic analysis, a flutter analysis, and a once-per-revolution (one-p) forced response life prediction capability. The STAT constraints include blade stresses, blade resonances, flutter, tip displacements and a one-p forced response life fraction. The STAT variables include all blade internal and external geometry parameters needed to define a composite material blade. The STAT objective function is dependent upon a blade baseline definition which the user supplies to describe a current blade design for cost optimization or for the tailoring of an aeroelastic scale model.

To perform a blade optimization, three component analysis categories are required: an optimization algorithm; approximate analysis procedures for objective function and constraint evaluation; and refined analysis procedures for optimum design validation. The STAT computer program contains an executive control module, an optimizer, and all necessary approximate and refined analyses. The optimization algorithm of STAT is the Automated Design Synthesis (ADS) optimization package, which is a proven tool for optimizations with a small to medium (1 to 30) number of design variables. A flowchart of the STAT procedure is shown in Figure 1.

The approximate analyses of STAT utilize an efficient, coarse mesh, plate finite element blade vibration analysis procedure. The finite element analysis provides blade natural frequencies and mode shapes, stress under centrifugal and pressure loads, and blade weight. Additional constraint evaluations, including flutter, power, acoustic and one-p calculations, utilize output from the finite element analysis.

After each completed design iteration, the current design is verified by applying refined analyses to assure that all constraints are satisfied. If the constraints are not all satisfied, then correction factors are applied to the approximate analyses to better calibrate them with the refined analyses results. The optimization process continues to the next completed design move until a local optimum design has been found whose constraints satisfy refined analyses.

To use the blade optimization system, design variable (parametric) curves used to describe the external and internal geometry of a turboprop fan blade must be defined. External geometry curves define blade thickness, section stacking, camber, chord, twist and conical sections. Internal geometry curves define individual layer thickness, percent chord coverage and position over the entire blade.

STAT PROCEDURAL OUTLINE

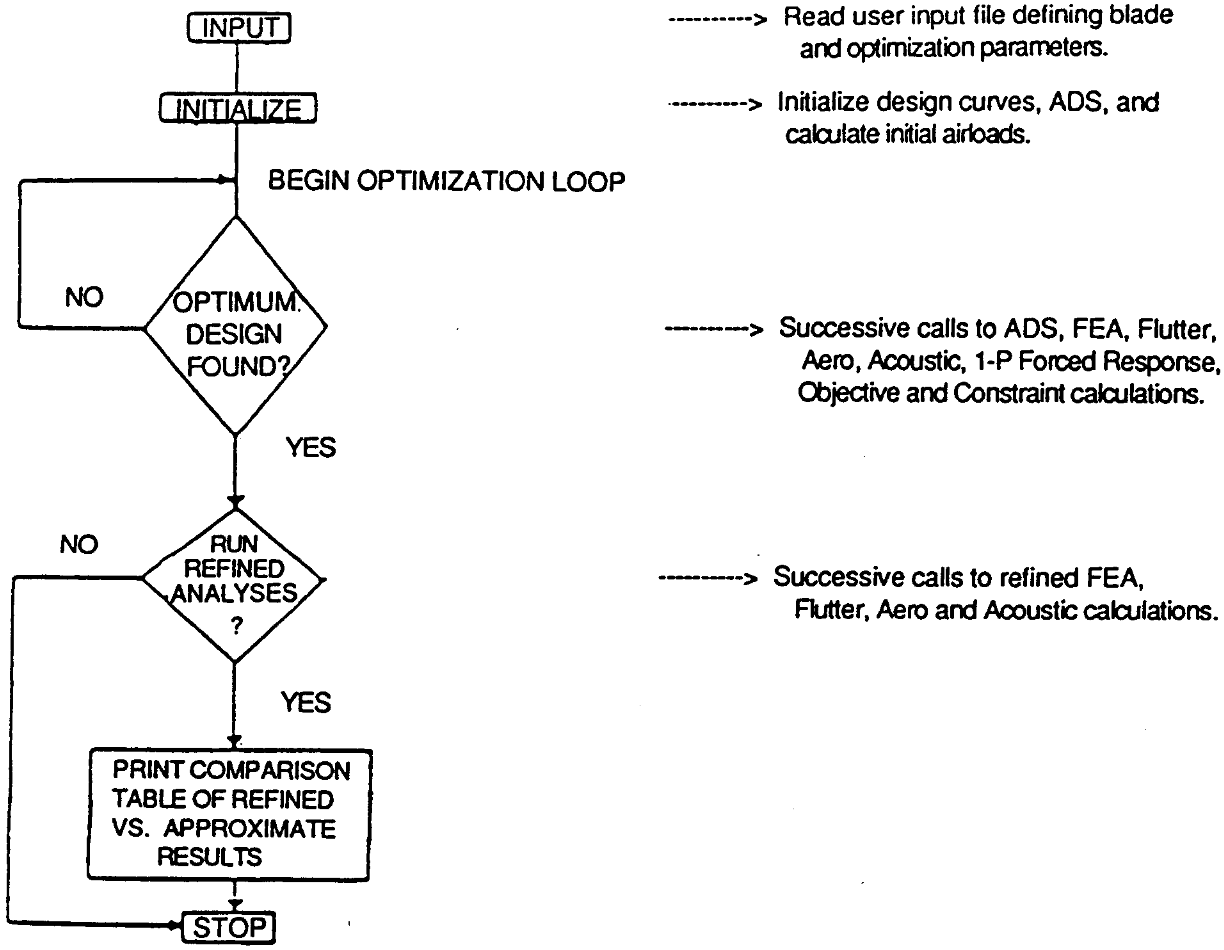
INPUT
Read user input file defining blade and optimization parameters.
INITIALIZE
Initialize design curves, ADS, and calculate initial airloads.
BEGIN OPTIMIZATION LOOP
NO
OPTIMUM DESIGN FOUND?
Successive calls to ADS, FEA, Flutter, Aero, Acoustic, 1-P Forced Response, Objective and Constraint calculations.
YES
NO
RUN REFINED ANALYSES ?
Successive calls to refined FEA, Flutter, Aero and Acoustic calculations.
YES
PRINT COMPARISON TABLE OF REFINED VS. APPROXIMATE RESULTS
STOP

Figure 1 Structural Tailoring of Advanced Turboprops Overall Program Flow

The STAT system has been applied to the Large-Scale Advanced Propfan (LAP) SR-7 blade, the LAP SR-7 aeroelastic scale model blade and the 18E SR-7 infeasible blade design, as detailed in Reference 1. The STAT program made significant improvements in all three cases and demonstrated the great potential for design enhancements through the application of numerical optimization to turboprop fan blades of composite construction.

This manual describes the functionality of the STAT system from a programmer's viewpoint. It provides a top-down description of module intent and interaction. The purpose of this manual is to familiarize the programmer with the STAT system so that he/she may enhance or verify the program's function.

1.1 PROGRAM DESCRIPTION

The STAT program was originally written in FORTRAN 77 and compiled with the IBM VSFORT 1.4.1 compiler. It has since been modified to be compatible with the CRAY CFT 1.14BF4 compiler.

STAT provides for the numerical optimization of turboprop blades. The program's goal is to iteratively improve the design of a blade by modification of the design variables of the analysis. The initial blade, called the 'baseline', is described by the user's data set. The output of the program is a set of design variable values which describe improvements to the baseline blade. User input to the program may be categorized as follows:

1. Data which describes the baseline design
2. Data which describes the design variables and constraints
3. Data which controls the method of the optimization
4. Data which controls the output of information to the user.

1.2 CATEGORIZATION OF MODULES

The STAT program consists of four logically different types of modules. The first of these is the analysis module. The analysis modules provide the system with a description of the behavior of the blade that is currently defined by the design variables. The aerodynamic analysis, the finite element analysis, and the flutter analysis are examples of analysis modules.

The second type of module is the objective function. The objective function is the user's numerical description of goodness of a design, stated so that the best design produces the miminum value of the objective function.

Thirdly, STAT contains modules which comprise the optimizer. The optimizer uses the current design variable definition of the blade, the value of constraints on blade behavior and the value of the objective function to suggest a modification to the current blade (again stated in terms of the design variables). STAT utilizes the ADS optimizer, developed by G. N. Vanderplaats (ref. 2).

The final class of modules provides communication between the design vector description of the blade and the analysis modules. These routines are collectively referred to as the ADSREAD system. ADSREAD provides for the mnemonic description of the blade given by the user and a means for analysis routines to retrieve a meaningful description of the current design. Understanding the operation of the ADSREAD subsystem is necessary for understanding the STAT program's function.

2.0 THE ADSREAD SUBSYSTEM

The ADSREAD subsystem provides a means for the analysis modules to access a description of the current blade design. This description is by means of cubic spline curves which the user has defined through CURVE cards. Many of these CURVE cards are required; the curve name is reserved and understood by the geometry and model pre-processors.

ADSREAD is also used to define variables and constraints on the model to be optimized. This ability is provided through VARIABLE, CUTOFF and CONSTRNT cards. These cards are explained in the STAT User's Manual (ref. 3).

There are three services provided by ADSREAD. First, ADSREAD allows the user to describe the baseline model. Second, ADSREAD allows the user to constrain and vary the model. Finally, ADSREAD provides a means to interpret the ADS design variables in terms of the design variables of the model.

All aspects of the model design, which equate to design variables and define the dynamic description of the blade, are contained in the array COEF. Primarily, this array contains cubic spline coefficients of the current design variable curves. The COEF array is acted upon in the following ways.

Baseline Description

Before entering the optimization loop, the program calls STATIN to read the baseline description. The data read describes the design by means of abscissa and ordinate curve definitions which parameterize the design. The curve description is then translated into cubic spline coefficients in the routine OPT003. Both of these routines are visible from the main routine.

Current Curve Update

After each iteration of the optimization loop, ADS returns a new design vector. The design vector must then be interpreted in terms of the curve(s) it modifies. The routine OPT009, visible from the main routine, is responsible for the translation. Note that at the beginning of execution, the design vector is zero. This is because the current design is represented by a perturbation of the baseline design. That is, the current design is equal to the baseline design curve plus the delta addition to that curve provided by the ADS design vector.

Design Curve Value Retrieval

The analysis modules need a description of the blade to be analyzed. Some of the information about this model is found in the data dedicated to that analysis (such as ASETs and SPCs of the FEA). Most of the data which describes the current geometry of the model is retrieved from the COEF splines using the routine OPT001. OPT001 simply takes as input the name of the curve on which values are required and the abscissa values along that curve where the values are required. The output is ordinate values of that curve.

3.0 PROGRAM FLOW CHARTS

This section depicts the flow of the STAT program, and describes the function of each high-level module.

3.1 MAIN PROGRAM

FLOW	ROUTINES USED
START	
Initialization	
Read user input	STATIN
Define objective function variables	ASSOCS
Initialize design curves	OPT003 OPT009
Initialize ADS	ADS
Initial AERO analysis	EFFICH
Begin Approximate Optimization Loop This loop terminates when ADS has found an optimum design (and returns INFO = 0), or when the maximum number of iterations (MXITER) has been exceeded, or when the maximum number of design steps (IDM) has been exceeded.	
Call optimizer for next design vector	ADS
Update design curves	OPT009
Produce shape description of airfoil	BDS
Finite Element Analysis (FEA)	FEARUN
If FEA yields a singular stiffness matrix during a warm-start: Then re-run FEA in cold-start mode.	
If FEA yields a singular stiffness matrix during a cold start: Then STOP	

```
    Flutter analysis                                              FLUTER
    Deflected geometry update                                     GAEROH
    Aero analysis                                                 EFFICH
    Acoustic analysis                                             DBNF
    1-P aerodynamic analysis                                      ONEP
    1-P forced response                                           ONEPFR
    Objective function                                            OBJF
    Constraint calculations                                       DEFCON
    Output results                                                REPORT
End Approximate Optimization Loop
Update design curves with final design variables and plot         OPT009
final design variables
Output final results of approximate optimization                  REPORT
Determine if sufficient data has been input to perform a          REFINT
refined analysis
If data is insufficient:
Then STOP
Begin Refined Analysis
Produce refined shape description of optimum airfoil              BDS
Refined finite element analysis                                   FEARUN
IF FEA yields a singular matrix:
Then STOP
Refined flutter analysis                                          FLUTRR
(This is not available in the NASA public version)
Deflected geometry update                                         GAEROH
Refined aero analysis                                             EFFICH
Refined acoustic analysis                                         NOISE
(This is not available in the NASA public version)
Output table comparing results of approximate and                 REFREP
refined analyses.
STOP
```

3.2 STATIN

STATIN reads all program input except finite element data (FEA data is read separately due to the program's storage strategy). Optimization, Objective function, Aero Analysis and Mesh generation data are read here.

FLOW	ROUTINES USED
START	
Initialization	
Process optimization input contained on the following cards: OPTIMIZE VARIABLE CONSTRNT CURVE ABSCISSA DEBUG CONSTANT DEPEND PLOT CUTOFF MATERIAL LAY-UP PRIORITY *END_OPT $ (comments) Performs initial validity check.	ADSRD
Process efficiency analysis input contained on the following cards: BLADE ENVIRON AIRFOIL AXIALV FILR/R *END_EFF Performs initial validity check.	RDEFF
Process objective function input contained on the following cards: OBJTYPE BASELINE SENSE BLDDATA BLDFREQ BLDMASS BLDDEFL *END_OPJ Performs initial validity check.	RDOBJ
Process geometry generation input contained on the following cards: GEOMGEN CHORDTAB SPANTAB ATTACHMT *END_GEN Performs initial validity check.	RDGGEN
RETURN	

3.3 ASSOCS

ASSOCS associates locations of the input arrays BASE and SENSE with particular variables used by the objective function. By using this method rather than a direct read into those variables, it is easier for the programmer to change objective functions. The programmer assigns different meanings to the fields of the BASE and SENSE cards and writes a new ASSOCS routine.

FLOW	ROUTINES USED
START	
Associate user input data in arrays BASE and SENSE with specified variables used by the objective function.	
RETURN	

3.4 OPT003

OPT003 requires the abscissa and ordinate values from the input curve data, read by STATIN. It outputs the spline coefficients of the baseline design curves.

```
                                                           ROUTINES
   FLOW                                                      USED

   START

   Initialization

   Begin Loop 1
   For i = 1...number of curves do

      Verify that an ABSCISSA card has been input which      ENDIT
      matches the abscissa of curve(i).

      If no match exists
      Then

         print error message

         STOP

      Endif

      Save baseline coefficients of curve(i)                 OPT002
      in array COEF.

      Plot curve(i)                                          PLOTEM

   End Loop 1

   Begin Loop 2
   For i = 1...number of cutoffs do

      Enter value of cutoff(i) into array COEF.              OPT004

      If no cutoff is explicitly specified

      Then set the value to the appropriate abscissa
      endpoint.

   End Loop 2

   RETURN
```

3.5 OPT009

OPT009 transforms the baseline curves in array COEF into current curves based on the ADS design vector, X.

```
                                                              ROUTINES
   FLOW                                                         USED

   START

   Initialization                                             ZERO

   Begin Loop 1
   For i = 1...number of curves do

   If no variables or constants reference curve(i)
   Then goto End Loop 1.

      Begin Loop 2
      For j = 1...number of design variables do

      If design variable(j) does not reference curve(i)
      Then goto End Loop 2.

      Create delta design variable curve. X vector consists of
      abscissa values contained in VARABS and DEPLOC arrays.
      Y vector consists of delta design variables contained
      in ADS design variable array.

      End Loop 2

      Begin Loop 3
      For j = 1...number of constants do

      If constant(j) does not reference curve(i)
      Then goto End Loop 3.

      Create constant curve. X vector consists of abscissa
      values obtained from CONSTANT cards. Y vector consists
      of ordinate values from CONSTANT cards.

      End Loop 3

   Merge the delta design variable curve and the constant curve  VMERGE
   to form a single delta design variable (DDV) curve.

   Fit a cubic spline through the DDV curve and determine the    BMFIT
   coefficients for an interpolating polynomial.
```

```
|  Evaluate the DDV curve at the points where the baseline        BEVALE
|  curve(i) is defined.
|
|  Add the delta curve values to the baseline curve(i) and
|  store as the current curve(i).
|
|  Re-spline the current curve(i) and place the coefficients       OPT007
|  in array COEF.
|
|  If requested, plot current curve(i).                             PLOTEM
|
└──End Loop 1

Update COEF array with any cutoff design variable.                  OPT008

Update COEF array with any material angle design variable.          SCLRVR

RETURN
```

3.6 ADS

ADS is a general purpose numerical optimization program containing a wide variety of algorithms. Its purpose is to minimize an objective function subject to various user-defined constraints.

```
FLOW                                                        ROUTINES
                                                              USED

START

If first time in routine
Then

    Check for valid input combinations

    Initialize scalar parameters

    Define work array storage

    RETURN

Endif

Get scalars from work array

Set idv = idv + 1

If idv le number of design variables
Then

    Calculate finite difference gradient of design
    variable(idv)

    Evaluate objective function

    Begin Loop 1
    For j = 1...number of constraints

        Evaluate constraint(j)

    End Loop 1

    Output values of gradients, objective, and constraints

Else

    Set idv = 0

    Perform one-dimensional search
```

```
Evaluate objective function

Begin Loop 2
For j = 1...number of constraints

    Evaluate constraint(j)

End Loop 2

Output values of gradients, objective, and constraints

Check convergence criteria to determine if optimization
is complete

If optimization is complete
Then output final results

Endif

RETURN
```

3.7 EFFICH

EFFICH performs an aerodynamic analysis on the airfoil. It outputs propfan performance characteristics and blade loads.

FLOW	ROUTINES USED
START	
Initialization	
Retrieve t/b, chord, and cone angle design curves from work array (HOB, BOD, and CONE)	OPT001
If first time in routine Then	
Retrieve design curves representing 'cold' geometry blade (BETA, CLD, XOR, YOR, and ZOR)	OPT001
Set 'cold' blade angle	
Else	
Retrieve design curves representing 'hot' geometry blade (BETAH, CLDH, XORH, YORH, and ZORH)	OPT001
Calculate 'hot' blade angle	
Endif	
Calculate activity factors	AFFIDC
Calculate aerodynamic blade loads	LINTHT
PNPINT	
Calculate propfan performance parameters (thrust, shaft h.p., efficiency,...)	MNPANP
Store propfan data in common for later use	AIRFLT
RETURN	

3.8 BDS

BDS, the Blade Design System, interpolates design curve data and outputs a set of radial splines which define the airfoil external surface, leading edge radius centers, stacking points, and trailing edge radius centers.

FLOW	ROUTINES USED
START	
Initialization	
Interpolate blade design curves from sparse definition (design variables) to dense definition. Blade design curves are HOB, CLD, BOD, BETA, CONE, XOR, YOR, and ZOR.	OPT001
Calculate current blade angle	ANGLE
Calculate external airfoil conic sections, stacking points, leading edge radius and trailing edge radius centers.	BDS004
RETURN	

3.9 FEARUN

FEARUN controls execution of the finite element input reader, mesh generator, and finite element analysis.

FLOW	ROUTINES USED
START	
If entering routine for the first time Then	
Initialize counters and storage arrays.	INTINT
Allocate storage in work array for input items.	PREMAP
Read user input finite element bulk data.	INPUTT
Else Then retrieve previous deflection vectors and bulk data from scratch file.	GETMAT INPUTT
Endif	
Create blade finite element model, including material definition.	FEAGEO
If requested, print bulk data listing.	GEODIG
Perform finite element analysis.	FEA
If FEA yields a singular stiffness matrix Then RETURN	
Recover stresses of laminate materials and calculate Tsai-Wu failure criteria	PSTCMP
Store current deflection vector and bulk data on scratch file.	PUTMAT
RETURN	

3.10 FLUTER

FLUTER determines the classical flutter stability of propellers with swept blades in a high subsonic flow environment. It takes beam equivalent mode shapes that have been derived from the plate finite element analysis, and calculates the damping ratio and natural frequencies for six Mach number cases for the first four natural modes. A stall flutter parameter and the flutter Mach number are output from this routine.

```
                                                                  ROUTINES
    FLOW                                                            USED

    START

    Initialization

    Define non-dimensional radial sections for analysis

    Define the four solution modes

    Calculate the stall flutter parameter

    Define the six Mach number cases

10  Set npass = npass + 1

    Set air density = air density * 0.7

    Begin Loop 1
    For i = 1...number of Mach number cases

        Calculate the blade rotation needed to align the 3/4 chord
        with the 3/4 relative velocity(i)

        Calculate radius and relative velocity vectors

        Define aerodynamic sweep

        Calculate mode shapes at the previously defined          FQQ1
        analysis sections

        Calculate chord, sweep, and twist at the analysis        FQQ1
        sections

        Calculate gap chord ratio                                FQQ1

        Calculate radial span lengths
        Assemble aerodynamic modal matrix using calculated            FDD5
        section properties. Solve matrix to obtain modal frequencies
        and damping.

        Save frequencies and damping for number of modes for
        Mach number(i)

        If any mode of Mach number case(i) has damping gt 1.0
        Then goto 10   (divergence)

        If npass ge 20
        Then goto 20

    End Loop 1

    If any mode of any Mach number case has damping lt 0.0
    Then goto 10   (instability)

20  Calculate flutter Mach number, correcting for cascade
    effect

    RETURN
```

3.11 GAEROH

GAEROH calculates 'hot' propeller geometry.

FLOW	ROUTINES USED
START	
Initialization	
Calculate change in stacking due to blade deflections from f.e.a.	QSPLIN
Calculate change in radial tilt, tangential tilt, axial tilt and twist due to blade deflections from f.e.a.	
Calculate change in camber due to blade deflections from f.e.a.	OPT001
Generate and store spline coefficients for 'hot' geometry curves. These curves are BETAH (twist), CLDH (camber), XORH, YORH, AND ZORH (tilt).	OPT002
RETURN	

3.12 DBNF

DBNF calculates blade noise and passage frequency.

FLOW	ROUTINES USED
START	
Initialization	
Calculate blade passage frequency as a function of tip speed and blade diameter.	
Calculate corrected tip speed, tpcor	UNBAR
Calculate corrected tip clearance, tccor	UNBAR
Calculate corrected activity factor, afcor	UNBAR
Calculate corrected sweep, swcor	UNBAR
Calculate corrected number of blades, nbcor	UNBAR
Calculate total blade noise as a function of altitude, temperature, tpcor, tccor, afcor, swcor, and nbcor.	
RETURN	

3.13 ONEPFR

ONEPFR calculates maximum static and vibratory TSAI-WU stresses, and outputs the maximum stress failure ratio.

FLOW	ROUTINES USED
START	
Initialization	ZERO
Retrieve current f.e. deflection vectors and bulk data from scratch file.	GETMAT
Calculate aerodynamic loads	ALOAD
If specified, output aerodynamic loads	PFORCE
Calculate modal loads, generalized stiffness, and amplification factors.	
Calculate a steady stress Tsai-Wu value for each layer of every airfoil element.	GETMAT PSTCMP
Calculate an equivalent Tsai-Wu stress by summing the Tsai-Wu stresses for each mode multiplied by their respective amplification factors. The result will be the vibratory Tsai-Wu stress to be measured on a 'Tsai-Wu Goodman Diagram'.	GETMAT PSTCMP
Determine maximum static/vibratory Tsai-Wu stress ratio from 'Tsai-Wu Goodman Diagram'.	
RETURN	

3.14 OBJF

OBJF uses data generated by the analysis modules, such as stress, noise, efficiency, etc..., to calculate an objective function value. The value of the objective function is used by ADS to evaluate design goodness.

```
                                                              ROUTINES
FLOW                                                            USED

START

Initialization                                                ZERO

If objective function type eq 1
Then

Calculate scaled blade efficiency value, ddocl

Calculate scaled blade weight value, ddoc2

Calculate scaled blade noise value, ddoc3                     LNEAR1

Calculate blade acquisition cost, ddoc4                       LNEAR1

Calculate blade maintenance cost, ddoc5

Calculate objective function (delta direct operating
cost) by adding ddocl, ddoc2, ddoc3, ddoc4, and ddoc5.

Endif

If objective function type eq 2
Then

Calculate frequency correlation, ddocl

Calculate blade mass distribution, ddoc2

Calculate modal deflections at tip, ddoc3

Calculate static blade untwist, ddoc4

Calculate objective function (aero-differences between
full and scale models) by adding ddocl, ddoc2, ddoc3,
and ddoc4.

Endif

RETURN
```

3.15 DEFCON

DEFCON maps program analysis outputs into the appropriate positions of the TERMS array. NOTE: The positioning of TERMS values is application dependent and defines the values of ITERM on CONSTRNT cards.

FLOW	ROUTINES USED
START	
Initialization	
Frequencies for first 10 modes (From FEA) are stored in TERMS(1..10).	
Tsai-Wu layer stresses are stored in TERMS(11..30).	
Flutter Mach number is stored in TERMS(31).	
Stall flutter parameter is stored in TERMS(32).	
Power is stored in TERMS(33).	
Activity factor is stored in TERMS(34).	
Max. von Mises stress is stored in TERMS(35).	
Tip uncamber is stored in TERMS(36).	
Tip untwist is stored in TERMS(37).	
Tip leading edge axial displacement is stored in TERMS(38).	
Tip trailing edge axial displacement is stored in TERMS(39).	
One-p force response maximum stress failure ratio is stored in TERMS(40).	
RETURN	

3.16 REPORT

REPORT outputs program results.

```
                                                          ROUTINES
FLOW                                                        USED

START

Determine elapsed CPU time.                               JOBTIM

If optimization is not complete (INFO ne 0)
Then

Output results from current function call.

If current function call is not a design move
Then RETURN

Store design move results for summary print.

Output summary print every nine design moves.

RETURN

Endif

Output final optimization results.

RETURN
```

4.0 FILE UNITS

The STAT program uses auxiliary files for data storage in various ways. Particular FORTRAN unit numbers are used for certain program functions. Table I lists the file unit numbers. Any units not listed are available for use as scratch files by the user. All file units listed are sequential access.

Table I File Units

FORTRAN Unit Number	Description	Format
5	STAT input	Card image
6	STAT output	Line printer
7	Internal scratch file	Formatted
8	Internal scratch file	Formatted
10	Internal scratch file	Unformatted
14	Internal scratch file	Formatted
17	Internal scratch file	Formatted
18	Internal scratch file	Unformatted
19	STAT diagnostic output	Line printer
20	Internal scratch file	Formatted
21	Internal scratch file	Unformatted
22	Internal scratch file	Unformatted
24	Internal scratch file	Formatted
26	Internal scratch file	Formatted
30	Mesh generator diagnostic output	Formatted
39	Finite element bulk data	Formatted
40	STAT summary report	Line printer
41	Finite element scratch file	Unformatted
42	Finite element scratch file	Unformatted
43	Finite element scratch file	Unformatted
44	Finite element scratch file	Unformatted
45	Finite element scratch file	Unformatted
46	Finite element scratch file	Unformatted
47	Finite element modeshapes	Unformatted
48	Finite element stresses	Unformatted

5.0 STAT COMMON BLOCKS

```
COMMON /ABCONV/ ICONVG(10)

COMMON /ABDATA/ CLAFAB(100,10),CLGSAB(100,10),PHIAB(100,10),
. ALPHAB(100,10),BETAAB(100,10),ITNO(10)

COMMON /ABITER/ PHIINT,CLA1

COMMON /ADSTMP/ X1DM

COMMON /AERDAT/ THETA(15,6,2), AA(15,6,2), DIAG(15,6,2),
. CONST(15,6,2), CMACH(15,6,2), SMACH(15,6,2),
. CFDP(15,6,2),  SKEW(15,6,2)

COMMON /AFCPIT/ AFXX,CLIXX

COMMON /AIRALZ/ ALZA01(308),ALZA03(268),ALZA07(15 )

COMMON /AIRCDF/ CDFA01(50 ),CDFA03(11 ),CDFA08(17 )

COMMON /AIRCDM/ CDMA01(19 ),CDMA03(29 ),CDMA07(25 ),CDMA08(25)

COMMON /AIRDAT/ FAIR(1000)

COMMON /ALFEXT/ XCL2(10),ALPFS(10),CLFS(10)

COMMON /ALLDAT/ TITLE1(18)    ,TITLE2(18) ,TITLE3(7)  ,DATA(2260)

COMMON /APEFFC/ EFFX

COMMON /ARGOT2/ TEMP2, POP2

COMMON /ARGOUT/ ALTO, TSS, SHPDD, AFF, DIAMM, XNBB, SWEPTT

COMMON /BARSAV/ ICTBAR,IDBAR(10),ICPBAR(10),ICDBAR(10),
. IELBAR(10),IPDBAR(10),IG1BAR(10),IG2BAR(10),
. T1BAR(10),T2BAR(10),T3BAR(10),IPABAR(10),
. IMDBAR(10)

COMMON /BARX/ EE(500),GG(500),RIY(25),RIZ(25),RJ(25),YA1(25),
. ZA1(25),YA2(25),ZA2(25),AREA(25),IBAR(25),NBAR,
. RK1(25),RK2(25)

COMMON /BDSC01/ A0(20),A1(20),A2(20),C(20),C1(20),C2(20),C3(20),
. A5(5),A6(5),A7(5),C4(6)

COMMON /BDSC02/ TERAD,GOCODE
COMMON /BDSC03/ T10(48),T16(150),T63(48),T64(150),T65(48),
. T00(48), PX(22), PAN(55) , T66(48) , TCA(96) , T230(96) , XZ(18),
. NACA62 , NACA63, NACA64  , NACA65

COMMON /BIGMAT/ PHI(400),RW(6000),ZW(6000)

COMMON /CALSP/ EPS(50)

COMMON /CASDAT/ SIGMAX(15,2),THETAG(15,2),THETAB(15,6,2),
. TAUB(15,2)

COMMON /CCOM/ CHORD(15,2),ALCRAD(15,2),ALCPHI(15,2),ALCAXL(15,2)

COMMON /CDPER/ DFR,DPR

COMMON /CINCOM/ CINPUT(15,2),ALCIRD(15,2),ALCIPH(15,2),
. ALCIAX(15,2)

COMMON /CLCDDT/
. CLSAV(15,6,2),CDSAV(15,6,2),ALPHA(15,6,2),PHIHSD(15,6,2),
. CD0(15,6,2),CIRC(15,6,2),SAVCIR(15,6,2),FTRAN(15,6,2,3)

COMMON /CMOMNT/ CMC4

COMMON /CMPNUM/ NCOMP

COMMON /CM104C/ CM14C(15,6,2)

COMMON /COMPR / KC(10)

COMMON /CONDAD/ PI,RADDEG,DEGRAD

COMMON /CONSTI/ RPM,SOUND,DENSTY,VIMOM(2),BL,R(2),STN,
. THETA0(2),HUBQ(2),DPSI,REV,CPII,TOL,CNSECT,SC00(2),RADCAS(2),
. VORCOR,DCPDT,STACK,CTII,DCTDT,ZHUB,VKTASS,TIPM(2),RDTRAN(2)

COMMON /CONST1/ PII,RC,R4PI(2),ONO4PI(2),RSCSQ

COMMON /CONTS2/ DIA(2),OMGR(2),ZMSQ,UAX,PIND(2),ZJII,MU(2)

COMMON /CORD/ ICID(300),AI(3,300),AJ(3,300),AK(3,300)

COMMON /COSNSW/ COSS(15,6,2)

COMMON /CPTHET/ NCP,ICPOK,THETO(10,2),CPCALC(10),CTCALC(10)

COMMON /CRSAV/ DATCR(1260,2)
COMMON /CRUVS / VAIOV(10,2),VTIOV(10,2),LPRNT,ICRIT,ICR,KCONV,
. CTX(2),CPX(2),DIAMX(2),ZJXX(2),BETAX(2)

COMMON /CYC1/ IS1(600),IS2(600)

COMMON /CYJOIN/ NSEG,NSIDE,ISIDE1(50),ISIDE2(50)

COMMON /C10/ F(175)

COMMON /DATIP/ DAT(700)

COMMON /DZLETE/ DLEX, ZLEX, DTEX, ZTEX

COMMON /EFFICA/ EATA, BET75 , HP

COMMON /ELCONT/ IELID

COMMON /ELEPX / CLOCD(10),DFDR(10),DTDR(10)

COMMON /EMGEST/ ELID,ISILS(3),THETA,EPS,THEST(3),
. TDFALT,FI2,FI3,FMU,Z1,Z2,
. AVGTMP,MID1,MID2,MID3,MID4,R1(3),R2(3),R3(3),IC1,IC2,IC3

COMMON /EMGPRM/ IKGG,IMGG,NOGO,ICMBAR

COMMON /FCOM/
. FTOT(15,6,2),ALFRAD(15,6,2),ALFPHI(15,6,2),ALFAXL(15,6,2),
/ FLTOT(15,6,2),ALFLRD(15,6,2),ALFLPH(15,6,2),ALFLAX(15,6,2),
. FDTOT(15,6,2),ALFDRD(15,6,2),ALFDPH(15,6,2),ALFDAX(15,6,2)

COMMON /FLIGHT/ ZMO(15,2), ZJO(15,2)

COMMON /FLOWDT/ DENS(15,2),SOUN(15,2),VONVO(15,2),URONVO(15,2),
. VZERO(15,2),VZEROB(16,2)

COMMON /FLOWFD/ AOMT(50), VTOV2A(10,50), CMAA(10,50)

COMMON /FORCOM/ FRAD(15,2),FPHI(15,2),FAXL(15,2)

COMMON /FORCON/  ZMFS, ZMROT, PHIO1O(10), AAA(10)

COMMON /F91OUT/  CGAA(10), FAD(10), SWP1(10), BETAG(10),
. CMACHG(10), CL1G(10), CD1G(10)

COMMON /GAMUV/ GAMMA(10), UX(10), VX(10)

COMMON /GAUSIN/ GCINPU(10,2), GDTHET(10,2), GTOVER(10,2),
. GAFOIL(10,2), GDESCL(10,2), GRSCIN(10,2),
. GVONVO(10,2), GURONV(10,2), GTHETA(10,2)
```

```
COMMON /GCDIMD/ GCDIMZ(15,12),GCDIMT(15,12),GCDIMR(15,12)

COMMON /GCKGSK/ GCK(100),GSK(100),IRGCK(100),ICGCK(100),NGCK,
. IRGSK(100),ICGSK(100),NGSK

COMMON /GEODAT/ RSB(16,2),ZSB(16,2),YSB(16,2),
. RSBB(16,2),ZSBB(16,2),YSBB(16,2),PHIBB(16,2),
. RSC(15,2),RSCC(15,2),ZSCC(15,2),YSCC(15,2),PHICC(15,2),
. XSBB(16,2),XSCC(15,2)

COMMON /GEOINP/ VONVOX(16,2),CX(16,2),DTHETX(16,2),TOVERX(16,2)

COMMON /GEOMOD/ DESCLP(15,2)

COMMON /GEOOUT/ DTHETA(15,2),AFOIL(15,2),DESCL(15,2),
. TOVERC(15,2),BODD(15,2)

COMMON /GGCKSK/ GGCK(100),GGSK(100),IRCK(100),ICCK(100),NGCK1,
. IRSK(100),ICSK(100),NGSK1

COMMON /HARMNC/ NHARM,IHARM(100),LOOP

COMMON/HMTOUT/ HBUF(4)

COMMON /HSQSPL/ IPTNUM,TENSON,NSUB,IPT(200),COEFFX(300),
. COEFFY(300),CHDVAL(300),ITYP,IMODE,ITRAP

COMMON /INDEX/ INDX

COMMON /INDEX1/
. IBEGIN,IXCMGG,IXEE2 ,IXZ   ,IXP002,IXKGGS,IXKDIF,IXZA  ,IXPAA ,
. IXASET,IXAS2 ,IXSPC ,IXUSPC,IXPGG ,IXASFU,IXRBE ,IXRBEF,IXDZM1,
. IXFORC,IXP001,IXEE20,IXARST,IXBUFF,IXPGGS,IXID  ,IXICP ,IXX1

COMMON /INDEX2/
. IXX2  ,IXX3  ,IXICD ,IXIEID,IXIPID,IXIG1 ,IXIG2 ,IXIG3 ,IXTHET,
. IXT1  ,IXT2  ,IXT3  ,IXPIDA,IXMID1,IXMID2,IXMID3,IXT   ,IXBSTF,
. IXTSST,IXANSM,IXZ1I ,IXZ2I ,IXMID4,IXG11 ,IXG12

COMMON /INDEX3/
. IXG13 ,IXG22 ,IXG23 ,IXG33 ,IXRHO ,IXA   ,IXTREF,IXGEI ,IXST  ,
. IXSC  ,IXSS  ,IXMSCI,IXMID ,IXCORD,IXUX1 ,IXUX2 ,IXUX3 ,IXMGS ,
. ISTART,IDZSAV

COMMON /INDEX4/
. IXIFG ,IXFCID,IXFX  ,IXFY  ,IXFZ  ,IXIFG1,IXFG1 ,IXFG11,IXFG12,
. IXIFG2,IXFG2 ,IXFG21,IXFG22,IXFG23,IXFG24,IXIMG ,IXMCID,IXFMX ,
. IXFMY ,IXFMZ ,IXIMG1,IXFMG1,IXMG11,IXMG12,IXMG2 ,IXFMG2,IXMG21,
. IXMG22,IXMG23,IXMG24,IXFEXT
```

```
COMMON /INDEX5/
. IKGG  ,IMGG ,IMELM ,IKELM ,IKECS ,IKSF  ,IB001 ,IKNM  ,IG01  ,
. IB002 ,IG02 ,IGM   ,IKMM  ,IDZ   ,IFORCE,ISPAC ,ISPAC1

COMMON /INPTP/ UBHP, UPRPM, UALT, UVKTAS, UT

COMMON /INTDT1/ ITOT,KTOT,JTOT,NTOT,BJTOT,JTOT1,IBB,LTOT,NPROP,
. MSIZE,LTOT1

COMMON /INTDT2/ NCOMPR,NEVARD,IPRMAT,IPROPT,IPNT,ITYPES,NCFLOW,
. IWAKOP,NACWAK,IJUNK,ISKIN,ICASDE,ICAS,IDEBUG,MATSOL,NCBWAK,
. IPCH,ITYPCS

COMMON /IOUNIT/ IOIN,IOOUT,IODIAG,IDEBUG,IOKAAT,IOKAA1,IOKDLM,
. IOKELM,IOMGG,IOMODE,IOSTRS

COMMON /IOUNT/ NREAD,NWRITE,NPUNCH

COMMON /IUNITD/ IUNIT(2,2)

COMMON /JUNK99/ BLANGL(90)

COMMON /KCLSAV/ XKCL(10)

COMMON /MACHNO/ ZMO

COMMON /MATIN / MATID,INFLAG,ELTEMP,SINTH,COSTH

COMMON /MATOUT/ G3X3(6),RHOY,ALPHAS(3),TZERO,GE,G2X2(3)

COMMON /MATSAV/ ANGS,ATTS

COMMON /MAXSIZ/ MAXGRD,MAXEL,MAXPSL,MAXMAT,MAXCOR,MAXRFC

COMMON /MCONED/ XMC(2),YMC(2),ZMC(2),XMCT(2),YMCT(2),ZMCT(2)

COMMON /MCRITJ/ ZMCRTJ(16)

COMMON /MCRITS/ YMCRIT(10)

COMMON /MCRIT2/ ZMCR2D(16)

COMMON /MHCONE/ NSTAT(6,2),XKCONE(15,6,2)

COMMON /MZERO / ZMSQ

COMMON /NETF/ ICALC,XEFFA,XCTA,SEFF(3),YCTA(3),SARNO(10),SVOV(10),
. JRET

COMMON /NORCOM/ ALNRAD(15,2),ALNPHI(15,2),ALNAXL(15,2)
COMMON /NUMCAS/ NNCASE,JJCASE,IPOREV

COMMON /OPTOPT/ NCALC

COMMON /PHICOM/NPHI,DELPHI(16,37),COSDPH(16,37),SINDPH(16,37)

COMMON /PHIZER/ ABOVE(15,2)/ BELOW(15,2)

COMMON /PLTFRM/ IPLTFM

COMMON /PRESCV/ UOVERV(10),VOVERV(10),SWIRLA(10),UWOVO(10),
. TEMPR(10),ZMWOMO(10),PRESRA(10),ZETAD (10),HIHO2

COMMON /RECALL/ MATRIX(20,11),ITAPE,LPERC,MAXCAS,KERREC,MBEST,IEND

COMMON /REYNXX/ REYNOS(10)

COMMON /RFRC/ RFA,RFN1,RFN2,RFN3,RFDRPM

COMMON /RFRC8/ RFA8(8),RFN18(8),RFN28(8),RFN38(8),RFDRP8(8),
. IDRF8(8)

COMMON /ROLL/ TRUNCT(2),TRUNCI(2),ROLLUP(2)

COMMON /SAVECM/ CMA1(10),ZMETHD(10),CMA(10),IJWRIT

COMMON /SAVEPL/ DCPDX(15,6,2),DCTDX(15,6,2),PHIIDE(15,6,2),
. DELX(15,2),DD(15),DT(15),TL(2),QL(2),QOL(2),QIL(2),DQ(15),PL(2),
. HPL(2),VNT(15,6,2),VCT(15,6,2),VST(15,6,2),VTT(15,6,2),
. DTL(15),DTD(15),DDL(15),DDD(15),DQI(15),DQO(15)

COMMON /SAVH/ KICON,KIFOR,KNTYPE,KNOF,KIPR,TBHP,TPRPM,TALT,
. TVKTAS,TT,TPXI,TS1,TS2,TS3,TS4,PSZJ,PSZMN

COMMON /SAVP/ PSZJ, PSS1, PSZMN, PSD342, PSD343

COMMON /SAVPSI/ SINPSI(73),COSPSI(73), PSII(73)

COMMON /SER16/ QFREQ(801),QREAL(801),QIMAG(801)

COMMON /STACOM/ STABAR(15,2),ALSRAD(15,2),ALSPHI(15,2),ALSAXL(15,2)

COMMON /STCSTR/ STRSTC(17,500)

COMMON /STRS/ S(288),E(18),TT(54),P(36),AVGTHK,INERTA

COMMON /TANVEL/ V2OV2A(10,50),ITEST

COMMON /THDESV/ DTHETS(10), DECLS(10)
COMMON /THICKD/ THK(15,2),ALTIRD(15,2),ALTIPH(15,2),ALTIAX(15,2)

COMMON /TIPDAT/ S,CO,LAMLE,LAMTE

COMMON /TITEL/ TITLE(10)

COMMON /TQLOAD/ RIN1(10), XMC41(10), DTDR1(10), DFDR1(10)

COMMON /TSOTYP/ ITSO,TSO,TEK,TEKTYP

COMMON /TWENTY/ X20(20),CLO20(20),CLD20(20),TOB20(20),ZMCH20(20),
. ZMCOS(20),ZMIN20(20),ZMEF20(20)

COMMON /TWISTC/ DTWIST(16,2), TWIST(16,2)

COMMON /TYPE/ ITYPE(500)

COMMON /UICOM/ UIR(15,6,2),UIT(15,6,2),UIZ(15,6,2)

COMMON /UUCOM/ UR(15,2),UT(15,2),UZ(15,2)

COMMON /VCOM/
. ALPHAN(15,6,2),VS(15,6,2),VC(15,6,2),VN(15,6,2),
. VTOT(15,6,2),ALVRAD(15,6,2),ALVPHI(15,6,2),ALVAXL(15,6,2)

COMMON /VIDAT/ VIS(15,6,2),VIC(15,6,2),VIN(15,6,2)

COMMON /VKARM / VKOPT

COMMON /VONCOM/ INCONQ,IXC141

COMMON /WAKDAT/ KTRUCT(2),JTRUCT(2),JTRUCI(2)

COMMON /ZJ2ZJ2/ ZJ2(10)
```

5.1 COMMON BLOCKS SORTED BY ROUTINE

Common		Routine Names					
(BLNK)	:	ABFIX	ABITR	ABRAT	ACHART	AEROMO	AEROMX
		AIRFL	AIRFLT	AZPOS	CL1AS		
		CL1BS	CNCY	CPH444	EFFIC	EFFICH	HUBXX
		INPUTA	KAPXX	LINTHT	LINTRF		
		ONEP	OPTNS	PRECOV	RDEFF	SINGL	VMOM2
		YAWAN	ZLABEL	ZNTEGR			
ABCONV	:	ABFIX	ONEP				
ABDATA	:	ABITR					
ABITER	:	ABITR	CL1AS				
ADSTMP	:	ADS402	MAIN				
AERDAT	:	BLDGEO	CALCGC	COSN	FINAIR	LMCONE	PCHOUT
		PERFOR	SOLVEL	SOLVEN	VECTOR	VVECTR	
AFCPIT	:	LCPITR	LINTHT	LINTRF			
AIRALZ	:	AIRCDM	AIRFL	AIRFLT	AIRFLX	AIROFF	AIR23
		AIR24	CASARF	DRAG24	ISOAFL		
		ISOARF	LIFT24	MCLPFN	MCRCLO	RATARF	RENUM
		REYNLS	SWPARF	THIKAR	TMCRIT	VONKAR	ZEROAL
AIRCDF	:	AIRCDF	AIRFL	AIRFLT	AIRFLX	AIROFF	AIR23
		AIR24	CASARF	DRAG24	ISOAFL		
		ISOARF	LIFT24	MCLPFN	MCRCLO	RATARF	RENUM
		REYNLS	SWPARF	THIKAR	TMCRIT	VONKAR	ZEROAL
AIRCDM	:	AIRCDM	AIRFL	AIRFLT	AIRFLX	AIROFF	AIR23
		AIR24	CASARF	DRAG24	ISOAFL		
		ISOARF	LIFT24	MCLPFN	MCRCLO	RATARF	RENUM
		REYNLS	SWPARF	THIKAR	TMCRIT	VONKAR	ZEROAL
AIRDAT	:	AIRFL	AIRFLT	AIRFLX	AIROFF	AIR23	AIR24
		CASARF	DRAG24	ISOAFL	ISOARF		
		LIFT24	MCLPFN	MCRCLO	RATARF	RENUM	REYNLS
		SWPARF	THIKAR	TMCRIT	VONKAR	ZEROAL	
ALFEXT	:	AEROMO					
ALLDAT	:	ABFIX	ABITR	ABRAT	ACHART	AEROMO	AEROMX
		AIRFL	AIRFLT	AZPOS	CL1AS		
		CL1BS	CNCY	CONECR	COSN	CPH444	EFFIC
		EFFICH	HUBXX	INPUTA	KAPXX		
		LINTHT	LINTRF	ONEP	OPTNS	PERFOR	PNPINT
		PRECOV	RDEFF	SINGL	STATIN		
		VMOM	VMOM2	YAWAN	ZLABEL	ZNTEGR	

APEFFC	:	EFFIC	SINGL				
ARGOT2	:	EFFICH	LINTHT	LINTRF			
ARGOUT	:	EFFICH	LINTHT	LINTRF	PNPINT		
BARSAV	:	GTCOOR					
BARX	:	BAR1	EMGG	GEODIG	INPUTT	PRINX	
BDSC01	:	BDSC01	BDS018				
BDSC02	:	BDS018					
BDSC03	:	BGSC01	BDS014				
BIGMAT	:	MNPANP	SETMAT	SOLVEL	SOLVEN		
CALSP	:	ONEP					
CASDAT	:	FINAIR	FVECTR	INTIAL	LINTHT	LINTRF	NSTACO
		PCHOUT	PERFOR	PNPINT	RWZW1		
		RWZW7	SOLVEL	SOLVEN	VECTOR	VVECTR	
CCOM	:	FINAIR	FVECTR	INTIAL	LINTHT	LINTRF	NSTACO
		PCHOUT	PERFOR	PNPINT	SOLVEL		
		SOLVEN	VECTOR	VVECTR	WRITGC		
CDPER	:	LINTHT	LINTRF	PCHOUT	PERFOR	PNPINT	
CINCOM	:	FINAIR	FVECTR	INTIAL	LINTHT	LINTRF	NSTACO
		PCHOUT	PERFOR	PNPINT	SOLVEL		
		SOLVEN	VECTOR	VVECTR	WRITGC		
CLCDDT	:	FINAIR	FVECTR	INDVEL	PCHOUT	PERFOR	SOLVEL
		SOLVEN					
CMOMNT	:	AIRFLT	FINAIR				
CMPNUM	:	ONEP					
CM104C	:	FINAIR	PERFOR				
COMPR	:	ONEP					
CONDAD	:	ETR3D	FEA	INTINT	PSTRSS	REORDR	STR31D
		STR32D					
CONSTI	:	BLDGEO	CALCGC	CHKINP	EFFICH	FINAIR	FVECTR
		GCWAKE	INDVEL	INTIAL	LINTHT		
		LINTRF	MNPANP	NSTACO	PCHOUT	PERFOR	PHICAL
		PNPINT	PRWZW	RWZWIN	RWZW1		
		RWZW7	SOLVEL	SOLVEN	SOLVIT	VECTOR	VVECTR
		WRITGC					

CONST1	:	BLDGEO	CALCGC	CHKINP	FINAIR	FVECTR	GCWAKE
		INDVEL	INTIAL	LINTHT	LINTRF		
		MNPANP	NSTACO	PCHOUT	PERFOR	PHICAL	PNPINT
		PRWZW	RWZWIN	RWZW1	RWZW7		
		SOLVEL	SOLVEN	SOLVIT	VECTOR	VVECTR	WRITGC
CONTS2	:	BLDGEO	CALCGC	CHKINP	FINAIR	FVECTR	GCWAKE
		INDVEL	INTIAL	LINTHT	LINTRF		
		MNPANP	NSTACO	PCHOUT	PERFOR	PHICAL	PNPINT
		PRWZW	RWZWIN	RWZW1	RWZW7		
		SOLVEL	SOLVEN	SOLVIT	VECTOR	VVECTR	WRITGC
CORD	:	CORD2R	CTMASS	FEAGEO	FEARUN	GEODIG	GETVEC
		GMPC	GP6X6	GP6X6B	INPUTT		
		PGG1	PREFRC	PREPRO	RELDEF	RLOAD	TRANSD
COSNSW	:	FINAIR	PERFOR				
CPTHET	:	CTITER	FINAIR	LCPITR	MNPANP	PCHOUT	PERFOR
		RWZW7	SOLVEL	SOLVEN			
CRSAV	:	CPH444	EFFIC	SINGL			
CRUVS	:	CPH444	EFFIC	ONEP	SINGL	VMOM2	
CYC1	:	CYCLIC	INPUTT	PREPRO	REDUCE		
CYJOIN	:	CYCLC2	CYCLIC	INPUTT	PREPRO		
C10	:	C10					
DATIP	:	ONEPR	YAWAN				
DZLETE	:	CALSW2	CONECR				
EFFICA	:	EFFICH	PERFOR				
ELCONT	:	BAR1	EMGG	EMGPOM	ETR3D	PRINX	PSTRSS
ELEPX	:	ONEP					
EMGDS1	:	BAR1	CYCLC2	EMA	EMGG	ETR3D	FEA
		INTINT	PRINX				
EMGEST	:	BAR1	EMGG	ETR3D	PRINX	PSTRSS	RELDEF
		STR31D	STR32D				
EMGPRM	:	EMGG	ETR3D	PRINX			
FCOM	:	FVECTR	PCHOUT	PERFOR			

```
FLIGHT   :   FINAIR   INTIAL   PCHOUT   PERFOR   RWZW1    SOLVEL
             SOLVEN   VECTOR   VVECTR

FLOWDT   :   FINAIR   FVECTR   INTIAL   LINTHT   LINTRF   NSTACO
             PCHOUT   PERFOR   PNPINT   RWZW1
             RWZW7    SOLVEL   SOLVEN   VECTOR   VVECTR

FLOWFD   :   AEROMO   AIRFL    CL1BS    ONEP

FORCOM   :   PCHOUT   PERFOR

FORCON   :   CONECR   ONEP     VMOM2

F91OUT   :   EFFICH   PERFOR   PNPINT

GAMUV    :   ONEP

GAUSIN   :   LINTHT   LINTRF   PNPINT

GCDIMD   :   CALCGC   GCWAKE   WRITGC

GCKGSK   :   CYCLC2   CYCLIC   INPUTT

GEODAT   :   BLDGEO   CALCGC   FINAIR   FVECTR   GCWAKE   INDVEL
             INTIAL   LINTHT   LINTRF   MNPANP
             NSTACO   PCHOUT   PERFOR   PNPINT   PRWZW    RWZW1
             RWZW7    SOLVEL   SOLVEN   VECTOR   VVECTR

GEOINP   :   BLDGEO   CALCGC   FINAIR   FVECTR   GCWAKE   INDVEL
             INTIAL   LINTHT   LINTRF   MNPANP
             NSTACO   PCHOUT   PERFOR   PNPINT   PRWZW    RWZW1
             RWZW7    SOLVEL   SOLVEN   VECTOR   VVECTR

GEOMOD   :   BLDGEO   CALCGC   FINAIR   FVECTR   GCWAKE   INDVEL
             INTIAL   LINTHT   LINTRF   MNPANP
             NSTACO   PCHOUT   PERFOR   PNPINT   PRWZW    RWZW1
             RWZW7    SOLVEL   SOLVEN   VECTOR   VVECTR

GEOOUT   :   BLDGEO   CALCGC   FINAIR   FVECTR   GCWAKE   INDVEL
             INTIAL   LINTHT   LINTRF   MNPANP
             NSTACO   PCHOUT   PERFOR   PNPINT   PRWZW    RWZW1
             RWZW7    SOLVEL   SOLVEN   VECTOR   VVECTR

GGCKSK   :   CYCLC2   CYCLC3

HARMNC   :   INPUTT

HMTOUT   :   ETR3D
```

HSQSPL	:	CALSW2	EFFICH	GAEROH	LMCONE	PERFOR	PNPINT
		QSPLIN					
INDEX	:	BDS014					
INDEX1	:	FEA	FEAGEO	FEARUN	INPUTT	ONEPFR	PREMAP
INDEX2	:	FEA	FEAGEO	FEARUN	INPUTT	ONEPFR	PREMAP
INDEX3	:	FEA	FEAGEO	FEARUN	INPUTT	ONEPFR	PREMAP
INDEX4	:	FEA	FEAGEO	FEARUN	INPUTT	ONEPFR	PREMAP
INDEX5	:	FEA	FEARUN	ONEPFR			
INPTP	:	INPUTA	LINTHT	LINTRF	RDEFF		
INPT23	:	INPUTA	LINTHT	LINTRF	ONEP	RDEFF	
INTDT1	:	BLDGEO	CALCGC	CHKINP	EFFICH	FINAIR	FVECTR
		GCWAKE	INDVEL	INTIAL	LINTHT		
		LINTRF	MNPANP	NSTACO	PCHOUT	PERFOR	PHICAL
		PNPINT	PRWZW	RWZWIN	RWZW1		
		RWZW7	SOLVEL	SOLVEN	SOLVIT	VECTOR	VVECTR
		WRITGC					
INTDT2	:	BLDGEO	CALCGC	CHKINP	FINAIR	FVECTR	GCWAKE
		INDVEL	INTIAL	LINTHT	LINTRF		
		MNPANP	NSTACO	PCHOUT	PERFOR	PHICAL	PNPINT
		PRWZW	RWZWIN	RWZW1	RWZW7		
		SOLVEL	SOLVEN	SOLVIT	VECTOR	VVECTR	WRITGC
IOUNIT	:	ADDCD	ADSRD	ALOAD	ASEMBL	BANDER	BAR1
		BMADD	CHKLOG	CNN	CNSTRN		
		CNVRG1	CNVRT	COMAP	COPY88	CTMASS	DEFCON
		EIGEN	EIGS	ELAREA	EMA		
		EMGG	EMGPOM	ETR3D	FEA	FEAGEO	FEARUN
		FFORCE	FREQTB	FREQY	GEODIG		
		GETMAT	GMMATD	GMPC	GP6X6	GP6X6B	GTCOOR
		GTMODE	IDENT	INPUTT	INTINT		
		KGG1	KNN	LAJA	LAMIN8	LSTRAN	MAIN
		MAPPER	MATCMP	MERGE	MGG1		
		MODPRT	ONEPFR	OPT001	OPT002	OPT003	OPT004
		OPT005	OPT007	OPT008	OPT009		
		PARAM	PFORCE	PGG1	PRECMP	PRELAM	PREPRO
		PRINX	PRTBBB	PRTBFB	PRTFFF		
		PSTCMP	QIJ	RAJA	RDEFF	RDGGEN	RDOBJ
		REBAND	REDUCE	RELDEF	REORDR		
		REPORT	RESTOR	RLOAD	SCLRVR	SPCARD	SPCARR
		SPRING	STATIN	STRAIN	STRESS		
		STRPRT	STR31D	STR32D	THKEFF	TRANSD	TSAIWU
		VMERGE					

```
IOUNT    :   AF65A    BLDGEO    CALCGC    CHKINP    CSCD1     FINAIR
             FVECTR   GCWAKE    INDVEL    INTIAL
             LINTHT   LINTRF    LMCONE    MNPANP    NSTACO    PCHOUT
             PERFOR   PNPINT    PRWZW     RWZWIN
             RWZW1    RWZW7     SOLVEL    SOLVEN    SWPCOR    TIPGEO
             VECTOR   VVECTR    WAKMOD    WRITGC

ITER     :   ONEP

IUNITD   :   BLDGEO   CALCGC    CHKINP    FINAIR    FVECTR    GCWAKE
             INDVEL   INTIAL    IUNITD    LINTHT
             LINTRF   NSTACO    PCHOUT    PERFOR    PHICAL    PNPINT
             PRWZW    RWZWIN    RWZW1     RWZW7
             SOLVEL   SOLVEN    SOLVIT    VECTOR    VVECTR    WRITGC

JUNK99   :   ABFIX    ABITR     ABRAT     ACHART    AEROMO    AEROMX
             AIRFL    AIRFLT    AZPOS     CL1AS
             CL1BS    CNCY      CPH444    EFFIC     EFFICH    HUBXX
             INPUTA   KAPXX     ONEP      OPTNS
             PRECOV   RDEFF     SINGL     VMOM2     YAWAN     ZLABEL
             ZNTEGR

KCLSAV   :   ISOARF

MACHNO   :   EFFICH   PNPINT

MATIN    :   ETR3D    MAT       STR31D

MATOUT   :   ETR3D    MAT       STR31D

MATSAV   :   SCLRVR

MAXSIZ   :   INPUTT

MCONED   :   BLDGEO   CALCGC    FINAIR    FVECTR    GCWAKE    INDVEL
             INTIAL   LINTHT    LINTRF    MNPANP
             NSTACO   PCHOUT    PERFOR    PNPINT    PRWZW     RWZW1
             RWZW7    SOLVEL    SOLVEN    VECTOR    VVECTR

MCRITJ   :   AIRFLT

MCRITS   :   AIR23

MCRIT2   :   AIRFLT   EFFICH    MCRCLO

MHCONE   :   FINAIR   INTIAL    PCHOUT    PERFOR    RWZW1     SOLVEL
             SOLVEN   VECTOR    VVECTR

MZERO    :   AIRFLT
```

NETF	:	CPH444	ONEP	SINGL			
NORCOM	:	FINAIR	FVECTR	INTIAL	LINTHT	LINTRF	NSTACO
		PCHOUT	PERFOR	PNPINT	SOLVEL		
		SOLVEN	VECTOR	VVECTR	WRITGC		
NUMCAS	:	ONEP					
OPTOPT	:	ONEP					
PHICOM	:	CALCGC	WAKMOD				
PHIZER	:	PERFOR	PNPINT				
PLTFRM	:	ELAREA	MAIN	PRECMP	PRELAM	PSTCMP	THKEFF
		TOTARE					
PRESCV	:	ONEP	PRECOV				
RECALL	:	ABFIX	ABITR	ABRAT	ACHART	AEROMO	AEROMX
		AIRFL	AIRFLT	AZPOS	CL1AS		
		CL1BS	CNCY	CPH444	EFFIC	EFFICH	HUBXX
		INPUTA	KAPXX	ONEP	OPTNS		
		PRECOV	RDEFF	SINGL	VMOM2	YAWAN	ZLABEL
		ZNTEGR					
REYNXX	:	ABFIX	ABITR	ABRAT	ACHART	AEROMO	AEROMX
		AIRFL	AIRFLT	AZPOS	CL1AS		
		CL1BS	CNCY	CPH444	EFFIC	EFFICH	HUBXX
		INPUTA	KAPXX	ONEP	OPTNS		
		PRECOV	RDEFF	SINGL	VMOM2	YAWAN	ZLABEL
		ZNTEGR					
RFRC	:	CHKLOG	CTMASS	FREQTB	FREQY	INPUTT	ONEPFR
		RLOAD					
RFRC8	:	CHKLOG	INPUTT				
ROLL	:	INTIAL	LINTHT	LINTRF	PNPINT		
SAVECM	:	AEROMO	AEROMX	ONEP			
SAVEPL	:	PCHOUT	PERFOR				
SAVH	:	ONEP					
SAVP	:	LINTHT	LINTRF				

SAVPSI	:	BLDGEO	CALCGC	CHKINP	FINAIR	FVECTR	GCWAKE
		INDVEL	INTIAL	LINTHT	LINTRF		
		MNPANP	NSTACO	PCHOUT	PERFOR	PHICAL	PNPINT
		PRWZW	RWZWIN	RWZW1	RWZW7		
		SOLVEL	SOLVEN	SOLVIT	VECTOR	VVECTR	WRITGC
SER16	:	SER16					
STACOM	:	FINAIR	FVECTR	INTIAL	LINTHT	LINTRF	NSTACO
		PCHOUT	PERFOR	PNPINT	SOLVEL		
		SOLVEN	VECTOR	VVECTR	WRITGC		
STCSTR	:	BAR1	FEA	ONEPFR	PRINX	PSTCMP	PSTRSS
		STRPRT					
STRS	:	EMGG	PRINX	STR31D	STR32D		
TANVEL	:	ABFIX	ABITR	ABRAT	ACHART	AEROMO	AEROMX
		AIRFL	AIRFLT	AZPOS	CL1AS		
		CL1BS	CNCY	CPH444	EFFIC	EFFICH	HUBXX
		INPUTA	KAPXX	LINTHT	LINTRF		
		ONEP	OPTNS	PRECOV	RDEFF	SINGL	VMOM2
		YAWAN	ZLABEL	ZNTEGR			
THDESV	:	EFFICH	ONEP				
THICKD	:	FINAIR	FVECTR	INTIAL	LINTHT	LINTRF	NSTACO
		PCHOUT	PERFOR	PNPINT	SOLVEL		
		SOLVEN	VECTOR	VVECTR	WRITGC		
TIPDAT	:	LMCONE	TIPGEO				
TITEL	:	INPUTT					
TQLOAD	:	EFFICH	PERFOR				
TSOTYP	:	EFFICH	INPUTA	ONEP	OPTNS		
TWENTY	:	TWENTY					
TWISTC	:	FINAIR	MNPANP	PERFOR	PNPINT	SOLVEL	SOLVEN
TYPE	:	BANDER	CTRIA	EMA	EMGG	FEARUN	GEODIG
		INPUTT	PREPRO	PRINX	STRPRT		
UICOM	:	FVECTR	INDVEL	PCHOUT	PERFOR	SOLVEN	VVECTR
UUCOM	:	BLDGEO	FVECTR	INTIAL	NSTACO	PCHOUT	PERFOR
		SOLVEL	SOLVEN	VECTOR	VVECTR		

VCOM	:	BLDGEO	FVECTR	INTIAL	NSTACO	PCHOUT	PERFOR
		SOLVEL	SOLVEN	VECTOR	VVECTR		
VIDAT	:	FVECTR	INDVEL	PCHOUT	PERFOR	SOLVEN	VVECTR
VKARM	:	AIRFLT					
VONCOM	:	AIRFLT	LINTHT	LINTRF			
WAKDAT	:	CALCGC	INTIAL	RWZW7			
ZJ2ZJ2	:	ABFIX	ABITR	ABRAT	ACHART	AEROMO	AEROMX
		AIRFL	AIRFLT	AZPOS	CL1AS		
		CL1BS	CNCY	CPH444	EFFIC	EFFICH	HUBXX
		INPUTA	KAPXX	ONEP	OPTNS		
		PRECOV	RDEFF	SINGL	VMOM2	YAWAN	ZLABEL
		ZNTEGR					

6.0 ADDING REFINED ANALYSIS MODULES TO STAT

Due to the proprietary nature of the STAT refined acoustic and flutter modules, only dummy functional modules are currently provided in the public distribution version of STAT. To enable an independent user to insert his/her own analyses into STAT, subroutine call statements have been included in the refined analysis control section of the code. The full analysis calling statements from the refined analysis control module are included below, with module inputs and outputs listed.

```
      SUBROUTINE FLUTRR(NSTAS,GL,GT,GM,NFREQ,DGM,FREQX,GENMAS,XNB,
     1                  RPM,SOS,RHO,STALL,FMNEW,IDBG08,INFO,LOOPK)
C
C     REFINED FLUTTER ANALYSIS
C
C INPUT
C    NSTAS   -  Number of radial stations
C    GL      -  Leading edge displaced geometry  - xyz coordinates (in.)
C    GT      -  Trailing edge displaced geometry - xyz coordinates (in.)
C    GM      -  Mid-chord displaced geometry - xyz coordinates (in.)
C               Note:  GL, GT, and GM are the finite element
C                      coordinates for the hot (running position)
C                      blade, and are output from the finite element
C                      analysis. The nodal positions align with the
C                      locations at which the mode shapes, DGM, are
C                      output.
C    NFREQ   -  Number of frequencies (hz)
C    DGM     -  Mode shapes at mid-chord in the swept normal coordinate
C               system for all six degrees of freedom (in. and rad.)
C    FREQX   -  Natural frequencies (hz)
C    GENMAS  -  Generalized  masses (in-lb-s**2)
C    XNB     -  Number of blades
C    RPM     -  Propeller speed (rpm)
C    SOS     -  Speed of sound  (fps)
C    RHO     -  Air density (slugs / cu. ft.)
C    IDBG08  -  Debug indicator
C                 0 - Skip diagnostic messages
C                 1 - Print diagnostic messages
C    INFO    -  Optimization status indicator
C                 0 - Optimizer found an optimum design
C                 1 - Refined analysis being performed at requested
C                     design iteration
C    LOOPK   -  Approximate optimization loop counter
C
C OUTPUT
C    STALL   -  Stall flutter parameter
C    FMNEW   -  Classical flutter Mach number, least stable mode
C
C
```

```
      REAL*8    GL, GT, GM, FREQX, GENMAS, DGM
      DIMENSION GL(20,3), GT(20,3), GM(20,3), FREQX(5), GENMAS(5,5),
     1          DGM(20,6,5)

C
C     User coding
C
      RETURN
      END


      SUBROUTINE NOISE(XMO,XMT,TEMP,XNB,DIAM,X,HOB,BOD,CGA,ACB,FAD,BET,
     1           CLX,CDX,REL,TIPCL,POP,CLD,CRT,SWP,EXP,BPF1,TDB1,WORK)
C
C     REFINED ACOUSTIC ANALYSIS
C
C INPUT
C    XMO      -  Flight Mach number
C    XMT      -  Tip rotational Mach number
C    TEMP     -  Ambient temperature (F)
C    XNB      -  Number of blades
C    DIAM     -  Blade diameter (ft)
C    X        -  Gauss stations and tip station (X(1)=root  X(11)=1.0)
C    HOB      -  Thickness to chord ratio at 10 Gauss stations
C    BOD      -  Chord to diameter ratio at 10 Gauss and tip station
C    CGA      -  C.G. over diameter at 10 Gauss stations
C    ACB      -  Aero center (from l.e. to c.g.) at 10 Gauss stations
C    FAD      -  Face alignment over diameter at 10 Gauss stations
C    BET      -  Blade Twist (Beta) at 10 Gauss stations
C    CLX      -  Lift coefficient at 10 Gauss stations
C    CDX      -  Drag coefficient at 10 Gauss stations
C    REL      -  Section relative Mach number (10 Gauss stations)
C    TIPCL    -  Tip clearance (in blade diameters)
C    POP      -  Pressure ratio
C    CLD      -  Design lift coefficient at 10 Gauss stations
C    CRT      -  2-D critical Mach numbers at 10 Gauss stations
C                   (Created in Subroutine EFFICH of the aero module)
C    SWP      -  Aerodynamic sweep at 10 Gauss stations
C    EXP      -  Sweep exponents at 10 Gauss stations
C                   (Created in Subroutine EFFICH of the aero module)
C    WORK     -  Work array, used as scratch storage - 280,000 Words of
C                storage are available.
C
C OUTPUT
C    BPF1     -  Frequency of blade passage harmonic
C    SPL1     -  Maximum SPL located on fuselage surface, db
C
C
```

```
      DIMENSION X(11),HOB(10),BOD(11),CGA(10),ACB(10),FAD(10),BET(10),
     1 BET(10),CLX(10),CDX(10),REL(10),CLD(10),CRT(10),SWP(10),EXP(10),
     2 WORK(1)
C
C     User coding
C
      RETURN
      END
```

6.1 WORK STORAGE ARRAY

The STAT program is a large, multi-disciplinary system, tying together a number of diverse analyses. Due to the number of program modules employed, computer storage space has always been in short supply. To provide adequate storage without requiring program overlays, STAT utilizes a large work storage area (currently 280,000 words in length) that is accessible for use by all modules.

Within the approximate optimization loop, this work array contains details of the finite element mesh, as well as displacement and stress results, and vibratory mode shapes. These arrays are used by the finite element module, the geometry update module, the flutter module, and others. Areas of the work array beyond the space required for the above matrices may be used as temporary scratch storage space by any of the modules, thus alleviating local matrix storage requirements.

Due to the higher storage requirements of detailed, refined analyses, the biggest core storage requirements are seen in the final, refined analysis step. For refined analyses, the entire work storage area is available. This use of local scratch storage is recommended where possible to reduce overall storage requirements.

7.0 JCL FOR THE CRAY COS VERSION OF THE STAT PROGRAM

This section is intended to provide instructions for setting up job control for STAT on the NASA-Lewis Research Center CRAY XMP utilizing COS 1.14BF4. Note that this set of instructions is intended as a guideline only; dataset names may vary and local system modifications may necessitate JCL modifications.

7.1 EXECUTING THE STAT PROGRAM

The following JCL procedure may be submitted from VM or TSS. It will execute STAT and route the output, diagnostic, and summary report files back to the front-end machine.

Determining the exact time required for a STAT optimization run is a complex task, and very problem dependent. Experience, however, has shown that the following formula provides a good time estimation:

Time = 8 * (NDV * 20 + 30) seconds, NDV = number of design variables.

```
JOB,JN=jobname,MFL=1000000,T=time.
ACCOUNT,AC=acct,APW=pw.
ACCESS,DN=STAT,PDN=STATLIB,ID=SMSTAT,OWN=SMSTAT.
LDR,DN=STAT,LIB=IMSLLIB.
DISPOSE,DN=FT19,SDN=DIAG.
DISPOSE,DN=FT40,SDN=REPORT.
/EOF
Insert STAT input data here
/EOF
```

7.2 UPDATING THE STAT PROGRAM

The following JCL procedure may be submitted from VM or TSS. It will compile FORTRAN routine(s) and update the STAT object module. Output is sent back to the front-end machine.

NOTE: Prior to submitting this procedure, the user must be permitted WRITE access to STATLIB from userid SMSTAT.

```
JOB,JN=jobname,MFL=100000,T=time.
ACCOUNT,AC=acct,APW=pw.
ACCESS,DN=$OBL,PDN=STATLIB,ID=SMSTAT,OWN=SMSTAT,UQ.
CFT.
BUILD.
DELETE,DN=$OBL.
SAVE,DN=$NBL,PDN=STATLIB,ID=SMSTAT,OWN=SMSTAT,PAM=R.
/EOF
Insert FORTRAN update(s) here
/EOF
```

8.0 REFERENCES

1. Brown, K. W., Structural Tailoring of Advanced Turboprops (STAT) Interim Report, NASA CR 180861, August 1988.

2. Vanderplaats, G. N., H. Sugimoto and C. M. Sprague, "ADS-1: A New General Purpose Optimization Program," AIAA 24th Structures, Structural Dynamics and Materials Conference, Lake Tahoe, Nevada, May 1983.

3. Brown, K. W. and P. R. Harvey, Structural Tailoring of Advanced Turboprops (STAT) User's Manual (Draft), Pratt & Whitney PWA-5967-43, March 1987.

DISTRIBUTION LIST

NASA-Lewis Research Center
Attn: Contracting Officer, MS 500-313
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lewis Research Center
Attn: Tech. Report Control, MS 60-1
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lewis Research Center
Attn: Tech. Utilization Office, MS 7-3
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lewis Research Center
Attn: Div Contract File, MS 49-6 (2 copies)
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lewis Research Center
Attn: Library, MS 60-3
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lewis Research Center
Attn: Adv. Turboprop Project, MS 86-7
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lewis Research Center
Attn: R. E. Kielb, MS 49-8
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lewis Research Center
Attn: D. A. Hopkins, MS 49-8 (2 copies)
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lewis Research Center
Attn: C. C. Chamis, MS 49-8
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lewis Research Center
Attn: L. Berke, MS 49-6
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lewis Research Center
Attn: L. J. Kiraly, MS 23-3
21000 Brookpark Road
Cleveland, OH 44135

NASA-Lewis Research Center
Attn: T. O. Cressman, MS 86-10
21000 Brookpark Road
Cleveland, OH 44135

National Aeronautics & Space
Administration
Attn: NHS-22/Library
Washington, DC 20546

NASA S&T Information Facility
Attn: Acquis. Dept. (5 copies)
P. O. Box 8757
B.W.I. Airport, MD 21240

NASA Ames Research Center
Attn: Library
Mail Stop 202-3
Moffett Field, CA 94035

NASA John F. Kennedy Space Center
Attn: Library
Mail Stop AD-CSO-1
Kennedy Space Center, FL 32931

NASA Langley Research Center
Attn: Library
Mail Stop 185
Hampton, VA 23665-5225

NASA Langley Research Center
Attn: M. F. Card
Mail Stop 244
Hampton, VA 23665-5225

NASA Goddard Space Flight Center
Attn: 252/Library
Greenbelt, MD 20771

NASA LBJ Space Center
Attn: JM6/Library
Houston, TX 77001

DISTRIBUTION LIST (continued)

NASA Marshall Space Flight Center
Attn: AS61/Library
Marshall Space Flight Center, AL 35812

Jet Propulsion Laboratory
Attn: Library
4800 Oak Grove Drive
Pasadena, CA 91103

AFFDL/FBE
Attn: D. W. Smith
Wright-Patterson AFB, OH 45433

Air Force Aero. Propulsion Lab.
Attn: N. Khot
Wright-Patterson AFB, OH 45433

Air Force Systems Command
Aeronautical Systems Division
Attn: Library
Wright-Patterson AFB, OH 45433

Department of the Army
U. S. Army Material Command
Attn: AMCRD-RC
Washington, DC 20315

U. S. Army Missile Command
Redstone Scientific Info. Center
Attn: Document Section
Redstone Arsenal, AL 35808

Commanding Officer
U. S. Army Research Office (Durham)
Attn: Library
Box CM, Duke Station
Durham, NC 27706

Bureau of Naval Weapons
Department of the Navy
Attn: RRRE-6
Washington, DC 20360

Commander, U. S. Naval Ordnance Lab.
Attn: Library
White Oak
Silver Springs, MD 20910

Director, Code 6180
U. S. Naval Research Laboratory
Attn: Library
Washington, DC 20390

Aerospace Corporation
Attn: Library-Documents
2400 E. El Segundo Blvd.
Los Angeles, CA 90045

Boeing Company
Attn: Library
Wichita, KA 67201

General Dynamics
Attn: Library
P. O. Box 748
Fort Worth, TX 76101

General Dynamics/Convair Aero.
Attn: Library
P. O. Box 1128
San Diego, CA 92112

General Electric Company
Attn: Dr. R. L. McKnight, MS K221
Interstate 75, Bldg. 500
Cincinnati, OH 45215

Grumman Aircraft Eng. Corp.
Attn: Library
Bethpage, Long Island, NY 11714

Grumman Aircraft Eng. Corp.
Attn: H. A. Armen
Bethpage, Long Island, NY 11714

Lawrence Livermore Laboratory
Attn: Library
P. O. Box 808, L-42
Livermore, CA 94550

McDonnell Douglas Aircraft Corp.
Attn: Library
P. O. Box 516
Lambert Field, MO 63166

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.
NASA CR-182164

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Structural Tailoring of Advanced Turboprops (STAT)
Programmer's Manual

5. Report Date
March 1989

6. Performing Organization Code

7. Author(s)
K. W. Brown and P. R. Harvey

8. Performing Organization Report No.
PWA-5967-51

10. Work Unit No.

9. Performing Organization Name and Address
UNITED TECHNOLOGIES CORPORATION
Pratt & Whitney, Commercial Engineering
400 Main St., East Hartford, CT 06108

11. Contract or Grant No.
NAS3-23941

13. Type of Report and Period Covered
Programmer's Manual

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Lewis Research Center
21000 Brookpark Road, Cleveland, Ohio 44135

14. Sponsoring Agency Code

15. Supplementary Notes

Program Manager: D. Hopkins
NASA-Lewis Research Center, Cleveland, Ohio

16. Abstract

The Structural Tailoring of Advanced Turboprops (STAT) computer program was developed to perform numerical optimizations on highly swept propfan blades. This manual describes the functionality of the STAT system from a programmer's viewpoint. It provides a top-down description of module intent and interaction. The purpose of this manual is to familiarize the programmer with the STAT system should he/she wish to enhance or verify the program's function.

17. Key Words (Suggested by Author(s))
Approximate Analysis; Mathematical Optimization; Objective Function; Refined Analysis; User Instruction; Input; Output

18. Distribution Statement

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of pages
49

22. Price

DISTRIBUTION LIST (continued)

Northrop Space Laboratories
Attn: Library
3401 West Broadway
Hawthorne, CA 90250

North American Rockwell, Inc.
Rocketdyne Division
Attn: Library, Dept. 596-306
6633 Canoga Avenue
Canoga Park, CA 91304

North American Rockwell, Inc.
Space & Info. Systems Div.
Attn: Library
12214 Lakewood Blvd.
Downey, CA 90241

Republic Aviation
Fairchild Hiller Corporation
Attn: Library
Farmingdale, NY 11735

IIT Research Institute
Technology Center
Attn: Library
Chicago, IL 60616

United Technologies Corporation
Pratt & Whitney
Attn: Library, MS 732-11
P. O. Box 109600
West Palm Beach, FL 33410-9600

United Technologies Corporation
Pratt & Whitney
Attn: Library, MS 169-31
400 Main Street
East Hartford, CT 06108

United Technologies Corporation
Hamilton Standard
Attn: A. Weisbrich
Windsor Locks, CT 06096

United Technologies Corporation
Hamilton Standard
Attn: F. Metzger
Windsor Locks, CT 06096

United Technologies Corporation
Hamilton Standard
Attn: R. Sherman
Windsor Locks, CT 06096

United Technologies Corporation
Hamilton Standard
Attn: J. Baum
Windsor Locks, CT 06096

United Technologies Corporation
Hamilton Standard
Attn: B. Gatzen
Windsor Locks, CT 06096