

LANGLEY RESEARCH CENTER



3 1176 00509 1427

NASA Contractor Report 159007

NASA-CR-159007

19790014611

PARS - PROGRAMS FOR ANALYSIS AND RESIZING
OF STRUCTURES - USER MANUAL

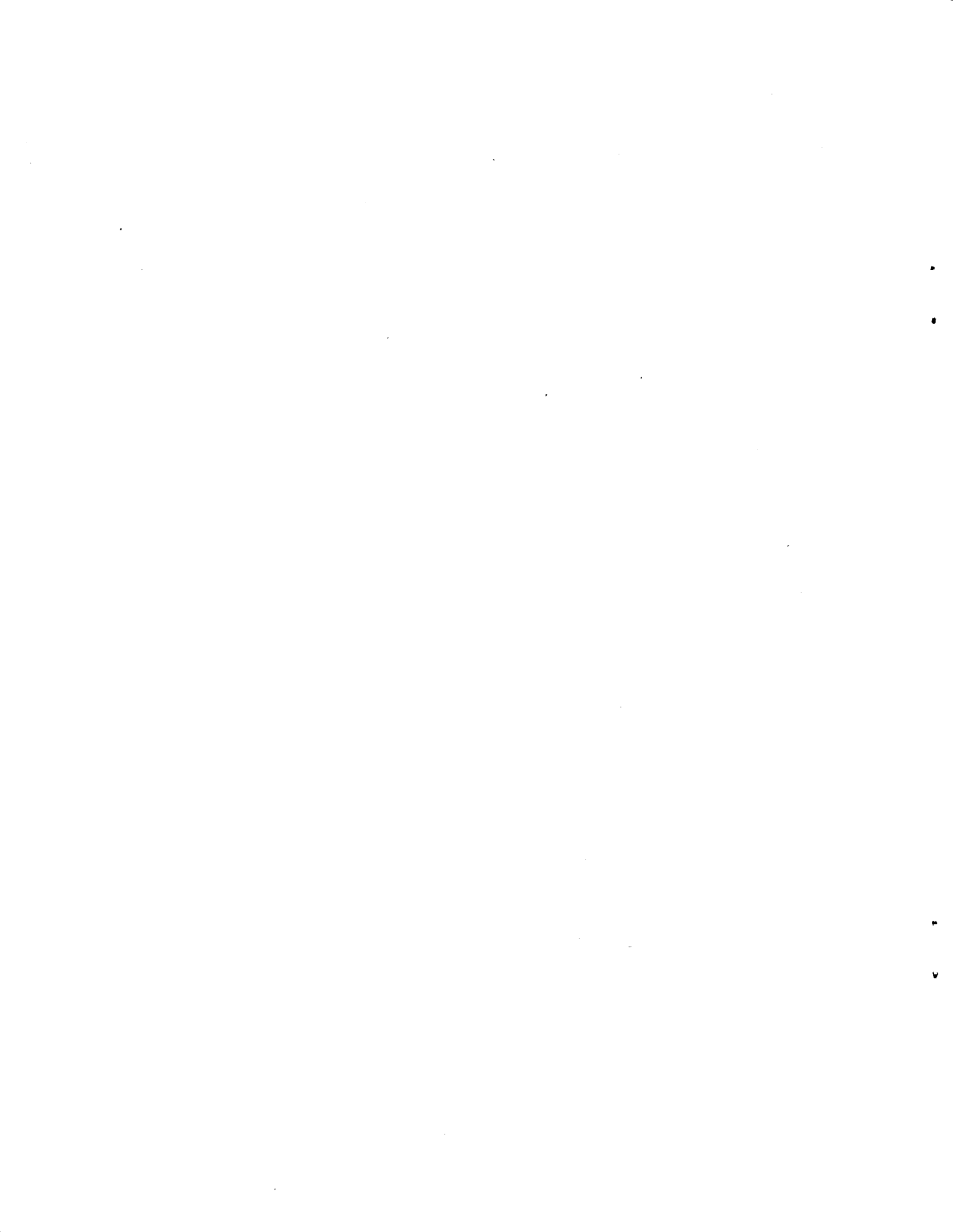
R. T. Haftka, B. Prasad, and U. Tsach

ILLINOIS INSTITUTE OF TECHNOLOGY
Chicago, Illinois 60616

NASA Grant NSG-1266
April 1979

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665



CONTENTS

	Page
1. INTRODUCTION	1
1.1 PARS processors and their use	2
1.2 Input Rules	5
2. DESIGN VARIABLE DEFINITION AND INTERFACE WITH FINITE ELEMENT MODEL	8
2.1 DESVAR - Design Variable Definition	9
2.2 KSDM - Structural Parameter Update	16
3. STATIC CONSTRAINTS AND THEIR DERIVATIVES	21
3.1 DKS - Derivatives of K Matrices	21
3.2 DGRAD - Gradient of Displacement Generator	22
3.3 TGRAD - Derivatives of Displacement Vectors With Respect to Design Variables Controlling Nodal Temperatures	24
3.4 SGRAD - Stress Gradient Generator	28
3.5 DCON - Displacement Ratio Generator	31
3.6 SCON - Stress Ratio Generator	34
4. FLUTTER ANALYSIS AND DERIVATIVES	44
4.1 AERO - Aerodynamic Input Processor	45
4.2 SUBKRN - Subsonic Kernel Function Aerodynamics Processor	49
4.3 GAF - Generalized Aerodynamic Force Generator	51
4.4 DKM - Generalized Stiffness and Mass Matrix Derivative Processor	53
4.5 FLT - Flutter Analysis Processor	54
5. OPTIMIZATION	59
5.1 NEWTON - Newton's Method Optimizer	61
5.2 SOL - Static Optimization Loop	67

		Page
5.3	FOL - Flutter Optimization Loop	74
5.4	FSDL - Fully Stressed Design Loop	79

SECTION 1

INTRODUCTION

PARS is an efficient optimization program for resizing structures subject to stress, displacement and flutter constraints. The optimization method and the static sensitivity calculations are developed in Ref. 1. The aerodynamic and flutter analysis and corresponding sensitivity calculations capabilities of PARS are based on the WIDOWAC program [2]. PARS is based on the SPAR finite element system [3] and like SPAR it is composed of a group of individual processors. These processors are used in a logical sequence to perform a desired analysis or synthesis task. The processors are able to communicate directly and automatically with a "Data complex" which contains the information generated by other processors during a computer run. A general characteristic of the technical processors is that they are efficient with respect to both core store and central processing time requirement. PARS provides an efficient code for sizing large or small scale finite element models in the presence of strength, thermal and aeroelastic constraints with minimum and maximum bounds on structural dimensions.

Three standard built in looping systems, provided for carrying out static strength optimization, flutter optimization, are available. The optimization task can also be achieved independently by calling a few processors, examining the intermediate results and then entering into the loop or continue calling each individual processors sequentially.

The optimization procedure is based on a mathematical programming method using the sequence of unconstrained minimization technique (SUMT) in which design requirements are incorporated using an extended interior penalty function approach. Approximate analysis techniques, design variable linking, constraint deletion and approximate expressions for the second derivatives of the constraints are used in conjunction with Newton's method to reduce computational time and increase the order of the structural design problem that can be solved. In addition PARS may be used with the stress ratio technique to obtain a fully stressed design when the structure is subject only to stress constraints. Two versions of stress ratio technique are available to the user. The first one is the well known fully stressed design (FSD) and the second is the thermal fully stressed design, TFSD [4].

PARS can be used to resize structural systems that can be idealized using truss, beam, shear panels, and triangular or quadrilateral membrane or plate elements. All elements except the beam elements may be resized. It is assumed that this manual would be used as a companion to the SPAR Reference Manual [3] and that the reader is familiar with the use of the SPAR program and its terminology for input/output and data complex.

1.1 PARS processors and their use

There are several ways in which PARS could be used by the structural designer. The following are a few suggested sequences for calling SPAR and PARS processors to accomplish analysis, sensitivity and resizing tasks.

- (i) Calculation of derivatives of displacement and stresses with respect to one or several design variables.

- a. Use SPAR to calculate displacement and stresses (processors TAB, ELD, ... GSF)
- b. @ XQT DESVAR - define design variables
- c. @ XQT DKS - calculate derivatives of stiffness matrices (needed only if design variables control plate bending elements)
- d. @ XQT DGRAD - derivative of displacements with respect to thickness or area variables
- e. @ XQT TGRAD - derivatives of displacements with respect to temperatures design variables
- f. @ XQT SGRAD - derivatives of stresses

(ii) Flutter Analysis

- a. Use SPAR to calculate vibration modes and frequencies (processors TAB, ELD ... EIG)
- b. @ XQT AERO - define aerodynamic model
- c. @ XQT SUBKRN - calculate aerodynamic matrices
- d. @ XQT GAF - calculate generalized aerodynamic forces
- e. @ XQT FLT - flutter analysis

(iii) Sensitivity analysis for stresses, one resizing move and reanalysis

- a. Use SPAR to calculate displacement and stresses
- b. @ XQT DESVAR
- c. @ XQT DKS (if resizing plate bending elements)
- d. @ XQT DGRAD
- e. @ XQT TGRAD (if using temperature design variables)
- f. @ XQT SGRAD
- g. @ XQT DCON - calculate displacement ratio for displacement constraints, if any

- h. @ XQT SCON - calculate stress ratios
- i. @ XQT NEWTON - one resizing move, new design variables.
- j. @ XQT KSDM - update finite element model for new design variables
- k. Use SPAR processors starting with E, EKS, to calculate new displacement and stresses.

(iv) Flutter analysis, one resizing move and reanalysis

- a. Use SPAR to calculate displacements and stresses
- b. @ XQT DESVAR
- c. @ XQT AERO
- d. @ XQT SUBKRN
- e. @ XQT GAF
- f. @ XQT DKM - calculate derivatives of generalized stiffness and mass matrices
- g. @ XQT FLT - flutter analysis and derivatives
- h. @ XQT NEWTON - one resizing move, new design variables
- i. @ XQT FLT - new flutter analysis, no mode update
or
- i. @ XQT KSDM - update finite element model for new design variables
- j. Use SPAR to generate new modes (processors E, EKS through EIG)
- k. @ XQT GAF
- l. @ XQT FLT - new flutter analysis, after mode update.

(iv) Automated static optimization

- a. @ XQT TAB
- b. @ XQT ELD
- c. @ XQT TOPO
- d. @ XQT DESVAR

e. @ XQT SOL (with IDP = 28, See SOL), (for mathematical programming)
or

e. @ XQT FSDL (with IDP = 28, see FSDL), (for FSD or TFSD resizing)

(v) Automated Flutter Optimization

a. @ XQT TAB

b. @ XQT ELD

c. @ XQT TOPO

d. @ XQT DESVAR

e. @ XQT AERO

f. @ XQT SUBKRN

g. @ XQT FOL (with IDP = 28, see FOL)

1.2 Input Rules

The input rules used for SPAR apply to the PARS processor with one addition that of a namelist type input. The namelist input is very similar to RESET input except that it provides also for array input. The following are namelist input rules. It is assumed that the namelist name is XX.

The first character in card to be read is always ignored. The second column of the first card must be a \$ immediately followed by the namelist name and a blank. If the name is not XX, the input medium is searched for \$XX. The remainder of the first card and following cards may contain any combination of the legal data items which are separated by commas (a comma after the last data group is ignored). The last input record is terminated by a blank followed by \$END.

The forms the data items may take are:

Variable Name = Constant

Subscribed Variable = Constant

Subscripted Variable = Set of Constants

The set is represented by constants separated by commas, or k^* constant where k^* constant may represent k constants (k is an unsigned integer constant).

Array Name = Set of Constants

The set is represented by constants separated by commas, or k^* constant may represent k constants (k is an unsigned integer).

The data items that appear on the input records need not appear in the same order as the corresponding variable or array names in the namelist. All variable or array names of the namelist need not have a corresponding data item in the input records; if none appears, the default values will be used.

Blanks must not be embedded in a constant or a repeat constant field, but may be used freely elsewhere in a data record. The name of an array and the value of its first elements must appear on the same record. The last item on each record that contains data items must be a constant followed by a comma. The comma is optional in the record that contains or precedes the \$END sentinel.

Example:

	Col. 2
First Data Card:	\$NAM1 I(2,3) = 5,7,9,
Second Data Card:	0,10,L(3) = 4.3, A=4,3,
Third Data Card:	8 * 4.3, J=4.2,
Last Data Card:	\$END

If the data is used with the namelist example the following action takes place: the first data card is read and examined to verify that its name is the same as the namelist name. The integer, constant 5 is placed in I(2,3), 7 in I(3,3) and 9 in I(4,3). When the second data card is read, integer 0 is placed in I(5,3), and 10 in I(6,3); the real constant 4.3 is converted to integer 4 and placed in L(3); integers 4 and 3 are converted to real constants and placed in A(1) and A(2), respectively. When the third data card is read, the real constant 4.3 is stored in A(3), A(4), ..., A(10); and the real constant 4.2 is converted to integer 4 and placed in J.

SECTION 2

DESIGN VARIABLE DEFINITION AND INTERFACE WITH FINITE ELEMENT MODEL

Processors DESVAR and KSDM described in this section, establish the relationship between design variables and the finite element model and provide the means of changing that model as the design variables are resized. Defining the design variables is the first task for a PARS user. Sometimes there is a simple one-to-one relationship between design variables and parameters of the finite element model. For example, the design variable may be the thickness of a panel and all the plate finite elements are set equal to the value of that design variable. The relation could be more complicated, for example if design variable linking is used. The PARS program allows for any general linear relationship between design variables and the resized parameters of the finite element model. These resized parameters may be thickness or cross sectional areas of finite elements, nodal temperatures and nodal masses. Some elements in SPAR (such as the beam elements and the fluid elements) cannot be controlled by design variables. Thus, if such elements are used in a model, their structural dimensions remain unchanged during optimization.

The thickness and cross sectional areas of finite elements are controlled in SPAR by section property input (SA, SB, etc.). Consequently in PARS the design variables control section properties and all the elements that have the same section type and number have the same section properties.

The following is a list of the types of elements belonging to SPAR element library which can be resized, and their section type.

<u>Name</u>	<u>Description</u>
E23	Bar elements having only axial stiffness, section type BC.
E31	Triangular membrane, constant-stress element, section type SA.
E32	Triangular bending element, section type, SA.
E33	Triangular membrane + bending element, section type SA.
E41	Quadrilateral membrane, element section type SA.
E42	Quadrilateral bending element, section type SA.
E43	Quadrilateral membrane + bending element, section type SA.
E44	Quadrilateral shear panel, section type SB.

The resized dimension for section type BC is the bar cross sectional area and for section types SA or SB it is the plate thickness.

2.1 DESVAR Design Variable Definition

Function: In a structural optimization problem, it is necessary to define the design variables which control various parameters of the structure.

DESVAR permits the user to relate a structural parameter q as a linear function of the design variables in the following form,

$$q = q_0 + \sum_{i=1}^n v_i F_i \quad (2.1-1)$$

where,

q is a structural parameter.

q_0 is the minimum value of the parameter

v_i is the value of i th design variable (always positive)

F_i is the amount by which a structural parameter can be changed by making a unit change in the value of the i th design variable, i.e., v_i .

n is the number of design variables*

Most of the F_i are zero, the user has to specify only the nonzero ones.

$$\text{Also } \frac{\partial g}{\partial v_i} = F_i \quad (2.1-2)$$

A structural parameter, which is controlled by a design variable can be either

- (i) a tuning mass lumped at a grid point or
- (ii) a thickness or an area of a finite element or
- (iii) a nodal temperature at a grid point.

Parts of the structure may be assigned no design variables and remain unchanged during the optimization process.

Input/Output Data Sets: Input data sets are element and section definition sets generated by TAB and ELD.

Output data sets are governed by the input information relating to the design variable definitions.

The following output data sets are written into library DLIB (see reset controls):

1. MASS DV 0 0, contains data relating design variables and tuning masses, in the following order n_j n_v f which indicates that design variable n_v controls the mass concentrated at joint n_j by a scaling factor f . $N_I = 3$, $N_J =$ number of controlled masses.

*At most five nonzero factors are allowed. When FSD or TFSD resizing is used each section should be controlled by a single design variable.

2. NTDV SEQ tset 0; contains the identity of the design variables that control nodal temperatures. NI = 1, NJ = maximum number of variables that control one nodal temperature.
3. TEMP DERI nset 0, contains the scaling factors which relate design variables and nodal temperatures. NI = maximum number of variables that control one nodal temperature, NJ = number of joints.
4. NSDV sname 0 ID. where sname is a section name (BC, SA or SB) and ID is the section's identification number (11, 13 or 14) contains the number of design variables (max. 5) that control each section. NI = number of sections, NJ = 1
5. KSDV sname 0 ID; contains the identity of the design variable that control each section. NI = number of sections, NJ = minimum number of variables that control a section.
6. FSDV sname 0 ID; contains the scaling factors for each section. NI = number of sections, NJ = maximum number of variables that control a section.
7. DCON BLOCK 0 0 Contains data about element types and design variables.
8. DES1 VARI 0 0 Contains the values of the design variables. NI=1, NJ = number of design variables.
9. GAGE VARI 0 0 contains the reference values of the design variable (often corresponds to the minimum structural parameter controlled by the design variable). NI = 1, NJ = number of design variables.
10. DVMX VARI 0 0 contains the maximum values of the design variables NI = 1, NJ = number of design variables.

Reset Controls:

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
DLIB	1	Destination Library for design variable definition data sets
SET	999	The set number associated with design variables controlling nodal temperatures
BLDV	5000	Block length reserved for generating design variable definition data sets.
BLIB	1	Library containing data sets generated by TAB

Execution Sequence:

The following is the sequence of data cards

@XQT DESV

RESET Cards

Design variable definition cards

Design Variable Definition:

The following is the sequence of the design variable definition data cards.

(a) MASS: $n_{j_1}, n_{v_1}, f_1: n_{j_2}, n_{v_2}, f_2: n_{j_3}, n_{v_3}, f_3: \dots : \dots \$$

The first record identifies tuning MASS parameters. The subsequent records form triads that are separated by a terminator mark (: on UNIVAC; on CDC). A triad n_j, n_v, f indicates that the design variable n_v controls the mass concentrated at node n_j by a scaling factor f .

Example: MASS: 51, 2, 3.25: 14, 4, 1.15 \$

25, 3, 2.15: 17, 2, 1.75 \$.

The above data indicates that design variable 2 controls two masses m_{51} and m_{17} lumped at nodes 51 and 17 respectively; design variable 3 controls the mass m_{25} ; design variable 4 controls the mass m_{14} . Also,

$$\frac{\partial m_{51}}{\partial v_2} = 3.25, \quad \frac{\partial m_{17}}{\partial v_2} = 1.75, \quad \frac{\partial m_{14}}{\partial v_4} = 1.15, \quad \text{and} \quad \frac{\partial m_{25}}{\partial v_3} = 2.15$$

(b) SECTNAME: nv_1 $isec_1$, $jsec_1$, f_1
 nv_2 , $isec_2$, $jsec_2$, f_2 : nv_3 , $isec_3$, $jsec_3$, f_3
 nv_4 , $isec_4$, $jsec_4$, f_4 :

The first record in each set of cards contains the section type. The subsequent records form quartets each separated by a terminator mark (:on UNIVAC; on CDC). A quartet nv , $isec$, $jsec$, f indicates that the thickness or area of elements controlled by sections $isec$ to $jsec$ (assigned in loop limit format, $jsec$ can be omitted if it is equal to $isec$) is controlled by design variable nv with a scaling factor f . Note that sections that are not resized should have higher numbers than sections which are resized.

Example 1

SA: 3, 2, 5, 1.5: 2, 1, 1.7 \$

The above data indicates that the thickness t_3 of sections 2 to 5 of type SA is controlled by design variable number 3. Also the thickness, t_2 of section 1 of the same section type is controlled by design variable number 2. Also

$$\frac{\partial t_3}{\partial v_3} = 1.5 \quad \text{and} \quad \frac{\partial t_2}{\partial v_2} = 1.7$$

(c) TEMPERATURES: nj_1 , mj_1 , nv_1 , f_1 \$
 nj_2 , mj_2 , nv_2 , f_2 : nj_3 , mj_3 , nv_2 , f_3 :----- \$

The first record identifies nodal temperature parameters. A quartet nj , mj , nv , f indicates that the nodal temperatures specified at the joints nj to mj (assigned in loop limit format) are controlled by design

variable nv with a scaling factor f .

Example

TEMPERATURES \$

5, 7, 7, 1.175: 46, 46, 8, 5.25: 50, 50, 7, 3.25 \$

33, 33, 7, 1.00 \$

The above data indicates that design variable 7 controls the temperatures T_5 , T_6 , T_7 , T_{50} and T_{33} at joints 5 to 7, 50 and 33 respectively; design variable 8 controls the temperature T_{46} with a factor 5.25.

Also,

$$\frac{\partial T_{46}}{\partial v_8} = 5.25, \frac{\partial T_{50}}{\partial v_7} = 3.25, \frac{\partial T_{33}}{\partial v_7} = 1.0 \text{ and}$$

$$\frac{\partial T_j}{\partial v_7} = 1.175, \text{ for } j = 5, 7.$$

(d) DESIGN: $nv_1, mv_1, v_1: nv_2, mv_2, v_2$ \$

----- : nv_k, mv_k, v_k \$

This specifies the values of the design variables. A triad nv, mv, v indicates that design variables nv to mv (assigned in a loop limit format) have the values v .

Example

DESIGN \$

1, 4, 1.76: 5, 7, 2.58: 8, 50, 1.69 \$

The above indicates that design variables 1 to 4 are equal to 1.76; design variables 5 to 7 are equal to 2.58 and design variables 8 to 50 are each equal to 1.69.

(e) GAGE: $nv_1, mv_1, g_1 : nv_2, mv_2, g_2 : \text{----- } \$$

The GAGE card specifies the reference values v_{mi} for the design variables (see discussion in section 5.1). It is recommended that the reference values be set to the minimum gage of the structural parameters controlled by the design variables. A triad nv, mv, g indicates that design variables nv to mv (assigned in a loop limit format) have the reference value equal to g .

Example

GAGE \$

1, 15, 0.05: 16, 50, .01 \$

The reference values for the design variables 1 to 15 are .05 and for design variables 16 to 50 are equal to .01.

If one or more of entries (a) to (c) is not present in the structural system then the corresponding input cards can be omitted. For example: when thermal constraints are not considered in design procedures, there is no need to specify TEMPERATURE design variables.

(f) DMASS: $nv_1, mv_1, dm_1 : nv_2, mv_2, dm_2 : \text{----}$

The DMASS card specifies the base values for the mass derivatives, which may account for mass contributions such as secondary structure which are not included in the finite element model but are proportional to the design variables. A triad nv, mv, dm indicates that design variables nv to mv have the base mass dm .

Example

DMASS \$

1, 15, 0.001: 16, 16, .005 \$

The base mass derivative values to design variables 1 to 15 are .001 and for design variable 16 is .005.

(g) DVMX: $nv_1, mv_1, vm_1: nv_2, mv_2, vm_2: \dots$

The DVMX card specifies upper bounds on design variables. A triad nv, mv, vm indicates that design variables nv to mv have the maximum value vm . This data is needed only if upper limits on design variables are desired. Note that data items (d), (e), (f), and (g) can be skipped and AUS used instead to create the corresponding data sets.

2.2 KSDM: Structural parameter update Function: KSDM performs one or more of the following functions depending upon the values of the reset parameters (see RESET controls)

1. Updates the values of structural dimensions (areas for one dimensional elements, thicknesses for 2-dimensional elements) nodal temperatures and tuning masses which are controlled by design variables (see processor DESVAR). The dependence of a structural parameter q on the design variables is

$$q = q_0 + \sum_{i=1}^n v_i F_i \quad (2.2-1)$$

where q_0 : is the minimum value of the parameter q .

v_i : is the i th design variable (> 0 .)

n : is number of design variables

F_i : is the amount by which a structural parameter q is changed due to a unit change in the value of i th design variable, v_i .

The user has two options for defining the minimum values q_0 of structural parameters which are cross sectional properties (such as thickness or areas).

The first option is to use the structure defined by SPAR processor TAB as the minimum gage structure. This option is selected by setting the

reset parameter MING to -1 the first time KSDM is executed. The other option is to use the structure defined by TAB as the initial design and to have KSDM calculate the minimum gage values from Eq. (2.2-1) using the initial values of the design variables. This option is selected by setting MING = 0.

2. To generate the minimum-gage values q_0 of the thicknesses t_0 and areas A_0 for individual section of a given section type, say SA. This calculation is made when the structure defined by SPAR is the initial design (reset control MING = 0).

3. To evaluate the derivatives of mass with respect to design variables. The derivatives of the mass is the sum of base values supplied by the user in processor DESVAR and the derivatives of the structural mass. The latter (derivatives of structural mass) are generated internally in KSDM. The base values of the mass derivatives may be used to account for mass contributions such as secondary structure which are not included in the finite element model but are proportional to the design variables. For the design variables which control nodal temperatures, the base mass derivatives may represent the mass penalty (e.g., insulation) associated with temperature change. It can also be used to represent the cost penalty associated with installation and maintenance of any cooling system device to reduce temperature. In the latter case equivalent mass penalty is used for the base mass derivatives.

Input Data Sets:

1. The data set DESI VARI 0 0 (NI = 1, NJ = NDV) where NDV is the number of design variables contains the value of the design variables on library DLIB.

2. The data set DMAS DV 0 0(NJ=1, NI=NDV) or library DLIB contains the base mass derivative values (if any) which may account for mass contributions such as secondary structure which are not included in the finite element model but are proportional to the design variables.

The above two data sets have to be generated by DESVAR or AUS the first time KSDM is executed. DESI VARI data set is updated by NEWTON or SCON.

3. Data sets generated by TAB on library BLIB.
4. The element packets from library ELIB.
5. Data sets generated by DESVAR on library DLIB.
6. Nodal temperature (if any) data set generated by AUS on library TLIB.

Output Data Sets/generated on library DLIB

1. CDIM namesect 0 ID (NJ=1, NI=NSECT) contains current structural parameters values.
2. GDIM namesect 0 ID (NJ=1, NI=NSECT) contains minimum gage structural parameters values.
3. ISCT element name 0 ID (NJ, NI=NELEMT) contains the corresponding section number of each element.
4. INGS elemt name 0 0 (NJ=5, NI=NDV) contains the elements which are controlled by each design variable and by what factor.
5. MASS DER 0 0 (NI = NDV, NI = 1) contains the derivatives of the mass of the structure with respect to the design variables.
6. MING BW 0 0 (NI = 1, NJ = 1) contains the minimum gage structural weight.

Reset Controls:

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
BLIB	1	Library containing TAB generated data sets.
ELIB	1	Library containing the element packets.
GAZERO	1.E-20	Zero-test parameter, (beam area) x (shear modulus).
CLZERO	1.E-20	Zero-test parameter, beam non-uniform torsion constant.
TLIB	1	Source library for nodal temperatures.
BLOC	10	Maximum number of blocks in INGS data set. Should be set to zero to save core if FSD or TFSD resizing is used.
DLIB	1	Source library for design data sets created by DESVAR and destination library for data sets created here.
MING	1	<p>MING = -1: The original structure generated by TAB) is taken as the minimum-gage structure.</p> <p>MING = 0: The minimum gage thicknesses are determined using Eq. (2.2-1)</p> <p>MING > 0: It is assumed that minimum gage values for structural parameters are already generated in the previous executions).</p>

Reset Controls (Continued):

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
IND	0	IND=0; when KSDM is executed alone, IND=1; when KSDM is in an optimization loop.
ITDV	-1	ITDV=0 indicates that temperature design variable are used. This reset control is not required when KSDM is executed as part of SOL or FSDL
MASS	0	MASS=0 indicates that no tuning mass design variables are used. MASS≠0 indicates that tuning mass design variables are used.
G	1.	Gravitational constant. Used when mass properties are desired while the TAB data included weight densities.

SECTION 3

STATIC CONSTRAINTS AND THEIR DERIVATIVES

The following sections describe PARS processors that calculate derivatives of displacements, stresses, and displacement and stress constraints. Processors DGRAD and TGRAD calculate the derivatives of the displacements with respect to design variables that control structural dimensions and nodal temperatures, respectively. Processor SGRAD calculates derivatives of the element stresses. Processors DCON and SCON are used to define displacement and stress constraints and calculate the derivatives of these constraints with respect to design variables.

3.1 DKS: Derivative of K Matrices

Function: DKS Computes the derivatives of K matrix with respect to design variables for plate bending elements. This is needed for computing the analytical derivatives of the displacement vectors.

Input data sets:

Input data sets are EFIL and design variable definition data sets generated by processors E, EKS and DESVAR respectively.

Output Data Set is

DERK EXY ltype nnodes: This data set contains the derivatives of the stiffness matrix for each element in the intrinsic form. It is stored in the following form

DERK (NDOF, NEL, NDV)

where NDOF is the number of intrinsic degrees of freedom

NEL is the number of elements

NDV is the number of design variables

Reset Controls:

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
BLIB	1	Library containing TAB generated data sets
ELIB	1	Library containing the element packets
KSLI	1	Library containing the output data sets from this processor
DVLI	1	Source library for design variable data sets

3.2 DGRAD - Gradient of Displacement Generator

Function: DGRAD is used to calculate the derivative of displacement vectors with respect to design variables. The design variable is a structural dimension which controls the thickness or area of one or several finite elements of one or more types.

In calculating the derivatives of the displacements the dependence of thermal or dislocation loads on the design variables is accounted for while the dependence of inertia loads on the design variable is not accounted for.

Input/Output Data Sets:

Input data sets are usually the joint displacement data sets resident in QLIB as STAT DISP iset ncon or SOURCE=N1, N2, n₁, n₂.

The SOURCE statement names a data set resident in QLIB which replaces STAT DISP iset ncon as the source of joint motion data.

The Output data set is: - GRAD DISP iset ncon; where iset, ncon have conventional SPAR meanings. It is stored in library GDLIB as an array of the following dimensions:

GRAD (JDF * JT, NDV, NCASE)

where

JDF: is the number of active degrees of freedom.

JT: is the total number of joints.

NDV: is the number of design variables.

NCASE: is the number of load cases in the specified iset. (The third entry for the data set name). The GRAD DISP iset ncon data set is generated for icode=icase1, icase1+1,....,icase2.

Reset Controls:

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
ELIB	1	Source library for the element packets.
QLIB	1	Source library for STAT DISP iset ncon or the SOURCE data set supplied alternatively.
SET	1	Load Set (iset).
DVLI	1	Library for design variable data sets (DESVAR)
SILI	1	Library for initial strain data sets.
L1	0	icase1 (if zero, the first load case number used in STAT DISP data set, if present, is assumed).
L2	0	icase2 (if zero, the last load case number used in STAT DISP data set, if present, is assumed).

Reset Controls (Continued):

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
CON	0	4th word of STAT DISP iset ncon.
IEA	0	Read error control parameter: IEA=0: Both LIB and Fin errors are displayed, and run aborted. IEA=1: LIB error is displayed only. IEA=2: Neither LIB nor FIN error are displayed.
GDLIB	1	Destination library for the gradient of displacement data set GRAD DISP iset ncon.
KILIB	1	Library containing factored stiffness matrix.
K	K	Name of the stiffness matrix.

Execution sequence:

The following is the sequence of data cards:

@XQT DGRA

RESET Cards

SOURCE=N1, N2, n1, n2 (trailing words mask - filled if omitted).

Note that n1 will override the SET reset control.

3.3 TGRAD Derivatives of Displacement Vectors with respect to Design Variables Controlling Nodal Temperature

Function: The processor TGRAD computes the derivatives of displacement vector with respect to design variables which control the nodal temperatures.

Temperatures specified at several joints can be controlled by the same design variable or conversely a single joint temperature can be controlled by more than one design variable. Basic definitions and input information for the temperature design variables are provided in the section describing processor DESVAR.

A nodal temperature T_j at the joint j is related to the design variables by

$$T_j = T_{o_j} + \sum_{i=1}^{i=n} v_i * F_{ij} ; j = 1, m$$

where

T_j : nodal temperature at joint j

T_{o_j} : reference temperature at joint j .

This usually corresponds to the minimum temperature at the node j .

v_i : Value of the i th design variable (always positive)

F_{ij} : the amount by which the temperature T_j can be changed by making a unit change in the value of the i th design variable. That is

$$\frac{\partial T_j}{\partial v_i} = F_{ij}$$

For those nodes which are not controlled by temperature design variables

$$F_{ij} = 0$$

n : the number of design variables.

m : the number of nodes.

Output Data Sets:

Output data sets are:

1. EQNF FORC tset iblock: This data set contains the derivatives of equivalent nodal force generated due to the presence of thermal loadings, with respect to design variables controlling T_j .

2. IS EXY tset iblock: This data set contains the derivatives of strain produced due to the presence of thermal loadings with respect to design variables controlling T_j .

3. GRAD DISP iset icon: This data set is originally created by processor DGRAD. TGRAD fills in empty block spaces of this data set with the derivatives of displacement vectors with respect to design variables controlling nodal temperatures. They are stored as

GRAD (JDF, JT, NDV, NCASE)

where

JDF: is the number of unconstrained degrees of freedom.

JT: is the number of joints.

NDV: is the number of design variables.

NCASE: is the total number of load cases in a given load set (iset).

Input Data Sets:

The following two input data sets generated in processor DESVAR are used by TGRAD

1. TEMP DERI tset 0. Stores the derivatives of nodal temperature vector $[T] = [T_j]$ of dimension JT with respect to design variables, JT being the number of joints.

2. NTDV SEQ tset 0: Contains the design variable numbers associated with the design parameter controlling the nodal temperatures.

Reset Controls:

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
SET	1	Load set number.
TSET	999	Set number associated with derivatives of equivalent nodal force data and derivative of initial strain data. Do not use same number for RESET parameters SET and TSET.
L1	0	First load case number. In default it takes the lowest case number previously generated.
L2	0	Last load case number. In default it takes the highest load case number previously generated.
ELIB	1	Library containing element packets.
INLI	1	Source library for all applied load data (e.g., nodal or element temperatures etc.).
FEFLIB	1	Destination library for derivatives of equivalent nodal force data.
ISLIB	1	Destination library for derivatives of initial strain data.
ISBL	896	Block length for generating the derivatives of initial strain data.
CON	1	Constraint case (see discussion for processor INV)

Reset Controls: (Continued)

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
GDLIB	1	Destination library for derivatives of displacement vectors data.
K	K	Name of stiffness matrix.
KILIB	1	Library containing factored stiffness matrix.
DVLI	1	Library containing data sets created by DESVAR.

3.4 SGRAD Stress Gradient Generator

Function: SGRAD evaluates the derivatives of element stresses at the centroid of each element with respect to the design variables.

Input Data Sets:

1. GRAD DISP iset icon: Contains the derivatives of displacement vector with respect to design variables.
2. IS EXY tset mask: Contains the derivatives of initial strains with respect to design variables controlling nodal temperatures.
3. NTDV SEQ tset mask: Stores design variable numbers controlling nodal temperature.

Data sets 2 and 3 are required if temperature design variables are present.

Output Data Sets:

1. GOST EXY iset icase: This data set contains the derivatives of the stress tensor (at the centroid of each element) with respect to design variables in the following form:

GOST (LSFE, NEL, NDV)

where,

LSFE: is the number of stress components generated per element.

NEL: is the number of the elements in a given element type EXY.

NDV: is the number of the design variables.

Reset Controls:

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
ELIB	1	Library containing the element packets.
QLIB	1	Library containing the derivatives of initial strain data sets.
L1	0	First load case number. In default it assumes the lowest load case number for which derivatives of displacement vectors are generated.
L2	0	Last load case number. In default it assumes the highest load case number for which the derivatives of displacement vector are generated.
SET	1	Load set number.
CON	MASK	Constraint case number (see INV discussion).
LREC	5600	Block length required for generating the derivatives of stress data set.
INST	25	The source library for INST EXY mask mask data set.
GDLI	1	Source library for derivatives of displacement data.

Reset Controls: (Continued)

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
GSLI	1	Destination library for derivatives of stress data.
DVLI	1	Library containing data sets generated by processor.
TSET	999	Load set associated with temperature design variables (see DESVAR for discussion).

Execution Sequence

@XQT SGRA

Reset cards (if any)

SOURCE = N₁ N₂ n₁ n₂ (trailing words MASK - filled if omitted)

The SOURCE statement names a data set, resident in QLIB which replaces GRAD DISP iset icon, as the source of displacement gradient data.

Subsequent control cards designate specific groups of elements for which gradient of stress data with respect to design variables is to be computed, as illustrated below:

E21: 4,7: 10\$ E21 groups 4 through 7, and group 10.

E33\$ All E33 elements

E43: 1:7:10, 12\$ E43, groups. 1,7,10,11,12.

E44: 1,5:7,17\$ etc.

If the control cards above are not given, derivatives of stresses will be computed for all elements which are controlled by design variables.

3.5 DCON: Displacement Ratio Generator

Function: DCON determines the displacement ratios and the derivatives of these ratios at the nodal points of the structure where maximum displacements are prescribed. The displacement ratio is the absolute value of the ratio of a computed displacement component at a joint to its allowable value.

If Q_i is the i th displacement ratio, the displacement constraints are formulated in the following manner:

$$g_i = 1 - Q_i \geq 0; i = 1, \text{NDC} \quad (3.5.1)$$

NDC is the number of displacement constraints.

Input data sets are:

1. STAT DISP iset icon: Contains the displacement vectors stored in SYSVEC format. It consists of n blocks, where n is the number of cases in the designated load set. Block 1 is the solution corresponding to case 1, block 2 corresponds to case 2, etc.
2. GRAD DISP iset icon: Contains the derivatives of displacement vectors with respect to design variables. All the NDV blocks of case 1 are stored first, NDV blocks of case 2 are stored next and so on, where NDV is the number of design variables.

Output data sets are:

3. DISP CON 0 0: Contains the allowable values of displacements.
4. EADS CON iset icase: Contains the absolute value of the ratio of the computed displacement component at a joint to its maximum allowable value. This data set is a vector of dimension

NDC where NDC is the maximum number of displacement constraints.

5. GGI CON set coid - contains the constraint functions given by Eq. 3.5.1 for those constraints for which $g_i \leq GLIM$ (see reset controls) (i.e. most critical constraints)
6. DGI con set coid - contains the derivatives of the constraints included in the data set GGI CON. Each block in this data set contains the derivatives of a single displacement constraint with respect to all design variables.

Reset Controls:

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
ELIB	1	Library containing the element information packets.
QLIB	1	Library containing the displacement data sets. Also destination library for EADS CON data set
GDLIB	1	Library containing the derivatives of displacement data. Also destination library for GGI CON and DGI CON data sets.
SET	1	Load set number.
CON	1	Constraint case number (see discussion for INV processor)
L1	1	First load case number.

Reset Controls (continued):

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
I2	0	Last load case number. In default it assumes the maximum number of load cases for which STAT DISP data set exists.
DISC	25	Destination or source library for DISP CON mask mask data set.
GLIM	1	Critical constraint cutoff point. Only constraints that satisfy $g_i \leq \text{GLIM}$ are included in the data set GGI CON.
COID	1	Constraint identification number. Used as the last name of data sets GGI CON and DGI CON. Should be different than any constraint identification number for stress, flutter or any other constraints generated by other processors.

Execution Sequence:

The following is the sequences of data cards.

@XQT DCON

Reset cards (if any)

DISCON: $n_1, j_1, \text{adv}_1: n_2, j_2, \text{adv}_2: n_3, j_3, \text{adv}_3$ \$:---: n_k, j_k, adv_k \$

The DISCON card specifies the allowable values for the nodal displacements.

The entries in between the two record delimiters (:on UNIVAC) indicate

that nodal displacement at joint n in direction j has an absolute value less than or equal to adv .

Example

DISCON: 35, 3, 0.57: 35, 1, 0.05\$

25, 3, 0.45\$

The allowable values for w_{35} and u_{35} at joint 35 are 0.57 and 0.05 units respectively. The allowable value for w_{25} is 0.45 unit.

3.6 SCON: Stress ratio generator

FUNCTION: SCON determines the equivalent stress ratios and either generates their derivatives with respect to the design variables when mathematical programming optimization is used, or generates resize factors when FSD (Fully Stressed Design) or TFSD (Thermal Fully Stressed Design [4]) is applied. A design variable can control the magnitude of nodal temperatures or the structural dimensions of the finite elements.

The equivalent stress ratio for the different element types are calculated as follows:

(a) Axial bar element E23:

$$SR = \frac{\sigma_1}{S_1} \quad (3.6-1)$$

σ_1 is the calculated stress in the element which can be either compressive or tensile, and

S_1 is its appropriate allowable tensile XT or compressive XC depending on the sign of σ_1 .

(b) For two dimensional elements E31 E32 E33

E41 E42 E43

the Tsai Wu theory [5] is used, with the square root of the failure criterion used as a stress ratio,

$$SR = \left\{ \left(\frac{1}{XT} + \frac{1}{XC} \right) \sigma_1 + \left(\frac{1}{YT} + \frac{1}{YC} \right) \sigma_2 - \frac{(\sigma_1)^2}{XT \cdot XC} - \frac{(\sigma_2)^2}{YT \cdot YC} + \left(\frac{\sigma_{12}}{S_{12}} \right)^2 + 2 F_{12} \sigma_1 \sigma_2 \right\}^{1/2} \quad (3.6-2)$$

where σ_1 , σ_2 , σ_{12} are the computed values for the normal and shear stresses evaluated at the centroid of the element.

XT and XC are the tensile and compressive allowables in the X direction and YT and YC are the allowables in the Y direction. S_{12} is the allowable shear stress. The compressive allowables (XC & YC) have negative values. For isotropic elements $F_{12} = \frac{1}{2 XT \cdot XC}$ while for anisotropic elements F_{12} is input by the user. For composite materials it is recommended to use $F_{12} = 0$ [6].

(c) Shear panel element E44:

$$SR = \frac{\sigma_{12}}{S_{12}} \quad (3.6-3)$$

σ_{12} is the computed shear stress in the element and S_{12} is its allowable. For stress constraints the constant function is defined as

$$g = 1 - SR \geq 0 \quad (3.6-4)$$

Fully Stressed Design (FSD) is applicable for the following elements:

- (a) Axial bar elements E23
- (b) Two dimensional membrane elements E31, E41
- (c) Shear panel elements E44

For the bar and shear panel elements the stress ratios are the resize factors, but for the membrane elements the resize factors are calculated as follows:

$$B_i = \left(\frac{1}{X_T^i} + \frac{1}{X_C^i}\right) \sigma_1^i + \left(\frac{1}{Y_T^i} + \frac{1}{Y_C^i}\right) \sigma_2^i \quad (3.6-5)$$

$$C_i = \frac{-(\sigma_1^i)^2}{X_T^i \cdot X_C^i} - \frac{(\sigma_2^i)^2}{Y_T^i \cdot Y_C^i} + \frac{(\sigma_{12}^i)^2}{S_{12}^i} + 2F_{12}^i \sigma_1^i \cdot \sigma_2^i \quad (3.6-6)$$

Resize factor for the i-th elements is

$$RF_i = \frac{1}{2} (B_i + (B_i^2 + 4C_i)^{1/2}) \quad (3.6-7)$$

When TFSD is applied the resize factors are calculated as follows:

(a) Axial bar elements E23:

$$RF = \frac{\sigma_M}{S_1 - \sigma_t} \quad (3.6-8)$$

where σ_M is the mechanical stress in the element, σ_t is the thermal stress and S_1 is either the tensile or compressive allowable depending on the sign of σ_M .

(b) Two dimensional membrane elements E31

$$BT = \left(\frac{1}{X_T} + \frac{1}{X_C}\right) \sigma_{1t} + \left(\frac{1}{Y_T} + \frac{1}{Y_C}\right) \sigma_{2t} \quad (3.6-9)$$

$$BM = \left(\frac{1}{X_T} + \frac{1}{X_C}\right) \sigma_{1M} + \left(\frac{1}{Y_T} + \frac{1}{Y_C}\right) \sigma_{2M} \quad (3.6-10)$$

$$\begin{aligned}
CT &= \frac{-(\sigma_{1t})^2}{XT \cdot XC} - \frac{(\sigma_{2t})^2}{YT \cdot YC} + \frac{\sigma_{12t}^2}{S_{12}} + \\
&+ 2 \cdot F_{12} \cdot \sigma_{1t} \cdot \sigma_{2t}
\end{aligned} \tag{3.6-11}$$

$$\begin{aligned}
CM &= -\frac{(\sigma_{1M})^2}{XT \cdot XC} - \frac{(\sigma_{2M})^2}{YT \cdot YC} + \frac{\sigma_{12M}^2}{S_{12}} + \\
&+ 2F_{12}\sigma_{1M} \cdot \sigma_{2M}
\end{aligned} \tag{3.6-12}$$

$$\begin{aligned}
DMT &= -\frac{2\sigma_{1t}\sigma_{1M}}{XT \cdot XC} - \frac{2\sigma_{2t} \cdot \sigma_{2M}}{YT \cdot YC} + \frac{2\sigma_{12t} \cdot \sigma_{12M}}{(S_{12})^2} + \\
&+ 2F_{12}(\sigma_{1M} \cdot \sigma_{2t} + \sigma_{2M} \cdot \sigma_{1t})
\end{aligned} \tag{3.6-13}$$

$$\Delta = (BM + DMT)^2 - 4 \cdot CM (BT + CT - 1) \tag{3.6-14}$$

$$RF_{1,2} = \frac{1}{2}[-(BM + DMT) \pm (\Delta)^{1/2}]/2(BT + CT - 1) \tag{3.6-15}$$

and the resize factor is the minimum nonnegative root. If both RF_1 & RF_2 are negative, the program sets RF to 2.

If the denominator of Eq. 15 is zero,

$$RF = \frac{-CM}{BM + DMT} \tag{3.6-16}$$

in Eqs. 3.6-9 to 3.6-13 subscript t and M are for thermal and mechanical stress respectively.

(c) Shear panel elements E44:

$$RF = \frac{\sigma_{12M}}{S_{12} - \sigma_{12t}} \quad (3.6-17)$$

where σ_{12M} & σ_{12t} are the mechanical and thermal shear stress in the element.

When one of the resize factors in the bar or shear panel elements is negative, the stress ratio is used for resizing this element.

Input Data Sets:

1. STRS EXY iset icase: Generated by GSF in library QLIB and contains the stresses and stress resultants.
2. GOST EXY iset icase: Generated by DGRA in library GSLIB & contains the derivatives of the stresses. This data set is read when derivatives of stress ratios are required.
3. Secname BTAB MASK ID: Generated by TAB in library BLIB and contains section properties.
4. ELTS ISCT 0 0: Generated by TAB in library BLIB and contains section identification number of all the element types in the problem.
5. ISCT EXY 0 ID: Generated by KSDM in library DLIB and contains the section numbers that control the elements.
6. LOAD ALST 0 0: To be generated by the user (by AUS) in library ALST when there is more than one set of allowable stresses. NI = number of load cases, NJ = 1. Contains the allowable set number corresponding to each load case. When FSD or TFSD resizing is applied, SCOR reads also the following data sets:

7. KSDV Secname 0 ID: Generated by DESVAR in library DLIB and contains the design variable numbers which control every section. FSD and TFSD assume that every section is controlled by a single design variable.
8. FSDV Secname 0 ID: Generated by DESVAR in library DLIB and contains the corresponding factors of the design variables, see DESVAR for details.
9. DESI VARI 0 0: Generated by DESVAR or AUS in library DLIB and contains the current values of the design variables.
10. CDIM Secname 0 ID: Generated by KSDM in library DLIB and contains the current structural dimensions of the sections.
11. GDIM Secname 0 ID: Generated by KSDM in library DLIB and contains the minimum gage values of the structural dimensions.
12. MASS DER 0 0: Generated by KSDM in library DLIB and contains mass derivatives of the structural mass with respect to the design variables.
13. MING BW 0 0: Generated by KSDM in library DLIB and contain the minimum gage structural weight.

To apply TFSD rather than FSD the user has to generate the data set MCTR LCAS 0 0 data set in library QLIB by using AUS. (NI = number of load cases, NJ = 1). Each thermal and mechanical case should have a different load case number and this data set contains for each mechanical load case the corresponding thermal load case.

-1 corresponds to purely mechanical load case

0 correspond to thermal load case.

Output Data Sets:

1. EQST EXY iset icase: Generated in library QLIB and contains the magnitude of the equivalent stress ratio defined earlier at the centroid of each element in a given element type EXY. For the two dimensional elements this data set contains the six coefficients of Tsai Wu theory (Eq. 3.6-2).
2. If the allowable stresses are given as input in the input card stream ALST Secname set icase data set is generated as output in library ALST.
3. RESZ EXY iset icase: Generated in library RLIB when FSD or TFSD is applied and contains resize factors of the elements.
4. GGI CON set idcl: Generated into library QLIB and contains the critical constraints ($g \leq GLIM$) for each load case. The last name of the data set is equal to the load case number plus the reset control COID. This data set is generated only if LFSD = 0.
5. DGI CON set idcl: Generated into library GSLIB and contains the derivatives of the critical constraints.

When FSD or TFSD is applied data set DES1 VAR1 0 0 is updated according to the resize factors or stress ratios calculated by SCON.

Reset Controls:

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
ELIB	1	Library containing element information packets.
QLIB	1	Source library for displacement and stress data sets.

Reset Controls (Continued):

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
L1	0	The first load case number. In default it assumes the lowest load case number available for displacement data set.
L2	0	The last load case number. In default it assumes the highest load case for which displacement data are generated earlier.
SET	1	Load set number.
ALST	1	Library containing the allowable stress data.
CON	1	Constraint case number.
GSLIB	1	Source library for derivatives of stress data set.
ELIB	1	Source library for TAB and ELD generated data.
DLIB	1	Library containing the data sets created by DESVAR and KSDM.
LFSD	0	= 0 calculate stresses constraints and their derivatives = 1 calculate only stress ratios = 2 calculate stress ratios and update design variables using FSD or TFSD*

*TFSD is used when the data set MCTR LCAS 0 0 exists in library QLIB.

Reset Controls (Continued):

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
RLIB	25	Library containing resize factors calculated by FSD or TFSD
GLIM	1	Critical constraint cutoff point only constraints satisfying $g \leq \text{GLIM}$ are stored in data set GGI CON
COID	1	Constraint set identification. Last name (N4) of data set GGI CON set idcl is equal to COID plus load case number. No other constraint set may have same last name.

Execution Sequence:

The following is the sequence of data cards

@XQT SCON

Reset Cards (if any)

Stress allowables input as follows:

STRESS

Section type⁽¹⁾, Set number⁽¹⁾

$ns_1, ms_1, XT^{(1)}, XC^{(1)}, S_{12}^{(1)}, YT^{(1)}, YC^{(1)}$

$ns_2, ms_2, XT^{(2)}, XC^{(2)}, S_{12}^{(2)}, YT^{(2)}, YC^{(2)}$

.....

$ns_k, ms_k, XT^{(k)}, XC^{(k)}, S_{12}^{(k)}, YT^{(k)}, YC^{(k)}$

Section type⁽¹⁾, Set number⁽²⁾

.....

Section type⁽²⁾, Set number⁽¹⁾

.....

If there is only one set of allowable stresses, the set number is not required. But for solving a problem with more than one set, a data set LOAD ALST 0 0 should be generated by AUS in library ALST, (see Input Data Sets section).

The meaning of the line:

ns, ms, XT, XC, S₁₂, YT, YC

is that sections ns to ms have the following allowable stresses:

XT Allowable tensile stress in X direction

XC Allowable compressive stress (negative value) in X direction

S₁₂ Allowable shear stress

YT Allowable tensile stress in Y direction

YC Allowable compressive stress (negative value) in Y direction

YT & YC are used only for the two dimensional elements E31 E32 E33

E41 E42 E43.

If the allowable stresses correspond to a single section ms is to be omitted.

SECTION 4

FLUTTER ANALYSIS AND DERIVATIVES

Five processors AERO, SUBKRN, GAF, DKM and FLT are used to calculate of aerodynamic data, to perform flutter analyses and for obtaining derivatives of flutter constraints.

Processors AERO, SUBKRN and GAF define the aerodynamic parameters and produce the generalized aerodynamic forces. They should always be executed in the order shown above. FLT performs the flutter analysis (IOPT=0 or IOPT=1) and (if IOPT=2) also calculates the derivatives of the flutter constraints with respect to the design variables. The processor DKM, which calculates derivatives of the stiffness and mass matrices with respect to the design variables, is required only if derivatives of flutter constraints are to be calculated in FLT.

Two typical sequences in which these processors are used are given below.

a. Flutter analysis

TAB

.

.

.

EIG

} SPAR processors needed to obtain vibration modes

AERO

SUBKRN

GAF

FLT

b. Flutter analysis and one resizing iteration

TAB

.

.

.

EIG

} SPAR processors needed to obtain vibration modes

DESVAR	define design variables
KSDM	define minimum gages (with RESET parameter MING = 0)
AERO	
SUBKRN	
GAF	
DKM	
FLT	
NEWTON	calculate search direction and new values of design variables
KSDM	calculate structural sizes corresponding to new design variables

The aerodynamic processors AERO, SUBKRN and GAF and the flutter processor FLT are based to a large extent on the WIDOWAC program (Ref. 2, 7).

4.1 AERO - AERODYNAMIC INPUT PROCESSOR

Function: AERO reads and processes the planform and aerodynamic grid data associated with a wing structure. The wing is assumed to be in the x, y plane, the chordwise coordinate is x and the spanwise coordinate is y.

The aerodynamic grid consists of integration points used for calculating the generalized aerodynamic forces (GAF) and collocation points used for solving the pressure-downwash integral equation (see Ref. 1). These points lie along chordwise lines whose number (NYS) and location are specified by the user. All the chords are used for integration but only a subset is used for collocation. The user has to specify the number (NYC) and identity of the collocation chords and the number of collocation points per chord (NS). Usually, both NS and NYC are either four or six. When optimization is to be performed

it is recommended that the effect of these numbers (NS and NYC) on the flutter results be checked and the lowest values which yield acceptable accuracy be chosen because the optimization time is quite sensitive to these numbers.

The vibration modes which are used in the calculation of the generalized aerodynamic forces are available at the joints of the finite element structural model. A two dimensional spline is used to interpolate between the finite element nodes and the collocation and integration points. The user specifies the list of structural joints that are to be used to construct the spline for each chord. One of the considerations in choosing chord locations for the aerodynamic grid is the proximity of enough structural joints so that interpolation errors are minimized. If all the structural nodes lie on one chord the processor artificially doubles their number by constructing an image set at an adjacent spanwise station. For this reason NXP (see RESET controls) must be doubled in this case.

Reset Controls

<u>Name</u>	<u>Default Value</u>	<u>Meaning</u>
ALIB	1	Destination library for aerodynamic data.
BLIB	1	Library containing the structure definition sets generated by TAB
NL	0	Number of leading edge cranks* (Maximum - eight)
NT	0	Number of trailing edge cranks* (Maximum - eight)

*See discussion of namelist WNGEOM

<u>Name</u>	<u>Default Value</u>	<u>Meaning</u>
NYS	7	Number of input spanwise stations+
NYC	4	Number of spanwise stations used for collocation (less than or equal to NYS)+
NS	4	Number of collocation points per chord+
NXP	10	Maximum number of structural joints per spanwise station (note that if all nodes have same y coordinate NXP must be doubled)

The reset controls are followed by the following sets of data:

1. PLAN - Planform definition

A card consisting of the word PLAN is followed by namelist WNGEOM

Namelist WNGEOM

This namelist defines the wing planform:

XLR	X coordinate, leading-edge root
XLT	x coordinate, leading-edge tip
XTR	x coordinate, trailing-edge root
XTT	x coordinate, trailing-edge tip
YTIP*	y coordinate, tip
XL ()	a vector of length NL containing the x coordinates of crank points on leading edge
YL ()	a vector of length NL containing the y coordinates of crank points on leading edge

+ See discussion of aerodynamic grid

* It is assumed that at the root $y = 0$

XT () a vector of length NT containing the x coordinates of
 crank points on trailing edge

YT () a vector of length NT containing the y coordinates of
 crank points on trailing edge

For an uncranked wing, the following parameters may be used in namelist
 WNGEOM instead of the preceding list:

AR aspect ratio, $\text{Span}^2/\text{Area}$

AREA surface area, $(\text{Span}) (\text{Root chord} + \text{Tip Chord})/2$

ANGLE1 leading-edge sweep angle, degrees

TAPER taper ratio, $\text{Tip chord}/\text{Root chord}$

2. AERO - Aerodynamic grid definition

The purpose of this input data is to define the locations of the
 aerodynamic chords to be used in the aerodynamic calculation and the set
 of structural joints that lie in the vicinity of these chords. Five to
 ten such chords are usually adequate for the aerodynamic calculations.
 Their location should be chosen to be uniformly distributed over the
 wing and to minimize the distance between the chordline and the
 structural joints that are used with it.

A record consisting of the word AERO is followed by the following
 records:

- (i) Chord definition coordinates; NYS non-dimensional y coordinates
 (fractions of semispan), must be monotonically increasing
- (ii) Definition of collocation chords - NYC integers. It is
 recommended that even numbers of collocation chords be used.
- (iii) Node definition records, one per chord - the numbers of structural
 joints lying in the vicinity of the aerodynamic chords.

Input/Output Data Sets

The processor AERO needs basic data sets generated by TAB to define the joint coordinates. It generates the following data sets containing the definition of the wing geometry and the aerodynamic grid. These are:

COL WING, BL76 WING, BLK5 WING, SUBK CRNK, and AERO GRID

INPUT EXAMPLE:

RESET ALIB=3, NXP=14, NL=1, NT=1\$

PLAN\$ WING PLANFORM DATA FOLLOWING ON NAMELIST WNGEOM

\$WNGEOM XLR=0., XLT=1030., XTR=1180., XTT=1180.,

YTIP=850., XL(1)=515., YL(1)=425., XT(1)=1180., YT(1)=425.\$END

AERO\$ AERODYNAMIC GRID DEFINITION

0. .25 .45 .61 .74 .85 .93\$ NONDIMENSIONAL CHORD

STATIONS

2 4 6 7 \$ SECOND, FOURTH, SIXTH, AND SEVENTH CHORD

USED FOR COLLOCATION

1 2 3 4 5 6 7\$ NODES 1 THRU 7 DEFINE THE FIRST CHORD

10 11 12 13 14 15

18 19 20 21

22 23 24 25 : 26 27 28 29\$

30 31 32 33 : 34 35 36\$

SUBKRN - SUBSONIC KERNEL FUNCTION AERODYNAMICS PROCESSOR

Function

SUBKRN calculates aerodynamic matrices based on a subsonic kernel function formulation.

RESET Controls

<u>Name</u>	<u>Default Value</u>	<u>Meaning</u>
ALIB	1	Source and destination library for aerodynamic data reduced-frequencies set number

The reset controls are followed by a card containing the word KSET and namelist NAML.

Namelist NAML

The namelist defines the reduced frequency set

<u>Name</u>	<u>Default Value</u>	<u>Meaning</u>
MACH	0.	Mach number
KMAX	4.	Maximal reduced frequency
KMIN	0.	Minimal reduced frequency
KMED	1.	Median reduced frequency
VK ()		Values of reduced frequencies
NK	20	Number of reduced frequencies

The reduced frequency is defined as $k = (b_0 \omega / V)$ where b_0 is the root semi-chord of the wing, ω is a frequency, and V is the air speed. There are two options for defining the reduced frequency set. One option is to read in the values as the vector VK. If no values are input into VK SUBKRN calculates the aerodynamic matrices for a set of NK reduced frequencies, half of which fall between KMIN and KMED and half between KMED and KMAX. There is no need to calculate the aerodynamic matrices for a large number of reduced matrices. Rather, it is more economical to have NK on the order of 5-20, and then use interpolation (in the flutter analysis) to obtain values for intermediate values of the reduced frequencies.

Input/Output Data Sets

Processor SUBKRN generates two data sets using only input data sets produced by processor AERO:

1) KSET bbbb* kset 0

is of length $2*NK$ and contains the vector VK of the reduced frequencies and a vector VDK of the same frequencies mapped into the interval $(-1,1)$. The variable kset is the frequencies set number

2) AERO MATR kset 0

contains mode independent aerodynamic matrices in NK blocks. Each block contains a real and imaginary matrix of order $NS*NYC$ (see processor AERO), that is, the total number of collocation points.

4.3 GAF - GENERALIZED AERODYNAMIC FORCE PROCESSOR

Function

GAF calculates the generalized aerodynamic forces and their derivatives with respect to the reduced frequencies for a given set of reduced frequencies.

The generalized aerodynamic forces calculated in GAF are the array

$$A_{ij} = C \iint_S \Delta p^j(x,y) z^i ds$$

where S is the wing area, Δp^j is the pressure differential associated with the j -th vibration mode, and z^i is the z component of the i -th mode (the wing middle-surface is assumed to be close to the x - y plane). The constant C

* bbbb denotes a blank name

$$C = 8\pi ql^2/b_o$$

where q is the dynamic pressure, l is the wing semi-span and b_o the root semi-chord.

Reset Controls

<u>Name</u>	<u>Default Value</u>	<u>Meaning</u>
ALIB	1	Source library for data sets generated by AERO and SUBKRN
VLIB	1	Source library for vibration modes
NEIG	2	Number of vibration modes to be used from data set VIBR MODE
KSET	1	Reduced frequencies set number
L1	1	First vibration mode to be used from VIBR MODE (that is, modes L1, L1+1, . . . are used).
BLIB	1	Source library for data sets generated in TAB

Input/Output Data Sets

Processor GAF uses most of the data sets generated in processors AERO and SUBKRN. Also, data sets defining the joint coordinates and active degrees of freedom from TAB and the vibration modes data set VIBR mode are needed.

The only data set generated by GAF is GAF bbbb* kset 0. This data set contains two blocks of real and imaginary generalized aerodynamic forces, respectively. These have the dimension (NK, NEIG, NEIG) where NK is the number of reduced frequencies. This data set is written back into library ALIB.

* bbbb denotes a blank name

4.4 DKM - GENERALIZED STIFFNESS AND MASS MATRIX DERIVATIVE PROCESSOR

Function

This processor is used to calculate derivatives of the generalized stiffness and mass matrices with respect to the design variables.

Reset Controls

<u>Name</u>	<u>Default Value</u>	<u>Meaning</u>
ELIB	1	source library for element packets
VLIB	1	source library for vibration modes
DLIB	1	source library for design variables
DKML	1	destination library for derivatives of generalized stiffness and mass matrix blocks
L1	1	First mode to be used from data set VIBR MODE MASK MASK
NEIG	2	Number of vibration modes data
MXGR	100	Maximum number of section for any section type
MXGD	5	Maximum number of design variables that control the thickness of the same section
NC	2	Number of design variables
GDIV	1	GDIV should be set to the value of the acceleration of gravity (e.g. 386.4 in/s ²) if weight rather than mass data was supplied in TAB. Otherwise GDIV=1.

DKM can handle only element types E23, E31, E41, and E44.

Input/Output Data Sets

The processor DKM reads most of the data sets generated by processor DESVAR. Additionally it requires data sets generated by ELD, E and EKS defining element properties and the vibration mode data set VIBR MODE. Two data sets are generated by DKM.

1) DKM bbbb* 0 0

containing the array DKM (NEIG, NEIG,NC+1). The first NC matrices stored in DKM are the derivatives of the generalized stiffness matrix with respect to the design variable. The last matrix is the generalized stiffness matrix obtained when all design variables are set to zero.

2) DGM bbbb* 0 0

This data set contains derivatives of the generalized mass matrix with respect to the design variables and has the same structure as the DKM data set.

4.5 FLT - FLUTTER ANALYSIS PROCESSOR

Function

This processor may be used to perform flutter analyses and to calculate derivatives of flutter constraint functions. The vibration modes of the structure are used as generalized coordinates for the flutter analysis. When the structure is resized new modes are calculated as a linear combination of the modes of the initial structure. For this process to be effective the number of nodes calculated for the initial structure should be considerably larger than the number of modes needed for

* bbbb denotes a blank name

the flutter analysis. Typical numbers are 15 to 20 modes for the initial structure (NEIG) and 5 to 8 modes for the flutter analysis (NM).

The flutter constraint used in the optimization process is the continuous flutter constraint of Ref. 3. This constraint formulation includes not only the flutter point but also "hump" mode critical points as constraints. The critical points on the V-g diagram are referred to in the output as "danger points".

Reset Controls

<u>Name</u>	<u>Default Value</u>	<u>Meaning</u>
ALIB	1	Source library for aerodynamic data
DLIB	1	Source library for design variables
VLIB	1	Source library for vibration modes
DKML	1	Source library for derivatives of mass and stiffness matrices
NC	0	Number of design variables
MXDP	6	Three times the maximum number of danger points on the V-g diagram per flutter condition*
NEIG	2	Number of vibration modes for initial structure (order of GAF and DKM matrices)
NM	2	Number of modes needed for flutter analysis
L1	1	First vibration mode used in flutter analysis
GLIB	1	Destination library for flutter constraints

* For each danger point on the V-g diagram the two closest points are also included as danger points.

Reset Controls (Continued)

<u>Name</u>	<u>Default Value</u>	<u>Meaning</u>
		and their derivatives
HALF	1	Set to 0.5 if only one half of one wing structure is modeled (aerodynamic forces are multiplied by HALF)
NF	1	Number of flutter conditions
SHIFT	0.	Eigenvalue shift, used when rigid body modes are present; usually the same value as used in EIG
IOPT	1	Analysis/optimization flag. IOPT=1 for V-g analysis starting from initial altitude and moving toward matched altitude. IOPT=0 for V-g analysis at a single altitude. IOPT=2 same as IOPT=0 plus calculation of derivatives of flutter constraints.
LUNIT	1	Ratio of user length unit to one meter. For example, if the length unit is one foot LUNIT=0.3048
MUNIT	1	Ratio of user mass unit to 1 kilogram. For example, if the mass unit is the slug, MUNIT=14.6
CASE	1	Last name of flutter constraint data set - GG1 cor 1 CASE

The reset controls are followed by a card containing the word FLUTTER and then namelist FLUTER which is repeated NF times.

Namelist FLUTER

<u>Name</u>	<u>Default Value</u>	<u>Meaning</u>
H	0	Initial (for analysis) constraint (for optimization) altitude
REF	1. rad/sec.	Reference frequency
IGAIN	10	Ratio of number of entries in V-g table to NK (number of reduced frequencies for which GAF were computed)
KMED	1.	Median reduced frequency (needed only if automatic reduced frequency generation was called for in processor SUBKRN)
SG	0.02	Structural damping coefficient.
KSET	1.	Reduced frequency set to be used in flutter analyses
MACH	0.	Mach number

Namelist FLUTER is repeated NF times and then followed by namelist OPTIMUM for design variables values and increments.

Namelist OPTIMUM

<u>Name</u>	<u>Default Value</u>	<u>Meaning</u>
C(I)	0.	Vector of design variables*
DC(I)	10^{-4}	Vector of design variable increments used for finite difference calculations of derivatives. On UNIVAC DC (I) should be about 1000 times smaller than the magnitude of the design variables. On CDC-10 ⁶ times smaller.

* Required only if user wishes to override values obtained from data set
DESI VARI 0 0

Input/Output Data Sets

Processor FLT uses most of the data sets generated by processors AERO, SUBKRN, GAF and DKM. Additionally the values of the design variables, DESI VARI and the vibration frequencies VIBR EVAL MASK MASK are used. The data sets GG1 CON 1 case and DG1 CON 1 case containing, respectively, the values of the flutter constraints (as defined in Ref. 7) and their derivatives with respect to the design variables are generated by FLT. Two data sets containing the definition of the flutter conditions, FLTR PACK 0 0 and FLTR PACK -1 0, are also generated and written back into library ALIB.

SECTION 5
OPTIMIZATION

The PARS processors give the user the flexibility of having either automated optimization or "a man in the loop" process. The automated optimization is furnished through the use of loop processors, SOL and FOL and FSDL that call other processors in a predetermined sequence. The static optimization loop SOL and the flutter optimization loop FOL use mathematical programming for the optimization and call processor NEWTON for the resizing process. The third loop, FSDL, provides the user with the option of fully stressed design resizing when only stress constraints are considered. The user has the option of using these optimization loops or creating his own loops by calling the processors in sequence. The sequence does not have to be the same as the one provided by SOL, FSDL or FOL. The following are two examples of such nonstandard sequences.

a. Simultaneous optimization for some outside generated constraints say for example buckling, (which passes the information about the constraints and its derivatives through data sets) and some internally generated constraints like stress, displacement or thermal constraints.

It is assumed that BUCK is the name of the group of some outside processors which generate a set of buckling constraints and its derivatives during each new analysis process. We can create the following loop which performs the simultaneous optimization.

```
@XQT TAB
.
.   Static analysis
.
.
@XQT SSOL
```

@XQT GSF

XQT PSF

@XQT DESVAR \$ Define design variables

@XQT KSDM \$ Define minimum gage

* @XQT BUCK \$ BUCKLING Analysis and derivatives

@XQT DGRAD

@XQT SGRAD Displacement and stress constraints

@XQT DCON and their derivatives

@XQT NEWTON \$ Resize

@XQT KSDM \$ Update structural sizes

@XQT E

.

.

.

@XQT SSOL

@XQT GSF

Repeat sequence from (*)

b. Several one-dimensional searches based on one analysis.

The user may elect (following the PROCESS code [8]) to resize the structure several times based on values of the constraints obtained from processors DCON SCON and BUCK rather than repeat the analysis and derivative calculations after each one dimensional search as is done in processor SOL. This only requires calling NEWTON several times in succession.

5.1 NEWTON: Newton's Method Optimizer

Function: NEWTON finds a search direction and the minimum in that direction in a sequence of unconstrained minimizations (SUMT) using Newton's method. An extended interior penalty function, is employed to account for constraints. Even though NEWTON performs only a single one dimensional search it performs convergence tests which are used when this processor is used in a loop.

Optimization procedure

The object is to minimize the mass M of the structure subject to the constraints

$$g_i(v) \geq 0 \quad (i, \dots, \text{maxcon}) \quad (5.1-1)$$

where,

v represents the vector of n design variables. g_i is the i th constraint and maxcon is the number of constraints. The constrained optimization problem is transformed into the following series of unconstrained problems:

Minimize

$$P(r, v) = M(v) + r \sum_{i=1}^{i=\text{maxcon}} f_i(v) \quad (5.1-2)$$

$$\text{for } r_1, r_2, \dots, r_n \rightarrow 0$$

The sum which is added to the mass is called the penalty function.

The definition of f_i in (5.1-2) for a quadratic extended penalty function is

$$f_i(\underline{v}) = \begin{cases} \frac{1}{g_i} & \text{if } g_i \geq g_0 \\ \frac{1}{g_0} [(g_i/g_0)^2 - 3(g_i/g_0) + 3] & \text{if } g_i \leq g_0 \end{cases} \quad (5.1-3)$$

where g_0 is a transition point.

The definition of f_i in (5.1-2) for a cubic extended interior penalty function [9] formulation is

$$f_i(\underline{v}) = \begin{cases} \frac{1}{g_i} & \text{if } g_i \geq g_0 \\ \frac{1}{g_0} [A \left(-1 + \frac{g_i}{g_0}\right)^3 + \left(1 - \frac{g_i}{g_0}\right)^2 + \left(1 - \frac{g_i}{g_0}\right) + 1] & \text{if } g_i \leq g_0 \end{cases} \quad (5.1-4)$$

The transition parameter g_0 changes with r as

$$g_0 = g_{01} (r/r_1)^p$$

where g_{01} and r_1 are the initial values of g_0 and r , respectively, and p is a constant (usually 0.5). For each value of r the unconstrained minimization is performed using approximate second derivatives required for generating a suitable search direction. A one dimensional search is then conducted for locating the minimum in that direction. The minimization for a single value of r is terminated when any one of the following criteria is satisfied.

(1) The Euclidean norm of the gradient of P is smaller than 0.05 times the gradient of M. (gradient convergence).

(2) NEWTON estimates that at the beginning of the last one-dimensional search P(r,v) is within 2% from its minimum value (Quadratic convergence).

(3) LCNT (a value controlled by a reset parameter) number of one-dimensional searches have been executed.

When one of the convergence criteria is met r is reduced by a factor RDC (see RESET controls) unless both of the following convergence criteria are met

- (1) The penalty function is less than one percent of the weight
- (2) The reduction in the weight during the last one dimensional search is smaller than one percent.

Constraint Approximation

NEWTON performs a complete one dimensional search without calling any analysis routines by approximating a constraint function $g(v)$ based on their values at the initial point of the search v_0 (with components v_{0i} , $i=1, \dots, n$) and their derivatives at that point. The approximation is

$$g(v) \cong g(v_0) + \sum_{j=1}^n B_j (V_j - V_{0j}) \frac{\partial g}{\partial V_j} \quad (5.1-5)$$

where

$$B_j = \begin{cases} 1 & \text{if } \partial g / \partial V_j < 0 \\ \frac{V_{mj} + V_{0j}}{V_{mj} + V_j} & \text{if } \partial g / \partial V_j > 0 \end{cases} \quad (5.1-6)$$

and V_{mj} is a vector of constants input by the user (as GAGE in processor DESVAR). The value $B_j = 1$ corresponds to a linear approximation and the alternative expression in Eq. (5.1-6) corresponds to an inverse approximation. The choice is based on which one is more conservative (see [10], that is, which choice results in a smaller value of $g(v)$). Because in PARS a zero value for the design variable usually corresponds to a nonzero value of the structural parameter that it controls (see Eq. 2.1-1) the constants V_{mj} are introduced. In the simple case where a single design variable V_i controls a structural parameter q_j , Eq. (2.1-1) becomes

$$q_j = q_{oj} + V_i F_{ij}$$

In such a case $V_{mi} = q_{oj}/F_{ij}$ would correspond to an approximation linear in the reciprocal of q_j . In the more general case where a single design variable controls several structural parameters or vice versa the choice of a good value of V_{mj} is more difficult. The guideline to that choice is that when $V_j = -V_{mj}$ the approximation becomes unbounded. For example if V_j controls the thickness of one lamina in a composite material a reasonable choice for V_{mj} is the laminate thickness when $V_j = 0$. If a very large value is selected for V_{mj} the approximation becomes linear.

Relation to Other Processors

NEWTON can be used as an integral part of PARS or as a stand alone processor used in conjunction with another application program that calculates the constraints and their derivatives. If NEWTON is used in such a mode the user, through the use of processor AUS or through the application program, needs to supply data sets which are otherwise

generated by other PARS processors. The following data sets are required by NEWTON.

1. DESI VARI 0 0: Contains the values of the design variables at the beginning of the search. On exit it contains the updated value of the design variables. (NI = 1, NJ = NDV)
2. DVMX VARI 0 0: Contains the maximum values of the design variables and is required only if such bounds are used. (NI = 1, NJ = NDV)
3. GAGE VARI 0 0: Contains the values of V_{mi} (NI = 1, NJ = NDV)
4. MASS DER 0 0: Contains the derivatives of the mass with respect to the design variables.
5. GG1 CON set case: Contains the values of the constraint functions g_i (NI = number of constraints, NJ = 1).
6. DG1 CON set case contains the derivatives of the constraints with respect to the design variables (NI = NDV, NJ = 1, each constraint in a block).

NEWTON creates a data set D12A BLOC 0 0 which contains the parameters that control the optimization, most of which are reset parameters. If this data set exist it overrides the reset values.

The contents of this data set are:

R,RDC,GLIM,GMIN,P,GIC,DELC,LCNT,KOUNTR,NFINAL where

GMIN - the value of the most critical constraint function excluding design variable bounds

KOUNTR - the number of the present one dimensional search in the sequence for the current r-value

NFINAL - a termination flag. Set to one by NEWTON when termination criteria are met.

The other parameters in the data set are explained as reset parameters.

<u>Name</u>	<u>Default Values</u>	<u>Description</u>
LCNT	7*	LCNT is the maximum number of one dimensional searches
R	50.	Factor of penalty function [r]., recommended value, $r=0.1 g_0 M$ where M is the expected minimum mass.
RDC	20.*	Reduction factor for r [1/RDC].
GLIM	0.5	Limiting value for critical constraints.
P	0.5*	Exponent of [$g_0 = cr^P$].
GIC	0.32*	The initial value for g_0 .
DC	1.E-04	The increment in finite difference
HHLI	1	Destination library for Hession matrix.
GGLI	1	Library containing GGL CON mask mask and DGL CON mask mask data sets.
SET+	1	Constraint set.
L1+	1	First case constraint.
L2+	1	Last constraint case.
DVLIB	1	Library containing data sets generated by processor DESVAR.
OEPF	2	Order of extended interior penalty function to be used.

* Default value is recommended.

+ Newton will process all constraints stored in data sets GGL CON SET CASE where CASE runs from L1 through L2.

5 SOL - Static Optimization Loop

Function: The SOL processor provides a structural optimization loop for static-strength (stress and displacements) and minimum-gage constraints under several alternative load conditions. Various processors (the list is given below) are called in a sequence to obtain the minimum weight structural design. The user may elect to run the individual processors one by one instead of using SOL. This, however, will require repeating the reset cards for each call.

It is assumed that minimum gage and static stress constraints are always present. Constraints imposed on joint displacements may or may not be specified. Design variables can be a physical dimension of a structural element (thickness for 3 and 4 node elements and area for 2 node elements) or the nodal temperatures specified at certain joint locations.

The following is the sequence in which various processors are called. Processor identification number and their functions are outlined briefly.

<u>Processor Name</u>	<u>Processor Identification Number (IDP)</u>	<u>Description</u>
KSDM	28	Updates structural sizes.
E	35	Generates element packets
EKS	36	Calculate intrinsic stresses and stiffness matrices
K	13	Assembles system stiffness matrix.
INV	16	Factors assembled system stiffness matrix.
SSOL*	18	Computes displacements and reactions due to point loading applied at joints.

* These processors correspond to standard SPAR processors, however, they are modified here to some extent in order to include them in the optimization loop.

<u>Processor Name</u>	<u>Processor Identification Number (IDP)</u>	<u>Description</u>
EQNF*	17	Computes equivalent joint loading and initial strain data.
GSF*	20	Generates element stresses and internal load information.
PSF*	21	Prints element stresses and internal load information.
DGRAD	29	Calculates derivatives of displacements with respect to design variables controlling a thickness or area.
TGRAD	27	Calculates derivatives of displacement with respect to design variables controlling nodal temperatures.
DCON	32	Forms the displacement ratios for the nodes where nodal displacements are constrained.
SGRAD	31	Evaluate the derivatives of element stresses.
SCON	30	Forms the stress ratio at the centroid of finite elements
NEWTON	34	Implements the sequence of unconstrained minimization technique using Newton's Method.

* These processors correspond to standard SPAR processors, however, they are modified here to some extent in order to include them in the optimization loop.

One or more of the above processors can be skipped to restart the static optimization loop in case a run terminates in the middle due to some unforeseen circumstances like maximum time or any input data error. For detail refer to "Restart Execution Mode." later in this section.

Reset Controls:

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
ELIB	1	Library containing the element packets.
QLIB	1	Library containing static displacement data sets.
BLIB	1	Library containing output data sets generated by TAB and ELD processors.
MING	0	(See processor KSDM) if a non zero value of MING is specified in this processor the reset parameter IND should be set to a non zero value in processor KSDM.
IPRT	0	If other than zero, processor PSF is called to print out element stresses and derivatives of element stresses.
ITDV	2	ITDV=2 - Problem does not contain temperatures ITDV=1 - Temperatures are used but no temperature design variables ITDV=0 - Temperature design variable are used

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
AMAX	2	The maximum number of times the optimization loop is executed and the design variables are updated before the run is terminated.
IDP	28	This is used to restart the loop if SOL terminates abnormally maximum time or otherwise.

Execution Control Parameters:

The following is the sequence of data cards required when executing SOL for the first time (for restart refer "Restart Execution Mode")

@XQT SOL

Reset Cards (if any)

STOP\$ IDF = 33, SOL

RESET IND=1

a. (Additional Reset cards for KSDM)

STOP\$ IDP = 28, KSDM

b. (Reset cards for E)

STOP \$ IDP = 35

c. (Reset cards for EKS)

STOP \$ IDP = 36

d. (Reset cards for K)

STOP \$ IDP = 13, K

e. (Reset cards for INV)

STOP\$ IDP = 16, INV

f. (Reset cards for SSOL)

STOP\$ IDP = 13, SSOL.

- g. (Reset cards for GSF)
STOP\$ IDP = 20, GSF
- h. (Reset cards for PSF) } + These cards are optional and should
STOP\$ IDP = 21, PSF } be used only if reset parameter
- i. (Reset cards for DGRAD) 'IPRT' \neq 0. in 'SOL'
STOP\$ IDF = 29, DGRAD
- j. (Reset cards for TGRA)
STOP\$ IDP = 27, TGRAD
- k. (Reset cards for DCON)
DISCON: $i_1, j_1, adv_1; i_2, j_2, adv_2; i_3, j_3, adv_3$
STOP\$ IDP = 32, DCON
- l. (Reset cards for SGRA)
STOP\$ IDP = 31, SGRAD
- m. (Reset cards for DPSF) } +
STOP\$ IDP = 23, DPSF }
- n. (Reset cards for SCON)
STRESS: EXY: $ng_1, mg_1, s_1^{(1)}, s_2^{(1)}, s_{12}^{(1)}$ \$
 $ng_2, mg_2, s_1, s_2, s_{12}; ng_3, mg_3, s_1^{(3)}, s_2^{(3)}, s_{12}^{(3)}$:
-----\$
STOP\$ IDP = 30, SCON
- o. (Reset cards for NEWTON)
STOP\$ IDP = 34, NEWTON

STOP\$ card serves as an input data card separator for a processor. When it is encountered during a read mode, it stops scanning any further data cards.

Restart Execution Mode:

In restart, the reset and STOP\$ cards for main processor SOL are always required, and should be supplied first. The other cards should follow the normal execution mode order. A few of the typical situations are illustrated through examples.

Example 1.

Assume that while executing DGRAD (item i of SOL) the run was terminated due to maximum page or time limits or may be due to improper RESET cards. If now one intends to restart the program, the following sequence should be used:

@XQT SOL

RESET IDP = 29

(Other Reset cards if any)

STOP\$ IDP = 33, SOL

i. (Reset cards for DGRAD)

STOP\$ IDP = 29, DGRAD

j. (Reset cards for TGRAD)

STOP\$ IDP = 27, TGRAD

k. (Reset cards for DCON)

DISCON: $i_1; j_1, adv_1; i_2, j_2, adv_2; i_3, j_3, adv_3$

STOP\$ IDP = 32, DCON

l. (Reset cards for SGRAD)

STOP\$ IDP = 31, SGRAD

m. (Reset cards for PSF) } These cards are optional and should
 STOP\$ IDP = 23, PSF } be used only if reset parameter 'IPRT'

n. (Reset cards for SCON) $\neq 0$. in 'SOL'
 STRESS: EXY: $ng_1, mg_1, s_1^{(1)}, s_2^{(1)}, s_{12}^{(1)}$ \$
 $ng_2, mg_2, s_1^{(2)}, s_2^{(2)}, s_{12}^{(2)}$: $ng_3, mg_3, s_1^{(3)}, s_2^{(3)}, s_{12}^{(3)}$:--\$
 STOP\$ IDP = 30, SCON

o. (Reset cards for NEWTON)
 STOP\$ IDP = 34, NEWTON

Reset IND = 1

a. (additional Reset cards for KSDM)
 STOP\$ IDP = 28, KSDM

b. (Reset cards for E)
 STOP\$ IDP = 35

c. (Reset cards for EKS)
 STOP\$ IDP = 36

d. (Reset cards for K)
 STOP\$ IDP = 13, K

e. (Reset cards for INV)
 STOP\$ IDP = 16, INV

f. (Reset cards for SSOL)
 STOP\$ IDP = 18, SSOL

g. (Reset cards for GSF)
 STOP\$ IDP = 20, GSF

h. (Reset cards for PSF) } These cards are optional and
 STOP\$ IDP = 21, PSF } should be used only if reset
 parameter 'IPRT' $\neq 0$. in SOL.

Example 2.

Assume that each one of the processors a to h has already been executed separately outside SOL. To optimize the structure, the SOL loop can be entered at i using the same set of cards as shown for

Example 1.

5.3 FOL - FLUTTER OPTIMIZATION LOOP.

FUNCTION:

The FOL processor is a flutter optimization loop for the minimum weight design of a structure subject to flutter and minimum gage constraints. It calls the various processors required for the optimization and thus saves the user the tedium of calling them himself and repeating RESET and other input cards. Design variables are element thicknesses or areas (see DESVAR) and concentrated mass at structural joints.

The following is the sequence in which various processors are called.

<u>Processor Name</u>	<u>Processor ID Number</u>	<u>Function</u>
KSDM	28	Updates intrinsic stiffness and stress matrices and global (diagonal) mass matrix.
E	35	Generates element packets
EKS	36	Calculates intrinsic stiffness and stress matrices
K	13	Assembles global stiffness matrix.
AUS	23	Arithmetic Utility System*
INV	16	Factors assembled stiffness matrix.
EIG	22	Calculates vibration modes.

* AUS is most commonly used for forming the desired mass matrix or introducing a shift into the stiffness matrix.

<u>Processor Name</u>	<u>Processor ID Number</u>	<u>Function</u>
GAF	38	Calculates generalized aerodynamic forces.
DKM	40	Calculates derivatives of generalized stiffness and mass matrices.
FLT	39	Performs flutter analysis and calculates derivatives of flutter constraints.
NEWTON	34	Updates design variables using Newton's method with approximate second derivatives.

RESET CONTROLS:

<u>Name</u>	<u>Default Value</u>	<u>Description</u>
BLIB	1	Library containing TAB & ELD generated data sets.
ELIB	1	Library containing element packets.
MLIB	1	Library containing mass matrix.
MING	0	See processor (KSDM), if a nonzero value of MING is specified in this processor the reset parameter IND should be nonzero in KSDM.
IDP	28	ID of first processor called by FOL.
AMAX	2	The number of times the processor loop is repeated.
FRAC	0.1	Maximum fractional change in the square of the first vibration frequency that is acceptable without mode updating.

VARIATION IN LOOP STRUCTURE:

The two reset parameters IDP and FRAC give the user the possibility of changing the basic loop structure. IDP is the identification number of the first processor to be executed and thus permits the user to start the loop at any point. This freedom is especially important for restart. FRAC permits the user to loop for most of the design process only over the last two processors FLT and NEWTON. The idea is that if the changes in the structure are not significant there is no need to update the vibration modes every time the design variables are changed. The change in the dynamic properties of the structure is measured by the change in the first vibration frequency. The user specifies FRAC which is the maximum fractional change of the square of the first vibration frequency that may be tolerated without updating modes.

EXECUTION CONTROL PARAMETERS:

The following is the sequence of data cards required when executing FOL with IDP=28:

@XQT FOL

Reset Cards

Stop\$ FOL

Reset Cards for KSDM

Stop\$ KSDM

Reset Cards for E

Stop\$ E

Reset cards for EKS

Stop\$ EKS

Reset Cards for K

Stop\$ K

Reset Cards for AUS
 Input Cards for AUS
 Stop\$ AUS
 Reset Cards for INV
 Stop\$ INV
 Reset Cards for EIG
 Stop\$ EIG
 Reset Cards for GAF
 Stop\$ GAF
 Reset Cards for DKM
 Stop\$ DKM
 Reset Cards for FLT
 Data for FLT (Namelist FLUTER, etc.)
 Stop\$ FLT
 Reset Cards for NEWTON
 Stop\$ NEWTON
 Repeat Input Cards for AUS

Examples:

Several typical processor sequences are presented

a.) Optimization of a wing structure which has been analyzed in the past.

@XQT TAB	}	Define structural grid
@XQT ELD		
@XQT TOPO		
@XQT DESVAR \$		Define design variables
@XQT AERO		Basic aerodynamic calculation
@XQT SUBKRN		
@XQT FOL		Flutter optimization

b. Flutter Analysis and then optimization.

```
@XQT TAB
@XQT ELD          Define structural grid
@XQT TOPO
@XQT E
@XQT EKS
@XQT K           Calculate vibration modes
@XQT INV
@XQT AUS
@XQT AERO
@XQT SUBKRN      Calculate GAF's
@XQT GAF
@XQT FLT $       Flutter analysis
@XQT DESVAR $    Define design variables
@XQT SKDM $      Define minimum gage structure (set MING=0)
@XQT FOL$       Flutter optimization
```

In this case the RESET cards following @XQT FOL are RESET IDP=40, . . . \$

loop starts at DKM

stop\$ FOL

Reset cards for DKM

stop\$ DKM

Reset Cards for FLT

Data for FLT

stop\$ FLT

Reset cards for NEWTON

stop\$ NEWTON

Reset Cards for KSDM

stop\$ KSDM

⋮

Reset cards for GAF

c. Restart

Suppose a previous run stopped on maximum time after executing INV, and we wish to restart. It is assumed that data sets have been preserved.

@XQT FOL

Reset cards for FOL (IDP=22)

stop\$ FOL

Reset cards for EIG

stop\$ EIG

.

Reset cards for INV

stop\$ INV

5.4 FSDL - Full Stress Design Loop

The FSDL processor provides a Fully Stressed Design loop (FSD or TFSD) for static strength (applied forces and temperatures) and minimum gage constraints under several alternative load cases.

Various processors are called in a sequence to obtain the Fully Stress structural design. The user may elect to run the individual processors one by one instead of using FSDL. This, however, requires repeating the reset cards for each iteration.

TFSD (Temperature Full Stress Design) algorithm, which often converges faster than FSD when thermal stresses are significant, is also available. If the data set MCTR LCAS 0 0 is found in QLIB library (see SCON processor for details) TFSD is used rather than FSD.

The following is the sequence in which various processors are called:

<u>Processor Name</u>	<u>Processor Identification Number</u>	<u>Description</u>
KSDM	28	Updates structural sizes..
E	35	Generates element rockets.
EKS	36	Calculates intrinsic stresses and stiffness matrices.
K	13	Assembles system stiffness matrix.
INV	16	Factors assembled system stiffness matrix.
EQNF	17	Computes equivalent joint loading and initial strains.
SSOL	18	Computes displacements and reactions due to applied joint loadings.
GSF	20	Generates element stresses.
SCON	31	Generates nondimensional stress ratios and resize factors.

Reset controls:

Name	Default value	Description
MING	0	See processor KSDM.
AMAX	2	Maximum of iterations FSD (or TFSD) loop is executed, design variables and current section dimensions (CDIM data sets) are updated before termination.
IDP	28	Used to restart the loop if FSDL terminates abnormally.

ITDV	2	= 1 EQNF is executed in the loop, = 2 EQNF is not executed in the loop,
INPL	0	= 0 Old data sets are being disabled and new ones (created by executing the different processors in the current iteration) are generated in cumulative fashion. = 1 New data sets are generated in the same disk location as the old ones.

The following is the sequence of data cards required when executing FSDL (on CDC computer) when terminal load cases are applied.

@XQT FSDL

Reset cards for FSDC

STOP\$ FSDL IDP=33

Reset IND=1

Additional reset cards for KSDM

STOP\$ KSDM IDP=28

Reset cards for E

STOP\$ E IDP=35

Reset cards for EKS

STOP\$ EKS IDP=36

Reset cards for K

STOP\$ K IDP=13

Reset cards for INV

STOP\$ INV IDP=16

Reset cards for SSOL

STOP\$ SSOL IDP=18

Reset cards for GSF

STOP\$ GSF IDP=20

Reset cards for SCON

data for allowable stress (See in SCON processor)

STOP\$ SCON IDP=31

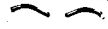
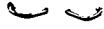
When nodal temperatures are applied execute FSDL with resetting ITDV to 1,
and add the reset control cards for EQNF and STOP\$ card after the STOP\$
card of INV.

REFERENCES

1. Prasad, B., "Automated Design of Structures Subjected to Stress, Displacement and Temperature Constraints," PH.D. Dissertation, Department of Mechanics, and Mechanical and Aerospace Engineering, Illinois Institute of Technology, Dec. 1977.
2. Haftka, R.T. and Starnes, J.H., Jr., WIDOWAC (Wing Design Optimization with Aeroelastic Constraints): Program Manual, NASA TM X-3071, 1974.
3. Whetstone, W. D., SPAR Reference Manual - System Level 11. Engineering Information Systems Inc. EISI/A2200, December 1976.
4. Adelman, H. M., and Narayanaswami, R. "Resizing Procedure for Structures Under Combined Mechanical and Thermal Loading", AIAA J. Vol. 14, pp. 1484-1486, 1976.
5. Tsai, S.W. and Wu, E.M., "A General Theory for Strength for Anisotropic Materials," J. Comp. Mat'l, Vol. 5, pp. 58, 1971.
6. Narayanaswami, R. and Adelman, H.M., "Evaluation of the Tensor Polynomial and Hoffman Strength Theories for Composite Materials", J. Comp. Mat'l, Vol. 11, pp 366-377, 1977.
7. Haftka, R. T., "Parametric Constraints with Applications to Optimization for Flutter Using a Continuous Flutter Constraint", AIAA J. Vol. 13, pp. 471-475, 1975.
8. Schmit, L. A., and Miura, H., "An Advanced Structural Analysis/Synthesis Capability - ACCESS 2, Proceedings of the 17th AIAA/ASME SAE Structures, Structural Dynamics and Materials Conference, King of Prussia, Pa., May 5-7, 1976, pp. 432-447.

References (cont'd)

9. Prasad B., and Haftka, R. T., "A Cubic Extended Interior Penalty Function for Structural Optimization". to be published in the International Journal for Numerical Methods in Engineering.
10. Starnes, J. H., Jr., and Haftka, R. T. "Preliminary Design of Composite Wings for Buckling, Strength and Displacement Constraints", AIAA Paper 78-466, 1978.



3
7

3
7

—

—

•