General Rotorcraft Aeromechanical Stability Program (GRASP) Version 1.03–User's Manual

A. Stewart Hopkins and Donald L. Kunz

NASA

National Aeronautics and
Space Administration

US ARMY
AVIATION
SYSTEMS COMMAND

AVIATION RESEARCH AND
TECHNOLOGY ACTIVITY

General Rotorcraft Aeromechanical Stability Program (GRASP) Version 1.03–User's Manual
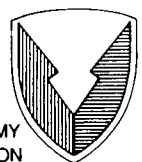
A. Stewart Hopkins
Donald L. Kunz, Aeroflightdynamics Directorate, U.S. Army Aviation Research and
    Technology Activity, Ames Research Center, Moffett Field, California

CONTENTS

PRECEDING PAGE BLANK NOT FILMED

PAGE _ll_ INTENTIONALLY BLANK

GENERAL ROTORCRAFT AEROMECHANICAL STABILITY PROGRAM

(GRASP) VERSION 1.03–USER'S MANUAL

A. Stewart Hopkins
Donald L. Kunz
Aeroflightdynamics Directorate
U.S. Army Aviation Research and Technology Activity
Ames Research Center

SUMMARY

The Rotorcraft Dynamics Division, Aeroflightdynamics Directorate, U.S. Army Aviation Research and Technology Activity has developed the General Rotorcraft Aeromechanical Stability Program (GRASP) to perform calculations that will assess the stability of rotorcraft in hovering flight and ground contact conditions. The program is designed to be a state-of-the-art, hybrid, finite-element/multibody code that can be applied to all existing and future helicopter configurations. While GRASP was specifically designed to solve rotorcraft stability problems, its innovative structure and formulation allow for applications to a wide range of structures. This manual describes the preparation of the input file required by Version 1.03 of GRASP, the procedures used to invoke GRASP on the NASA Ames Research Center Cray X-MP 48 computer, and the interpretation of the output produced by GRASP. The parameters used by the input file are defined, and summaries of the input file and the job control language are included.

1. INTRODUCTION

The General Rotorcraft Aeromechanical Stability Program (GRASP) is a collection of computer programs designed to assess the stability of rotorcraft in hovering flight or ground contact. To make use of the features provided by GRASP, a user must be able to prepare a definition of the problem to be solved, execute the program, and interpret the output from the program.

The first step in using GRASP is the transformation of the problem to be solved from analytical terms into a format which the program can accept. Section 2 defines the formats of the files which must be provided to GRASP. These files specify the models and the solutions desired.

The second step is the execution of the GRASP programs. Section 3 defines the procedures for invoking GRASP.

The third step is the interpretation of the results in the format presented by GRASP in terms of the analytical problem. Section 4 describes the formats of the steady-state and eigensolution results.

2. PREPARING INPUT

This section describes the formats of the files which must be input to GRASP. These files specify the models and the solutions desired.

2.1. User Data

User Data is a hierarchically ordered collection of records (i.e., card images) consisting of up to 80 characters each. Blank characters may be placed before, be freely interspersed with, or may follow the characters defining any record. Every record begins with a keyword (or keywords since blanks are not relevant) that identifies the record. If the record contains data, an equals delimiter "=" follows the keyword, and the data are entered thereafter. In some cases, a semicolon terminator ";" must follow the data. On any line of the file, all characters following an exclamation point "!" are ignored. This feature may be used to insert comments in the input stream. Comments may not, however, be inserted in the middle of a record.

The data specified in a record may consist of one or more integer, real, or character constants, each separated from the next by a comma delimiter, ",". An integer constant consists of an optional sign (either a minus sign "-", or a plus sign "+") followed by one or more of the integers (0,...,9). A real constant consists of an integer constant subfield, an optional fraction subfield and an optional exponent. The fraction subfield consists of a decimal point "." followed by zero or more integers. The exponent consists of an optional exponent character "D" or "E", an optional sign (however, the sign is mandatory if the exponent character is omitted) and one to three integers. A character constant consists of any collection of characters of less than or equal to the specified length, delimited by apostrophes, " ' " (blanks are relevant in the case of a character constant).

The overall hierarchical structure of the User Data file is outlined in Appendix A. The detailed requirements and meanings of each of the records are described below. The first record must be the key word:

<div align="center">BEGIN DATA</div>

and the last record must be the key word:

<div align="center">END DATA</div>

Three sets of records: "Configuration Data", "Control Data", and "Analysis Data" may appear in arbitrary order between the first and last records. These record sets are discussed in sections 2.1.1, 2.1.2, and 2.1.3, respectively.

2.1.1. Configuration Data

"Configuration Data" consists of no data or the following set of records:

```
BEGIN CONFIGURATION DATA
    DIAGNOSTICS LEVEL = (integer, default = 1)
    PAGE LENGTH = (integer, default = 60)
    PAGE WIDTH = (integer, default = 132)
    RUN TITLE = (72 character literal, default = ' ')
END CONFIGURATION DATA
```

The diagnostics level defines the level of error information to be printed. A diagnostics level of 1 prints warning and error messages. A level of 2 suppresses warning messages. A level of 0 dumps status information and the information manager tables, and prints error messages upon abnormal termination. The page length defines the number of lines to be printed on each page of output, while the page width defines the number of characters to be printed on each line. The run title is the title that will be placed on the opening and closing banner pages as well as at the top of each page of output. These records may appear in arbitrary order within the record set.

2.1.2. Control Data

"Control Data" controls the execution of GRASP and consists of the following ordered set of records:

```
BEGIN CONTROL DATA
    NUMBER OF CASE DEFINITIONS = (integer, default = 0)
    BEGIN DEFINITIONS OF CASES
        "Control Data for Cases"
    END DEFINITIONS OF CASES
END CONTROL DATA
```

where "Control Data for Cases" consists of the specified number of the following set of records:

```
BEGIN CASE DEFINITION
    CASE IDENTIFIER = (8 character literal)
    CASE TITLE = (72 character literal, default = ' ')
    NUMBER OF DIRECTIVES = (integer, default = 0)
    BEGIN DIRECTIVES
        "Directives"
    END DIRECTIVES
END CASE DEFINITION
```

The case definition consists of the case identifier, the case title, the number of directives, and a set of directives. The case identifier points to a BEGIN CASE DATA entry under "Data for Cases" (section 2.1.3). The case title defines the title that is placed in a message at the beginning and end of the case, and at the top of each page of output for that case. If the case title is defaulted, GRASP will print blank spaces.

"Directives" consists of the specified number of any of the following directives in the order in which they are to be executed:

```
SOLVE STEADY-STATE
    MDLID = (8 character literal)
    SSPPAR = (8 character literal, default = ' ')
```

```
MDLNAM = (8 character literal, default = 'MDLNAM$D')
SSPDEF = (8 character literal, default = 'SSPDEF$D')
SSPRSD = (8 character literal, default = 'SSPRSD$D')
SSPSLN = (8 character literal, default = 'SSPSLN$D')
;
```

The solve steady-state parameters include the model identifier MDLID, the steady-state problem parameters identifier SSPPAR, the name of the system model information structure MDLNAM, the name of the steady-state problem deformations information structure SSPDEF, the name of the steady-state problem residuals information structure SSPRSD, and the name of the steady-state problem solution information structure SSPSLN. The model identifier points to a BEGIN MODEL DEFINITION entry under "Model Definition" (section 2.1.3.2). The steady-state problem parameters identifier points to a BEGIN ZXSSQ DEFINITION entry under "Steady-state Problem Parameters" (section 2.1.3.1.1).

If the name of the system model is specified to be other than the default name, the system model will be printed immediately after the model building is complete. The system model is always automatically released when it is no longer needed by the directive. If the names of the steady-state problem deformations and residuals are not defined to be other than the default name, they are also released when they are no longer needed by the directive. If the name of the steady-state problem solution is not defined, it is automatically printed out, then released at the end of the directive. If the name is defined, it is neither printed nor released.

```
SOLVE ASYMMETRIC EIGEN-PROBLEM
     MDLID = (8 character literal)
     AEPPAR = (8 character literal, default = ' ')
     MDLNAM = (8 character literal, default = 'MDLNAM$D')
     REFDEF = (8 character literal, default = 'REFDEF$D')
     REFRSD = (8 character literal, default = 'REFRSD$D')
     AEPMTX = (8 character literal, default = 'AEPMTX$D')
     AEPISL = (8 character literal, default = 'AEPISL$D')
     AEPSLN = (8 character literal, default = 'AEPSLN$D')
     ;
```

The solve asymmetric eigenproblem parameters include the model identifier MDLID, the asymmetric eigenproblem parameters identifier AEPPAR, the name of the system model information structure MDLNAM, the name of the reference deformations information structure REFDEF, the name of the reference residuals information structure REFRSD, the name of the asymmetric eigenproblem mass, damping, and stiffness matrices information structure AEPMTX, the name of the asymmetric eigenproblem intermediate solution information structure AEPISL, and the name of the asymmetric eigenproblem solution information structure AEPSLN. The model identifier points to a BEGIN MODEL DEFINITION entry under "Model Definition" (section 2.1.3.2). The asymmetric eigenproblem parameters identifier points to a BEGIN COMPLEX EIGEN DEFINITION entry under "Asymmetric Eigenproblem Parameters" (section 2.1.3.1.2). If the name of the system model is specified to be other than the default name, the system model will automatically be printed after the model is built.

If the name of the system model is specified to be other than the default name, the system model will be printed immediately after the model building is complete. The system model

is always automatically released when it is no longer needed by the directive. The names of the reference deformations and residuals correspond to the SSPDEF and SSPRSD entries for a SOLVE STEADY-STATE directive. If either of these names are not defined, this directive will not work. If the name of the asymmetric eigenproblem matrices is specified to be other than the default name, the matrices will be printed immediately after their assembly is complete. The matrices are also always automatically released when they are no longer needed by the directive. If the name of the asymmetric eigenproblem intermediate solution is not defined to be other than the default name, it is released when it is no longer needed by the directive. If the name of the asymmetric eigenproblem solution is not defined, it is automatically printed out, then released at the end of the directive. If the name is defined, it is neither printed nor released.

    PRINT DATA SET
        DSNAME = (8 character literal, default = ' ')
        ;

The only parameter for this directive is the name of the information structure to be printed, DSNAME.

    RELEASE DATA SET
        DSNAME = (8 character literal, default = ' ')
        ;

The only parameter for this directive is the name of the information structure to be released, DSNAME.

Note that the parameters may appear in arbitrary order on the same line as the directive, or on subsequent lines. For each directive, a semicolon must be used after the last parameter to terminate that directive. All information structure names must contain only alphabetic (A - Z) and numeric (0 - 9) characters. The use of the dollar sign is reserved for default names. Information structures whose names have been defaulted through the directive parameters may not be printed, saved, or released.

2.1.3. Analysis Data

"Analysis Data" consists of one or more of the following ordered sets of records:

BEGIN ANALYSIS DATA
    NUMBER OF CASES OF DATA = (integer, default = 0)
    BEGIN DATA FOR CASES
        "Data for Cases"
    END DATA FOR CASES
END ANALYSIS DATA

where "Data for Cases" consists of the specified number of the following ordered record set in arbitrary order:

    BEGIN CASE DATA = (8 character literal)
        "Case Data"
    END CASE DATA

The data case identifier corresponds to one of the CASE IDENTIFIER entries defined under "Control Data" (section 2.1.2).

"Case Data" consists of one or more of the following ordered sets of records in arbitrary order:

"Solution Parameter Definitions"
"Model Identifiers Definitions"
"Model Parameter Definitions"

2.1.3.1.   Solution Parameter Definitions

"Solution Parameter Definitions" is the collection of information defining the conditions under which the selected solution methods are to be used. "Solution Parameter Definitions" consists of no data of the following ordered set of records:

BEGIN SOLUTION PARAMETERS
    "Solution Parameters Records"'
END SOLUTION PARAMETERS

where "Solution Parameters Records" consists of one or more of the following sets of records in arbitrary order:

"Steady-state Problem Parameters"
"Asymmetric Eigenproblem Parameters"

2.1.3.1.1.   Steady-state Problem Parameters .

"Steady-state Problem Parameters" is the collection of information that defines the parameters used in the steady-state equation solver (ZXSSQ), and consists of no data or the following ordered set of records:

NUMBER OF ZXSSQ PARAMETERS = (integer, default = 0)
BEGIN ZXSSQ PARAMETERS
    "ZXSSQ Parameters Definitions"
END ZXSSQ PARAMETERS

where "ZXSSQ Parameters Definitions" consists of the specified number of the following record set in arbitrary order:

BEGIN ZXSSQ DEFINITION = (8 character literal)
    NSIG = (integer, default = 10)
    EPS = (real, default = E-10)
    DELTA = (real, default = E-10)
    MAXFN = (integer, default = 100000)
    IOPT = (integer, default = 0)
    PARM = (real array of length 4, default = 0.01, 2.0, 120.0, 0.1)
END ZXSSQ DEFINITION

The ZXSSQ identifier corresponds to either the ZXPID entry under "Aeroelastic Beam Definitions" (section 2.1.3.3.2a) or the SSPPAR entry for the SOLVE STEADY-STATE directive under "Control Data" (section 2.1.2). The definition consists of the convergence criterion for the number

7

of significant digits NSIG, the epsilon convergence criterion EPS, the delta covergence criterion DELTA, the maximum number of function evaluations MAXFN, the input options parameter IOPT, and the input parameters PARM. These records may appear in arbitrary order within the record set.

The number of significant digits convergence criterion is satisfied if, on two consecutive iterations, every component of the state vector agrees with the previous iteration to the specified number of digits. The epsilon criterion is satisfied if, on two consecutive iterations, the difference between the sum of the squares of the residuals from the two iterations is less than the specified value. The delta criterion is satisfied if the Euclidean norm of the gradient of the state vector is less than or equal to the specified value. Both epsilon and delta must be positive, real numbers. The maximum number of function evaluations defines an upper limit to the number of times that ZXSSQ may call the function evaluation routine. If the maximum number of function evaluations is set to zero, the program will not iterate to a solution. Instead, it will simply perform one function evaluation in order to calculate the residuals corresponding to the initial values of the state vector. If the input options parameter equals 0, Brown's algorithm without steepest descent will be used to find the minimum. If it equals 1, steepest descent and the default values for the input parameters will be used, while a value of 2 allows the user to select the parameters. The first input parameter is the initial value of the Marquardt parameter used to scale the diagonal of the transpose of the Jacobian (default = 0.01). The other three are the scaling factor used to modify the Marquardt parameter (default = 2.0), the upper bound for the Marquardt parameter (default = 120.0), and the value of the norm of the gradient of the sum of the squares of the residuals that determines when central rather than forward differencing is used to calculate the Jacobian (default = 0.1).

2.1.3.1.2. Asymmetric Eigenproblem Parameters .

"Asymmetric Eigenproblem Parameters" is the collection of information that defines the parameters used in the complex eigenvalue/eigenvector solution, and consists of no data or the following ordered set of records:

NUMBER OF COMPLEX EIGEN PARAMETERS = (integer, default = 0)
BEGIN COMPLEX EIGEN PARAMETERS
    "Complex Eigenproblem Parameters Definitions"
END COMPLEX EIGEN PARAMETERS

where "Complex Eigenproblem Parameters Definitions" consist of a specified number of the following set of records in arbitrary order:

BEGIN COMPLEX EIGEN DEFINITION = (8 character literal)
    NVAL = (integer)
    NVEC = (integer, default = 0)
    CALPRI = ('TRUE' or 'FALSE', default = 'FALSE')
END COMPLEX EIGEN DEFINITION

The complex eigenproblem parameters identifier corresponds to the AEPPAR entry for the SOLVE ASYMMETRIC EIGEN-PROBLEM directive under "Control Data" (section 2.1.2). The definition consists of the number of eigenvalues to be calculated NVAL, the number of eigenvectors

to be calculated NVEC, and the calculate performance index flag CALPRI. These records may appear in arbitrary order within the record set.

2.1.3.2. Model Identifier Definitions

"Model Identifier Definitions" is the collection of information defining the field and system organization identifiers for each model to be analyzed. "Model Identifier Definitions" consists of no data or the following ordered set of records:

```
NUMBER OF MODELS = (integer, default = 0)
BEGIN MODELS
    "Model Definition"
END MODELS
```

where "Model Definition" consists of the specified number of the following set of records in arbitrary order:

```
BEGIN MODEL DEFINITION = (8 character literal)
    AIR DENSITY IDENTIFIER = (8 character literal)
    GRAVITY FIELD IDENTIFIER = (8 character literal)
    INERTIAL MOTION IDENTIFIER = (8 character literal)
    SYSTEM ORGANIZATION IDENTIFIER = (8 character literal)
END MODEL DEFINITION
```

The model identifier corresponds to one of the MDLID entries defined under the directives in "Control Data" (section 2.1.2). The identifiers for air density, gravity, and inertial motion point to the BEGIN AIR DENSITY DEFINITION, BEGIN GRAVITY DEFINITION, and BEGIN INERTIAL MOTION DEFINITION entries respectively, under "Field Parameters" (section 2.1.3.3.4); while the identifier for system organization points to the BEGIN SYSTEM DEFINITION entry under "System Organization" (section 2.1.3.3.1a). These records may appear in arbitrary order within this record set.

2.1.3.3. Model Parameter Definitions

"Model Parameter Definitions" is the collection of information defining all of the parameters that are needed to completely describe every model that will be be analyzed. "Model Parameter Definitions" consists of no data or the following ordered set of records:

```
BEGIN MODEL PARAMETERS
    "Model Parameters"
END MODEL PARAMETERS
```

"Model Parameters" consists of one or more of the following sets of records in arbitrary order:

```
"System Parameters" (see section 2.1.3.3.1)
"Element Parameters" (see section 2.1.3.3.2)
"Array Data" (see section 2.1.3.3.3)
"Field Parameters" (see section 2.1.3.3.4)
```

2.1.3.3.1. · System Parameters .

"System Parameters" is the collection of information defining the system organization, the subsystems and the constraints on the subsystems. "System Parameters" consists of no data or the ordered set of records:

> BEGIN SYSTEM PARAMETERS
>> "System Parameters Records"
> END SYSTEM PARAMETERS

where "System Parameters Records" consists of no data or one or more of the following sets of records in arbitrary order:

> "System Organization"
> "Structural Node Definitions"
> "Air Node Definitions"
> "Subsystem Constraint Sets"
> "Aeroelastic Beam Connectivity"
> "Air Mass Connectivity"
> "Fixed Frame Constraints"
> "Periodic Frame Constraints"
> "Periodic Structure Constraints"
> "Prescribed Motion Constraints"
> "Rigid-body Connection Constraints"
> "Rigid-body Mass Connectivity"
> "Rotating Frame Constraints"
> "Rotating Structure Constraints"
> "Screw Connection Constraints"

a.) "System Organization" consists of no data or the ordered set of records:

> NUMBER OF SYSTEMS = (integer, default = 0)
> BEGIN SYSTEM ORGANIZATION DATA
>> "System Definition"
> END SYSTEM ORGANIZATION DATA

where the "System Definition" consists of the specified number of the following set of records in arbitrary order:

> BEGIN SYSTEM DEFINITION = (8 character literal)
>> NUMBER OF SUB-SYSTEMS = (integer, default = 0)
>> BEGIN DEFINITIONS OF SUB-SYSTEMS
>>> "Subsystem Definitions"
>> END DEFINITIONS OF SUB-SYSTEMS
> END SYSTEM DEFINITION

The system identifier corresponds to the SYSTEM ORGANIZATION IDENTIFIER entry under "Model Definition" (section 2.1.3.2). "Subsystems Definitions" consists of the specified number of the following set of records *arranged in preorder.*

```
BEGIN SUB-SYSTEM DEFINITION
     PSSNAM = (8 character literal)
     SBSNAM = (8 character literal)
     SBSTYP = ('AEBEAM', 'AIRMAS', 'RBMASS', or 'SYSTEM')
     SBSID = (8 character literal, default = ' ')
     CNSSID = (8 character literal)
END SUB-SYSTEM DEFINITION
```

The definition of a subsystem consists of the parent subsystem name PSSNAM, the subsystem name SBSNAM which is the name of this subsystem, the subsystem type SBSTYP, the subsystem identifier SBSID, and the subsystem constraint set identifier CNSSID. These records may appear in arbitrary order within a record set.

The parent subsystem name refers to the SBSNAM entry in the definition of the parent subsystem. For the first subsystem in a system definition, the parent subsystem name may be defined as PSSNAM = ' ', since the name of the parent of the first subsystem is automatically defined to be the same as the MDLNAM entry from the directive associated with that model (section 2.1.2). The subsystem type identifies the type of subsystem being defined. If SBSTYP = 'AEBEAM', the subsystem identifier points to the BEGIN AE BEAM DEFINITION entry under "Aeroelastic Beam Definitions" (section 2.1.3.3.2a). If SBSTYP = 'AIRMAS', the subsystem identifier points to the BEGIN AIR MASS DEFINITION entry under "Air Mass Definitions" (section 2.1.3.3.2h). If SBSTYP = 'RBMASS', the subsystem identifier points to the BEGIN RB MASS DEFINITION entry under "Rigid-body Mass Definitions" (section 2.1.3.3.2j). If SBSTYP = 'SYSTEM', the subsystem identifier points to a BEGIN SS NODE DEFINITION SET entry under "Node Definitions for Subsystem" (section 2.1.3.3.1b) and to an ANDNAM entry under "Definition of Air Node i" (section 2.1.3.3.1c). The constraint set identifier points to a BEGIN SUB-SYSTEM CONSTRAINT SET entry under "Subsystem Constraint Sets" (section 2.1.3.3.1d).

If the subsystem identifier is defaulted, GRASP will construct its system model assuming that there are no structural nodes or air nodes defined for the subsystem. This default is only valid for system-type subsystems.

b.) "Structural Node Definitions" consists of the following ordered set of records:
```
NUMBER OF SS NODDEF SETS = (integer, default = 0)
BEGIN NODE DEFINITIONS FOR SS
     "Node Definitions for Subsystem"
END NODE DEFINITIONS FOR SS
```

where "Node Definitions for Subsystems" consists of the specified number of the following ordered set of records in arbitrary order:
```
BEGIN SS NODE DEFINITION SET = (8 character literal)
     LENGTH OF SS NODE DEFINITION SET = (integer, default = 0)
     BEGIN SUB-SYSTEM NODE DEFINITIONS
          "Definition of Subsystem Node i"
     END SUB-SYSTEM NODE DEFINITIONS
```

END SS NODE DEFINITION SET

The node definition set identifier corresponds to the SBSID entry under "System Organization" (section 2.1.3.3.1a) when the subsystem is a system-type subsystem. "Definition of Subsystem Node i" consists of the specified number of the following record set in arbitrary order:

        BEGIN DEFINITION OF SS NODE
            SSNNAM = (8 character literal)
            NDPSID = (8 character literal, default = ' ')
            NDDCID = (8 character literal, default = ' ')
        END DEFINITION OF SS NODE

The definition of a subsystem node consists of the subsystem node name SSNNAM; the identifier for R_F^{NF}, the position of the node relative to the subsystem frame in the frame basis NDPSID; and the identifier for C^{NF}, the orientation of the node relative to the subsystem frame NDDCID. The identifiers NDPSID and NDDCID point to the BEGIN VECTOR DEFINITION and BEGIN DIRCOS DEFINITION entries, respectively, in "Array Data" (section 2.1.3.3.3).

If the node position identifier is defaulted, GRASP assumes that the position vector is zero. If the orientation identifier is defaulted, GRASP assumes that the direction cosine array is identity.

c.) "Air Node Definitions" consists of the following ordered set of records:

        NUMBER OF SS AIR NODE SETS = (integer, default = 0)
        BEGIN SS AIR NODE DEFINITION SETS
            "Subsystem Air Node Definition Sets"
        END SS AIR NODE DEFINITION SETS

where "Subsystem Air Node Definition Sets" consists of the specified number of the following ordered set of records in arbitrary order:

        BEGIN DEFINITION OF SS AIR NODE SET = (8 character literal)
            LENGTH OF SS AIR NODE SET = (integer, default = 0)
            BEGIN DEFINITIONS OF AIR NODES
                "Definition of Air Node i"
            END DEFINITIONS OF AIR NODES
        END DEFINITION OF SS AIR NODE SET

The subsystem air node set identifier corresponds to the SBSID entry under "System Organization" (section 2.1.3.3.1a) when the subsystem is a system-type subsystem. "Definition of Air Node i" consists of the specified number of the following record set in arbitrary order:

        BEGIN AIR NODE DEFINITION
            ANDNAM = (8 character literal)
            OFFSID = (8 character literal, default = ' ')
            DRCSID = (8 character literal, default = ' ')
        END AIR NODE DEFINITION

The definition of an air node consists of the air node name ANDNAM; the identifier for R_F^{AF}, the position of the flow-field axis of symmetry relative to the subsystem frame in the frame basis OFFSID; and the identifier for C^{AF}, the orientation of the axis of symmetry relative to the

12

subsystem frame DRCSID. The identifiers OFFSID and DRCSID point to the BEGIN VECTOR DEFINITION and BEGIN DIRCOS DEFINITION entries, respectively, in "Array Data" (section 2.1.3.3.3).

If the air node position identifier is defaulted, GRASP assumes that the position vector is zero. If the direction cosines identifier is defaulted, GRASP assumes that the direction osine array is identity.

d.) "Subsystem Constraint Sets" consists of the ordered set of records:

NUMBER OF SS CONSTRAINT SETS = (integer, default = 0)
BEGIN CONSTRAINTS FOR SUB-SYSTEMS
    "Subsystem Constraint Set"
END CONSTRAINTS FOR SUB-SYSTEMS

where "Subsystem Contraint Set" consists of the specified number of the following ordered set of records:

BEGIN SUB-SYSTEM CONSTRAINT SET = (8 character literal)
    LENGTH OF SS CONSTRAINT SET = (integer, default = 0)
    BEGIN SS CONSTRAINT DEFINITIONS
        "Definition of Subsystem Constraint"
    END SS CONSTRAINT DEFINITIONS
END SUB-SYSTEM CONSTRAINT SET

The subsystem constraint set identifier corresponds to the CNSSID entry under "System Organization" (section 2.1.3.3.1a). The "Definition of Subsystem Constraint" consists of the specified number of the following set of records in arbitrary order:

BEGIN DEFINITION OF SS CONSTRAINT
    CNSTYP = (8 character literal)
    CNSID = (8 character literal)
END DEFINITION OF SS CONSTRAINT

The definition of a subsystem constraint consists of the constraint type CNSTYP and the constraint identifier CNSID. The constraint type may be any one of the following:

'ABC' - aeroelastic beam connectivity
'AMC' - air mass connectivity
'FFR' - fixed frame constraint
'PFR' - periodic frame constraint
'PRE' - prescribed constraint
'PST' - periodic structure constraint
'RBC' - rigid-body connection constraint
'RFR' - rotating frame constraint
'RST' - rotating structure constraint
'RMC' - rigid-body mass connectivity
'SCR' - screw constraint

while the constraint identifier points to the BEGIN xxx CONSTRAINT DEFINITION entry under one of the constraint specifications (sections 2.1.3.3.1e-o).

e.) "Aeroelastic Beam Connectivity" consists of the following ordered set of records:

LENGTH OF AEB CONNECTIVITY TABLE = (integer, default = 0)
BEGIN AEB CONNECTIVITY TABLE
    "Aeroelastic Beam Connectivity Definitions"
END AEB CONNECTIVITY TABLE

where "Aeroelastic Beam Connectivity Definitions" consists of a specified number of the following set of records in arbitrary order:

BEGIN ABC DEFINITION = (8 character literal)
    RSSNAM = (8 character literal, default = ' ')
    RNDNAM = (8 character literal)
    TSSNAM = (8 character literal, default = ' ')
    TNDNAM = (8 character literal)
    ASSNAM = (8 character literal, default = ' ')
    ANDNAM = (8 character literal, default = ' ')
    RTOFID = (8 character literal, default = ' ')
    RTDCID = (8 character literal, default = ' ')
END ABC DEFINITION

The aeroelastic beam connectivity identifier corresponds to the CNSID entry under "Subsystem Constraint Sets" (section 2.1.3.3.1d), when CNSTYP = 'ABC'. The definition consists of the root node subsystem name RSSNAM, the root node name RNDNAM, the tip node subsystem name TSSNAM, the tip node name TNDNAM, the air node node subsystem name ASSNAM, the air node name ANDNAM, the identifier for R_N^{RN}, the position of the element root relative to the root node in the root node basis RTOFID, and the identifier for C^{RN}, the orientation of the element root relative to the root node RTDCID. These records may appear in arbitrary order within the record set.

The root, tip, and air node subsystem names point to SBSNAM entries under "System Organization" (section 2.1.3.3.1a), while the root and tip node names point to SSNNAM entries under "Definition of Subsystem Node i" (section 2.1.3.3.1b). The air node name points to an ANDNAM entry under "Definition of Air Node i" (section 2.1.3.3.1c). The position and orientation identifiers point to the BEGIN VECTOR DEFINITION and the BEGIN DIRCOS DEFINITION entries, respectively, under "Array Data" (section 2.1.3.3.3).

The root, tip, and air nodes may be defined in any subsystem that precedes the aeroelastic beam element subsystem in preorder. If the subsystem name for any of the nodes is defaulted in the aeroelastic beam connectivity definition, GRASP assumes that the parent subsystem of the aeroelastic beam element is the subsystem in which that node is defined. If the air node name, as well as the air node subsystem name, is defaulted, GRASP will constrain the air node degrees of freedom for the beam. If the root node position identifier is defaulted, GRASP assumes that the position vector is zero. If the root node orientation identifier is defaulted, GRASP assumes that the direction cosine array is identity.

f.) "Air Mass Connectivity" consists of the following ordered set of records:

LENGTH OF AMS CONNECTIVITY TABLE = (integer, default = 0)
BEGIN AMS CONNECTIVITY TABLE
 "Air Mass Connectivity Definitions"
END AMS CONNECTIVITY TABLE

where "Air Mass Connectivity Definitions" consists of the specified number of the following set of records in arbitrary order:

BEGIN AMC DEFINITION = (8 character literal)
 SBSNAM = (8 character literal, default = ' ')
 ANDNAM = (8 character literal)
END AMC DEFINITION

The air mass connectivity identifier corresponds to the CNSID entry under "Subsystem Constraint Sets" (section 2.1.3.3.1d), when CNSTYP = 'AMC'. The definition consists of the air node subsystem name SBSNAM and the air node name ANDNAM. These records may appear in arbitrary order within the record set.

The subsystem name points to an SBSNAM entry under "System Organization" (section 2.1.3.3.1a), while the air node name points to an ANDNAM entry under "Definition of Air Node i" (section 2.1.3.3.1c).

The air node may be defined in any subsystem that precedes the air mass element subsystem in preorder. If the subsystem name is defaulted in the air mass connectivity definition, GRASP assumes that the parent subsystem of the air mass element is the subsystem in which the air node is defined.

g.) "Fixed Frame Constraint" consists of the ordered set of records:

LENGTH OF FFR CONSTRAINT TABLE = (integer, default = 0)
BEGIN FFR CONSTRAINT TABLE
 "Fixed Frame Constraint Definitions"
END FFR CONSTRAINT TABLE

where "Fixed Frame Constraint Definitions" consists of the specified number of the following set of records in arbitrary order:

BEGIN FFR CONSTRAINT DEFINITION = (8 character literal)
 FROFID = (8 character literal, default = ' ')
 FRDCID = (8 character literal, default = ' ')
END FFR CONSTRAINT DEFINITION

The fixed frame constraint identifier corresponds to the CNSID entry under "Subsystem Constraint Sets" (section 2.1.3.3.1d), when CNSTYP = 'FFR'. The definition consists of the identifier for R_S^{FS}, the position of the current frame relative to the superordinate frame in the superordinate frame basis FROFID; and the identifier for C^{FS}, the orientation of the current frame relative to the superordinate frame in the superordinate frame basis FRDCID. The position and orientation identifiers point to the BEGIN VECTOR DEFINITION and BEGIN DIRCOS DEFINITION

entries, respectively, under "Array Data" (section 2.1.3.3.3) and may appear in arbitrary order within the record set.

If the frame position identifier is defaulted, GRASP assumes that the position vector is zero. If the orientation identifier is defaulted, GRASP assumes that the direction cosine array is identity.

h.) "Periodic Frame Constraint" consists of the ordered set of records:

LENGTH OF PFR CONSTRAINT TABLE = (integer, default = 0)
BEGIN PFR CONSTRAINT TABLE
    "Periodic Frame Constraint Definitions"
END PFR CONSTRAINT TABLE

where "Periodic Frame Constraint Definitions" consists of the specified number of the following set of records in arbitrary order:

BEGIN PFR CONSTRAINT DEFINITION = (8 character literal)
    N = (integer)
    FROFID = (8 character literal, default = ' ')
    FRDCID = (8 character literal, default = ' ')
END PFR CONSTRAINT DEFINITION

The periodic frame constraint identifier corresponds to the CNSID entry under "Subsystem Constraint Sets" (section 2.1.3.3.1d), when CNSTYP = 'PFR'. The definition consists of the number of subsystems N that are equally spaced around the axis of symmetry; the identifier for R_S^{FS}, the position of the current frame from the superordinate frame in the superordinate frame basis FROFID; and the identifier for C^{FS}, the orientation of the current frame relative to the superordinate frame in the superordinate frame basis FRDCID. The position and orientation identifiers point to the BEGIN VECTOR DEFINITION and BEGIN DIRCOS DEFINITION entries, respectively, under "Array Data" (section 2.1.3.3.3) and may appear in arbitrary order within the record set.

If the frame position identifier is defaulted, GRASP assumes that the position vector is zero. If the orientation identifier is defaulted, GRASP assumes that the direction cosine array is identity. For this constraint, it is assumed that the axis of symmetry is aligned with the 1-axis of the parent frame.

i.) "Periodic Structure Constraint" consists of the ordered set of records:

LENGTH OF PST CONSTRAINT TABLE = (integer, default = 0)
BEGIN PST CONSTRAINT TABLE
    "Periodic Structure Constraint Definitions"
END PST CONSTRAINT TABLE

where "Periodic Structure Constraint Definitions" consists of the specified number of the following set of records in arbitrary order:

```
BEGIN PST CONSTRAINT DEFINITION = (8 character literal)
    ISSNAM = (8 character literal, default = ' ')
    INDNAM = (8 character literal)
    DSSNAM = (8 character literal, default = ' ')
    DNDNAM = (8 character literal)
END PST CONSTRAINT DEFINITION
```

The periodic structure constraint identifier corresponds to the CNSID entry under "Subsystem Constraint Sets" (section 2.1.3.3.1d), when CNSTYP = 'PST'. The definition consists of the independent (periodic) subsystem name ISSNAM, the independent node name INDNAM, the dependent (component) subsystem name DSSNAM, and the dependent node name DNDNAM. These records may appear in arbitrary order within the record set.

If the independent subsystem name is defaulted, GRASP assumes that the independent node is defined in the parent subsystem, while if the dependent subsystem name is defaulted, GRASP assumes that the dependent node is defined in the subsystem where the constraint is defined. For this constraint, it is assumed that the axis of symmetry is aligned with the 1-axis of the independent node.

j.) "Prescribed Constraint" consists of the ordered set of records:

```
LENGTH OF PRE CONSTRAINT TABLE = (integer, default = 0)
BEGIN PRE CONSTRAINT TABLE
    "Prescribed Constraint Definitions"
END PRE CONSTRAINT TABLE
```

where "Prescribed Constraint Definitions" consists of a specified number of the following set of records in arbitrary order:

```
BEGIN PRE CONSTRAINT DEFINITION = (8 character literal)
    DOFTYP = ('AIR' or 'STR', default = 'STR')
    DOFSET = (8 character literal)
    DOFNUM = (integer)
    DOFVAL = (real, default = 0.0)
END PRE CONSTRAINT DEFINITION
```

The prescribed constraint identifier corresponds to the CNSID entry under "Subsystem Constraint Sets" (section 2.1.3.3.1d), when CNSTYP = 'PRE'. The definition consists of the degree of freedom type DOFTYP, the degree of freedom set name DOFSET, the degree of freedom number DOFNUM, and the prescribed value of that degree of freedom DOFVAL; all of which may appear in arbitrary order within the record set.

The degree of freedom being constrained may be either an air or structural node degree of freedom. If an air node degree of freedom is being constrained, the degree of freedom set name points to an ANDNAM entry under "Definition of Air Node i" (section 2.1.3.3.1c), while constraining a structural node degree of freedom results in having the set name point to a SSNNAM entry under "Definition of Subsystem Node i" (section 2.1.3.3.1b). The degree of freedom number may take on values from 1 to 4 for air degrees of freedom and values from 1 to 6 for structural degrees of freedom. Air degree of freedom 1 is the static uniform inflow or the

dynamic collective inflow mode. Air degree of freedom 2 is the static inflow gradient, while degrees of freedom 3 and 4 are the dynamic cosine and sine cyclic inflow modes, respectively. Structural degrees of freedom 1, 2, and 3 are translations along the 1-, 2-, and 3-axes, respectively; while degrees of freedom 4, 5, and 6 are rotations about the 1-, 2-, and 3-axes, respectively.

If the degree of freedom type is defaulted, GRASP assumes that it is structural. For structural translation degrees of freedom, the degree of freedom value is in units of length, while the structural rotations are expressed in terms of Euler-Rodrigues parameters. The air node static uniform inflow is expressed in units of velocity, while the other air node degrees of freedom are in units of velocity per unit length. The default value of any prescribed degree of freedom is zero.

Version 1.0 Note: Because of the way in which the Rotating Frame constraint was implemented, Prescribed constraints defined in subsystems that are descendants of subsystems having Rotating Frame constraints will give erroneous results in the asymmetric eigenproblem. Nodal degrees of freedom that must be eliminated should be constrained using the Rigid-Body or Screw Connection constraints.

k.) "Rigid-body Connection Constraint" consists of the following ordered set of records:

LENGTH OF RBC CONSTRAINT TABLE = (integer, default = 0)
BEGIN RBC CONSTRAINT TABLE
    "Rigid-body Connection Constraint Definitions"
END RBC CONSTRAINT TABLE

where "Rigid-body Connection Constraint Definitions" consists of the specified number of the following set of records in arbitrary order:

BEGIN RBC CONSTRAINT DEFINITION = (8 character literal)
    ISSNAM = (8 character literal, default = ' ')
    INDNAM = (8 character literal)
    DNDNAM = (8 character literal)
END RBC CONSTRAINT DEFINITION

The rigid-body connection constraint identifier corresponds to the CNSID entry under "Subsystem Constraint Sets" (section 2.1.3.3.1d), when CNSTYP = 'RBC'. The definition consists of the independent subsystem name ISSNAM, the independent node name INDNAM, and the dependent node name DNDNAM. These records may appear in arbitrary order within the record set.

The independent subsystem name points to an SBSNAM entry under "System Organization" (section 2.1.3.3.1a), and the independent and dependent node names point to SSNNAM entries under "Definition of Subsystem Node i" (section 2.1.3.3.1b).

If the independent subsystem name is defaulted, GRASP assumes that the independent node is defined in the subsystem for which the constraint is specified. The dependent node must be defined in the same subsystem as the constraint.

l.) "Rigid-body Mass Connectivity" consists of the following ordered set of records:

18

LENGTH OF RBM CONNECTIVITY TABLE = (integer, default = 0)
BEGIN RBM CONNECTIVITY TABLE
    "Rigid-body Mass Connectivity Definitions"
END RBM CONNECTIVITY TABLE

where "Rigid-body Mass Connectivity Definitions" consists of a specified number of the following set of records:

BEGIN RMC DEFINITION = (8 character literal)
    SBSNAM = (8 character literal, default = ' ')
    NODNAM = (8 character literal)
    OFFSID = (8 character literal, default = ' ')
    DRCSID = (8 character literal, default = ' ')
END RMC DEFINITION

The rigid-body mass connectivity identifier corresponds to the CNSID entry under "Subsystem Constraint Sets" (section 2.1.3.3.1d), when CNSTYP = 'RMC'. The definition consists of the center-of-mass node subsystem name SBSNAM, the center-of-mass node name NODNAM, the identifier for R_N^{CN}, the position of the element mass center relative to the rigid-body node in the rigid-body node basis OFFSID, and the identifier for C^{CN}, the orientation of the element principal axes relative to the rigid-body node axes DRCSID.

The subsystem name points to an SBSNAM entry under "Subsystem Definition" (section 2.1.3.3.1a). The node name points to an SSNNAM entry under "Definition of Subsystem Node i" (section 2.1.3.3.1b). The identifiers for position and orientation point to BEGIN VECTOR DEFINITION and BEGIN DIRCOS DEFINITION entries, respectively, under "Array Data" (section 2.1.3.3.3).

The center-of-mass node may be defined in any subsystem that precedes the rigid-body mass element in preorder. If the center-of-mass subsystem name is defaulted, GRASP assumes that the parent subsystem of the rigid-body mass element is the subsystem in which the node is defined. If the node position identifier is defaulted, GRASP assumes that the position vector is zero. If the orientation identifier is defaulted, GRASP assumes that the direction cosine array is identity.

m.) "Rotating Frame Constraint" consists of the ordered set of records:

LENGTH OF RFR CONSTRAINT TABLE = (integer, default = 0)
BEGIN RFR CONSTRAINT TABLE
    "Rotating Frame Constraint Definitions"
END RFR CONSTRAINT TABLE

where "Rotating Frame Constraint Definitions" consists of the specified number of the following set of records in arbitrary order:

BEGIN RFR CONSTRAINT DEFINITION = (8 character literal)
    OMEGA = (real)
    FROFID = (8 character literal, default = ' ')
    FRDCID = (8 character literal, default = ' ')
END RFR CONSTRAINT DEFINITION

The rotating frame constraint identifier corresponds to the CNSID entry under "Subsystem Constraint Sets" (section 2.1.3.3.1d), when CNSTYP = 'RFR'. The definition consists of the angular speed OMEGA (in rad/sec) of the rotating frame relative to the nonrotating frame; the identifier for R_N^{RN}, the position of the rotating frame relative to the nonrotating frame in the nonrotating frame basis FROFID; and the identifier for C^{RN}, the orientation of the rotating frame relative to the nonrotating frame FRDCID. The position and orientation identifiers point to the BEGIN VECTOR DEFINITION and BEGIN DIRCOS DEFINITION entries, respectively, under "Array Data" (section 2.1.3.3.3) and may appear in arbitrary order within the record set.

If the frame position identifier is defaulted, GRASP assumes that the position vector is zero. If the orientation identifier is defaulted, GRASP assumes that the direction cosine array is identity. For this constraint, it is assumed that the axis of rotation is aligned with the 1-axis of the child frame.

n.) "Rotating Structure Constraint" consists of the ordered set of records:

LENGTH OF RST CONSTRAINT TABLE = (integer, default = 0)
BEGIN RST CONSTRAINT TABLE
 "Rotating Structure Constraint Definitions"
END RST CONSTRAINT TABLE

where "Rotating Structure Constraint Definitions" consists of the specified number of the following set of records in arbitrary order:

BEGIN RST CONSTRAINT DEFINITION = (8 character literal)
 ISSNAM = (8 character literal, default = ' ')
 INDNAM = (8 character literal)
 DSSNAM = (8 character literal, default = ' ')
 DNDNAM = (8 character literal)
END RST CONSTRAINT DEFINITION

The rotating structure constraint identifier corresponds to the CNSID entry under "Subsystem Constraint Sets" (section 2.1.3.3.1d), when CNSTYP = 'RST'. The definition consists of the independent (nonrotating) subsystem name ISSNAM, the independent node name INDNAM, the dependent (rotating) subsystem name DSSNAM, and the dependent node name DNDNAM.

If the independent subsystem name is defaulted, GRASP assumes that the independent node is defined in the parent subsystem, while if the dependent subsystem name is defaulted, GRASP assumes that the dependent node is defined in the subsystem where the constraint is defined.

o.) "Screw Connection Constraint" consists of the following ordered set of records:

LENGTH OF SCR CONSTRAINT TABLE = (integer, default = 0)
BEGIN SCR CONSTRAINT TABLE
 "Screw Connection Constraint Definitions"
END SCR CONSTRAINT TABLE

where "Screw Connection Constraint Definitions" consists of the specified number of the following set of records in arbitrary order:

```
BEGIN SCR CONSTRAINT DEFINITION = (8 character literal)
    ISSNAM = (8 character literal, default = ' ')
    INDNAM = (8 character literal)
    DNDNAM = (8 character literal)
    SCOFID = (8 character literal, default = ' ')
    SCAUVI = (8 character literal)
    SCRROT = ('FIX' or 'FREE', default = 'FREE')
    SCRTRA = ('FIX' or 'FREE', default = 'FREE')
    CROT = (real, default = 0.0)
    CTRA = (real, default = 0.0)
    KROT = (real, default = 0.0)
    KTRA = (real, default = 0.0)
    T0 = (real, default = 0.0)
    U0 = (real, default = 0.0)
END SCR CONSTRAINT DEFINITION
```

The screw connection constraint identifier corresponds to the CNSID entry under "Subsystem Constraint Sets" (section 2.1.3.3.1d), when CNSTYP = 'SCR'. The definition consists of the independent subsystem name ISSNAM; the independent node name INDNAM; the dependent node name DNDNAM; the identifier for R_I^{SI}, the position of the screw axis relative to the independent node in the independent node basis SCOFID; the identifier for e_I^{SCR}, the screw axis unit vector in the independent node basis SCAUVI; the screw rotation flag SCRROT; the screw translation flag SCRTRA; the screw rotation damping constant CROT; the screw translation damping constant CTRA; the screw rotation stiffness constant KROT; the screw translation stiffness constant KTRA; the screw rotation offset T0; and the screw translation offset U0. These records may appear in arbitrary order within the record set.

The independent subsystem name points to an SBSNAM entry under "System Organization" (section 2.1.3.3.1a), and the independent and dependent node names point to BEGIN DEFINI-TION OF SS NODE entries under "Node Definitions" (section 2.1.3.3.1b). The position and unit vector identifiers point to BEGIN VECTOR DEFINITION entries under "Array Data' (section 2.1.3.3.3b). The rotation and translation flags define whether or not the dependent node is free to rotate (in a right-handed sense) about and translate along the screw axis, with respect to the independent node. The stiffness and damping constants for the screw rotation and translation provide a means for including compliance in the screw motion when either or both of the screw degrees of freedom are free. The screw rotation and translation offsets are used to prescribe a constant deformation for either or both of the screw degrees of freedom when the degree(s) of freedom is fixed. All of the screw rotation constants (damping, stiffness, and offset) are defined using angular units of degrees.

If the independent subsystem name is defaulted, GRASP assumes that the independent node is defined in the current subsystem. The dependent node must be defined in the same subsystem as the constraint. The default value of the position identifier indicates that the position vector is zero. The default values of both the screw translation and rotation flags indicate that those degrees of freedom are not constrained.

Version 1.0 Note: Due to the way in which the Screw constraint was implemented, a Screw constraint may not be defined in a subsystem that contains a Periodic Frame constraint. An error message will be issued, and execution will be terminated.

2.1.3.3.2. Element Parameters .

"Element Parameters" is the collection of information defining the parameters that describe the elements in the model. "Element Parameters" consists of no data or the ordered set of records:

```
BEGIN ELEMENT PARAMETERS
    "Element Parameter Records"
END ELEMENT PARAMETERS
```

where "Element Parameter Records" consists of no data or one or more of the following sets of records in arbitrary order:

"Aeroelastic Beam Definitions"
"Aeroelastic Beam Aerodynamic Section Properties"
"Aeroelastic Beam Aerodynamic Coefficient Tables"
"Aeroelastic Beam Elastic Section Properties"
"Aeroelastic Beam Geometric Properties"
"Aeroelastic Beam Inertial Section Properties"
"Aeroelastic Beam Orders Definitions"
"Air Mass Definitions"
"Air Mass Property Definitions"
"Rigid-body Mass Definitions"

a.) "Aeroelastic Beam Definitions" has the form:

```
NUMBER OF AE BEAM DEFINITIONS = (integer, default = 0)
BEGIN DEFINITIONS OF AE BEAMS
    "Aeroelastic Beam Definition"
END DEFINITIONS OF AE BEAMS
```

where "Aeroelastic Beam Definition" consists of the specified number of the following set of records in arbitrary order:

```
BEGIN AE BEAM DEFINITION = (8 character literal)
    ASPID = (8 character literal, default = ' ')
    ESPID = (8 character literal)
    GEOMID = (8 character literal)
    ISPID = (8 character literal)
    ORDID = (8 character literal)
    ZXPID = (8 character literal, default = ' ')
END AE BEAM DEFINITION
```

The aeroelastic beam identifier corresponds to the SBSID entry under "System Organization" (section 2.1.3.3.1a). The aerodynamic properties identifier ASPID points to a BEGIN AS PROPERTIES DEFINITION entry (section 2.1.3.3.2b); the elastic section properties identifier

ESPID points to a BEGIN ES PROPERTY DEFINITION entry (section 2.1.3.3.2d); the geometric properties identifier GEOMID points to a BEGIN GEOM PROPERTY DEFINITION entry (section 2.1.3.3.2e); the inertial section properties identifier ISPID points to a BEGIN IS PROPERTY DEFINITION entry (section 2.1.3.3.2f); the beam orders identifier ORDID points to a BEGIN AEB ORDERS DEFINITION entry (section 2.1.3.3.2g); and the ZXSSQ parameters identifier ZXPID points to a BEGIN ZXSSQ DEFINITION entry (section 2.1.3.1.1). These identifiers may appear in arbitrary order within the record set.

If the aerodynamic properties identifier is defaulted, GRASP will not calculate the aerodynamic forces on the element. If the ZXSSQ parameters identifier is defaulted, the parameters take on the default values specified under the BEGIN ZXSSQ DEFINITION entry (section 2.1.3.1.1).

b.) "Aeroelastic Beam Aerodynamic Section Properties" has the form:

NUMBER OF ASP DEFINITIONS = (integer, default = 0)
BEGIN DEFINITIONS OF AS PROPERTIES
    "Aeroelastic Beam Aerodynamic Section Properties Definition"
END DEFINITIONS OF AS PROPERTIES

where "Aeroelastic Beam Aerodynamic Section Properties Definition" consists of the specified number of the following set of records in arbitrary order:

BEGIN AS PROPERTY DEFINITION = (8 character literal)
    SGNAA = (real, default = +1.)
    CDTID = (8 character literal)
    CLTID = (8 character literal)
    CMTID = (8 character literal)
    CDX = (real array of length 4, default = 1.0, 1.0, 1.0, 1.0)
    CLX = (real array of length 4, default = 1.0, 1.0, 1.0, 1.0)
    CMX = (real array of length 4, default = 1.0, 1.0, 1.0, 1.0)
    TZL = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
    CHD = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
    RQPZ2 = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
END AS PROPERTY DEFINITION

The aerodynamic section property identifier corresponds to the ASPID entry under "Aeroelastic Beam Definitions" (section 2.1.3.3.2a). A value of +1 for the sign of the angle of attack (SGNAA) indicates normal flow, -1 indicates reversed flow, and 0 indicates no flow. The coefficient of drag, lift, and moment table identifiers CDTID, CLTID, and CMTID point to the BEGIN AC TABLE DEFINITION entries (section 2.1.3.3.2c). The values for the lengthwise scale factor variations for the drag coefficient CDX, the lift coefficient CLX, and the moment coefficient CMX; and the values for theta (in degrees) of the zero-lift line TZL, the chord CHD, and the position (R_{Z2}^{QP}) of the aerodynamic center from the elastic axis RQPZ2 are specified at $x_3 = 0, \frac{l}{3}, \frac{2l}{3}, l$ to allow for up to a cubic spanwise variation of the aerodynamic section properties.

If SGNAA is defaulted, GRASP will calculate the aerodynamic forces on the element assuming normal flow. If the values for CDX, CLX, or CMX are defaulted, the associated aerodynamic coefficients will be held constant over the entire element length.

23

c.) "Aeroelastic Beam Aerodynamic Coefficient Table" has the form:

NUMBER OF AC TABLE DEFINITIONS = (integer, default = 0)
BEGIN DEFINITIONS OF AC TABLES
    "Aerodynamic Coefficient Table Definition"
END DEFINITIONS OF AC TABLES

where "Aerodynamic Coefficient Table Definition" consists of a specified number of the following ordered set of records in arbitrary order:

BEGIN AC TABLE DEFINITION = (8 character literal)
    LENGTH OF AC TABLE = (integer, default = 0)
    BEGIN AC TABLE ENTRIES
        "Aerodynamic Coefficient Table Entry i"
    END AC TABLE ENTRIES
END AC TABLE DEFINITION

The aerodynamic coefficient table identifier corresponds to the CDTID, CLTID, and CMTID entries under "Aeroelastic Beam Aerodynamic Section Properties" (section 2.1.3.3.2b), while "Aerodynamic Coefficient Table Entry i" consists of the specified number of the following ordered set of records in table order:

BEGIN ENTRY FOR AC TABLE
    ENTRY = (real array of length 3)
END ENTRY FOR AC TABLE

The first number is a value of angle of attack (in degrees), the second is the lift, drag, or moment coefficient, and the third is the coefficient derivative with respect to the angle of attack (in degrees).

d.) "Aeroelastic Beam Elastic Section Properties" has the form:

NUMBER OF ESP DEFINITIONS = (integer, default = 0)
BEGIN DEFINITIONS OF ES PROPERTIES
    "Aeroelastic Beam Elastic Section Properties Definition"
END DEFINITIONS OF ES PROPERTIES

where "Aeroelastic Beam Elastic Section Properties Definition" consists of the specified number of the following set of records in arbitrary order:

BEGIN ES PROPERTY DEFINITION = (8 character literal)
    E0 = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
    E1 = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
    E2 = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
    I1 = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
    I2 = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
    J = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
    B1 = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
    B2 = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
    B3 = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)

$$D0 = (\text{real array of length 4, default} = 0.0, 0.0, 0.0, 0.0)$$
$$D1 = (\text{real array of length 4, default} = 0.0, 0.0, 0.0, 0.0)$$
$$D2 = (\text{real array of length 4, default} = 0.0, 0.0, 0.0, 0.0)$$
$$D3 = (\text{real array of length 4, default} = 0.0, 0.0, 0.0, 0.0)$$
$$D4 = (\text{real array of length 4, default} = 0.0, 0.0, 0.0, 0.0)$$
END ES PROPERTY DEFINITION

The elastic section properties identifier corresponds to the ESPID entry under "Aeroelastic Beam Definitions" (section 2.1.3.3.2a). Most of the elastic section properties are input at $x_3 = 0, \frac{l}{3}, \frac{2l}{3}, l$ to allow for up to a cubic spanwise variation along the aeroelastic beam element.

The section properties consist of the axial stiffness E0, the first area moment of inertia of axial stiffness about the 1-axis E1, the first area moment of inertia of axial stiffness about the 2-axis E2, the bending stiffness about the 1-axis I1, the bending stiffness about the 2-axis I2, the torsional stiffness J, and the section integrals B1, B2, B3, D0, D1, D2, D3, and D4. The section integrals are defined in Appendix B. Within each set of elastic section properties, the property records may appear in arbitrary order.

e.) "Aeroelastic Beam Geometric Properties" has the form:

NUMBER OF G PROPERTY DEFINITIONS = (integer, default = 0)
BEGIN DEFINITIONS OF G PROPERTIES
 "Aeroelastic Beam Geometric Properties Definition"
END DEFINITIONS OF G PROPERTIES

where "Aeroelastic Beam Geometric Properties Definition" consists of the specified number of the following record set in arbitrary order:

BEGIN GEOM PROPERTY DEFINITION = (8 character literal)
 L = (real)
 THETA = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
END GEOM PROPERTY DEFINITION

The geometric properties identifier corresponds to the GEOMID entry under "Aeroelastic Beam Definitions" (section 2.1.3.3.2a). The geometric properties consist of the element length L and the pretwist angle (in degrees) along the span of the beam THETA. Values of THETA are input at $x_3 = 0, \frac{l}{3}, \frac{2l}{3}, l$ to allow for up to a cubic spanwise variation of the pretwist angle along the aeroelastic beam element. In the beam-element equations, it is assumed that the pretwist angle at the root of the element is zero. Therefore, if the pretwist angle at $x_3 = 0$ is assigned a nonzero value, that value is subtracted from the assigned values at $x_3 = \frac{l}{3}, \frac{2l}{3}, l$ in order to maintain the same relative spanwise pretwist distribution. A warning message is also printed. L and THETA may appear in arbitrary order within the record set.

f.) "Aeroelastic Beam Inertial Section Properties" has the form:

NUMBER OF ISP DEFINITIONS = (integer, default = 0)
BEGIN DEFINITIONS OF IS PROPERTIES
 "Aeroelastic Beam Inertial Section Properties Definition"

25

END DEFINITIONS OF IS PROPERTIES

where "Aeroelastic Beam Inertial Section Properties Definition" consists of the specified number of the following set of records in arbitrary order:

    BEGIN IS PROPERTY DEFINITION = (8 character literal)
        MSS = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
        MM1 = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
        MM2 = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
        MOI1 = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
        MOI2 = (real array of length 4, default = 0.0, 0.0, 0.0, 0.0)
    END IS PROPERTY DEFINITION

The inertial section properties identifier corresponds to the ISPID entry under "Aeroelastic Beam Definition" (section 2.1.3.3.2a). The inertial section properties are input at $x_3 = 0, \frac{l}{3}, \frac{2l}{3}, l$ to allow for up to a cubic spanwise variation along the aeroelastic beam element.

The inertial section properties consist of the mass per unit length MSS, the first area moment of mass about the 1-axis MM1, the first area moment of mass about the 2-axis MM2, the moment of inertia about the 1-axis MOI1, and the moment of inertia about the 2-axis MOI2. These section properties are defined in Appendix B. Within each set of inertial section property records, the property records may appear in arbitrary order.

g.) "Aeroelastic Beam Orders Definition" has the form:

    NUMBER OF ORD DEFINITIONS = (integer, default = 0)
    BEGIN DEFINITIONS OF AEB ORDERS
        "Aeroelastic Beam Orders Definition"
    END DEFINITIONS OF AEB ORDERS

where "Aeroelastic Beam Orders Definition" consists of a specified number of the following set of records in arbitrary order:

    BEGIN AEB ORDER DEFINITION = (8 character literal)
        ORDU1 = (integer, default = 3)
        ORDU2 = (integer, default = 3)
        ORDU3 = (integer, default = 1)
        ORDT3 = (integer, default = 1)
        NGAUSS = (integer, default = 0)
    END AEB ORDER DEFINITION

The beam orders identifier corresponds to the ORDID entry under "Aeroelastic Beam Definition" (section 2.1.3.3.2a). The records define the order of the U1 expansion ORDU1, the order of the U2 expansion ORDU2, the order of the U3 expansion ORDU3, the order of the T3 expansion ORDT3, and the number of Gaussian quadrature points NGAUSS. These records may appear in arbitrary order within the record set.

If the values of ORDU1 or ORDU2 are defaulted, their deformations will be expressed as cubic polynomials. If the values of ORDU3 or ORDT3 are defaulted, their deformations will be expressed as linear polynomials. If NGAUSS is defaulted, the number of Gauss points required

will be calculated from the expression $NGAUSS = \frac{3ORDxx}{2} + 1$, where ORDxx is the largest order of expansion. If the number of Gauss points is specified but is too small, a warning message will be printed and the calculation will continue using the specified number of points.

h.) "Air Mass Definitions" has the form:

```
NUMBER OF AIR MASS DEFINITIONS = (integer, default = 0)
BEGIN DEFINITIONS OF AIR MASSES
    "Air Mass Definition"
END DEFINITIONS OF AIR MASSES
```

where "Air Mass Definition" consists of the specified number of the following set of records:

```
BEGIN AIR MASS DEFINITION = (8 character literal)
    AMSID = (8 character literal)
END AIR MASS DEFINITION
```

The air mass identifier corresponds to the SBSID entry under "System Organization" (section 2.1.3.3.1a). The air mass properties identifier AMSID points to a BEGIN DEFINITIONS OF AMS PROPERTIES entry (section 2.1.3.3.2i).

i.) "Air Mass Properties Definitions" has the form:

```
NUMBER OF AMS PROPERTY DEFINITIONS = (integer, default = 0)
BEGIN DEFINITIONS OF AMS PROPERTIES
    "Air Mass Property Definition"
END DEFINITIONS OF AMS PROPERTIES
```

where "Air Mass Property Definition" consists of the specified number of the following set of records in arbitrary order:

```
BEGIN AMS PROPERTY DEFINITION = (8 character literal)
    TIP RADIUS = (real)
    ROOT CUTOUT RADIUS = (real, default = 0.0)
    NGAUSS = (integer, default = 3)
END AMS PROPERTY DEFINITION
```

The air mass property identifier corresponds to the AMSID entry under "Air Mass Definition" (section 2.1.3.3.2h). The properties definition consists of the tip and root cutout radii, which are measured from the center of the axisymmetric flow field, and the number of Gauss quadrature points to be used for integration. These records may appear in arbitrary order within the record set.

If the root cutout radius is defaulted, the momentum contributions to the inflow will be calculated from the center of the axisymmetric flow field out to the tip radius. The default value for the number of Gauss points is sufficient to integrate the linear inflow distribution, but may be increased if desired.

j.) "Rigid-body Mass Definitions" consists of the following ordered set of records:

NUMBER OF RB MASS DEFINITIONS = (integer, default = 0)
BEGIN DEFINITIONS OF RB MASSES
    "Rigid-body Mass Definition"
END DEFINITIONS OF RB MASSES

where the "Rigid-body Mass Definition" consists of no data or the specified number of the following set of records in arbitrary order:

BEGIN RB MASS DEFINITION = (8 character literal)
    MASS = (real)
    MOI = (real array of length 3)
END RB MASS DEFINITION

The rigid-body mass identifier corresponds to the SBSID entry under "System Organization" (section 2.1.3.3.1a). MASS and MOI are the mass and principal moments of inertia of the rigid body respectively, and may appear in arbitrary order within the record set.

2.1.3.3.3.   Array Data .

"Array Data" is the collection of information defining direction cosine arrays and position vectors. "Array Data" consists of no data or the ordered collection of records:

BEGIN ARRAY DATA
    "Array Definitions"
END ARRAY DATA

where "Array Definitions" consists of no data or one or more of the following sets of records in arbitrary order:

    "Direction Cosines"
    "Vectors"

a.) "Direction Cosines" consists of no data or the ordered set of records:

NUMBER OF DIRCOS DEFINITIONS = (integer, default = 0)
BEGIN DEFINITIONS OF DIRCOS
    "Definitions of Direction Cosines"
END DEFINITIONS OF DIRCOS

where "Definitions of Direction Cosines" consists of the specified number of the following ordered set of records in arbitrary order:

BEGIN DIRCOS DEFINITION = (8 character literal)
    "Direction Cosine Definition"
END DIRCOS DEFINITION

The direction cosine identifier corresponds to one or more orientations specified elsewhere in "Analysis Data". The "Direction Cosine Definition" consists of one and only one of any of the following set of records:

"Direction Cosine Array"
        DCA = (real array of length 9)

        ;
"Euler-Rodrigues Parameters"
        ERP = (real array of length 3)
"Euler Rotation"
        EUR = (real array of length 3), (real)
"Tait-Bryan Angles"
        TBA = (real array of length 3), (integer)

The values of the 3x3 array DCA are the 9 direction cosines, and are input by rows in one or more records with a semicolon after the final entry. The EUR entries consist of a vector that defines the direction of the rotation, and the rotation angle (in degrees). The TBA entries include the three Tait-Bryan angles (in degrees), and an integer of length three which defines the sequence of the rotations (123, 132, 231, 213, 312, or 321).

b.) "Vectors" consists of no data or the ordered set of records:

        NUMBER OF VECTOR DEFINITIONS = (integer, default = 0)
        BEGIN DEFINITIONS OF VECTORS
                "Definitions of Vectors"
        END DEFINITIONS OF VECTORS

where "Definitions of Vectors" consists of the specified number of the following ordered set of records in arbitrary order:

        BEGIN VECTOR DEFINITION = (8 character literal)
                VECTOR = (real array of length 3)
        END VECTOR DEFINITION

The vector identifier corresponds to one or more position vectors specified elsewhere in "Analysis Data". The three real numbers in the definition are the vector components along the 1-, 2-, and 3-axes of the local reference basis, respectively.

2.1.3.3.4. Field Parameters .

"Field Parameters" consists of the air density definitions, the gravity definitions and the inertial motion definitions. "Field Parameters" consists of the following set of records:

        BEGIN FIELD PARAMETERS
                "Air Density Definition"
                "Gravity Definition"
                "Inertial Motion Definition"
        END FIELD PARAMETERS

where the definitions may appear in arbitrary order within the record set.

a.) "Air Density Definitions" has the form:

NUMBER OF AIR DENSITY DEFINITIONS = (integer, default = 1)
BEGIN DEFINITIONS OF AIR DENSITY
    "Definition of Air Density"
END DEFINITIONS OF AIR DENSITY

where "Definition of Air Density" consists of the specified number of the following record set in arbitrary order:

BEGIN AIR DENSITY DEFINITION = (8 character literal, default = ' ')
    RHO = (real, default = 0.0)
END AIR DENSITY DEFINITION

The air density identifier corresponds to the AIR DENSITY IDENTIFIER entry under "Model Definition" (section 2.1.3.2). The definition consists only of the air density RHO. If the air density is defaulted but the ASPID entry under "Aeroelastic Beam Definitions" (section 2.1.3.3.2a) is not, GRASP will calculate the aerodynamic forces on every element in the model assuming that they are in vacuo.

b.) "Gravity Definitions" has the form:

NUMBER OF GRAVITY DEFINITIONS = (integer, default = 1)
BEGIN DEFINITIONS OF GRAVITY
    "Definition of Gravity"
END DEFINITIONS OF GRAVITY

where "Definition of Gravity" consists of the specified number of the following set of records in arbitrary order:

BEGIN GRAVITY DEFINITION = (8 character literal, default = ' ')
    GRAV = (real array of length 3, default = 0.0, 0.0, 0.0)
END GRAVITY DEFINITION

The gravity field identifier corresponds to the GRAVITY FIELD IDENTIFIER entry under "Model Definition" (section 2.1.3.2). The definition consists only of the gravity vector (GRAV) referenced to the root subsystem frame basis. If the gravity vector is defaulted, GRASP will calculate all of the gravitational forces for all of the elements in the model to be zero.

c.) "Inertial Motion Definition" has the form:

NUMBER OF IM DEFINITIONS = (integer, default = 1)
BEGIN DEFINITIONS OF IM
    "Definition of Inertial Motion"
END DEFINITIONS OF IM

where "Definition of Inertial Motion" consists of the specified number of the following set of records in arbitrary order:

BEGIN INERTIAL MOTION DEFINITION = (8 character literal, default = ' ')
    V = (real array of length 3, default = 0.0, 0.0 ,0.0)
    A = (real array of length 3, default = 0.0, 0.0, 0.0)

OMEGA = (real array of length 3, default = 0.0, 0.0, 0.0)
END INERTIAL MOTION DEFINITION

The inertial motion identifier corresponds to the INERTIAL MOTION IDENTIFIER entry under "Model Definition" (section 2.1.3.2). The definition consists of the velocity of the root subsystem reference frame relative to an inertial reference frame (in the root subsystem frame basis) V, the acceleration of the root subsystem reference frame relative to an inertial reference frame (in the root subsystem basis) A, and the angular velocity of the root subsystem reference frame relative to an inertial reference frame (in the root subsystem basis) OMEGA. These records may appear in arbitrary order within the record set.

If the inertial velocity, acceleration, or angular velocity are defaulted, GRASP will calculate the inertial forces on each element of the model without any contribution from the defaulted global inertial motion.

2.2. User Input File

The records which are specified in User Input override those parameters defined under "Configuration Data" (section 2.1.1) in User Data. Typically, User Input is placed at the end of the user's runstream (section 3), while User Data is a separate input file. The user can thus use User Input to change the format of the output (e.g., page width and length), the title of the run, or the level of error information to be printed in one run, without having to change the definitions in User Data.

The first record in User Input must be the keyword:

BEGIN USER INPUT

and the last record must be the keyword:

END USER INPUT

Between the first and last keywords, User Input may consist of no data or any of the following set of records in arbitrary order:

DIAGNOSTICS LEVEL = (integer)
PAGE LENGTH = (integer)
PAGE WIDTH = (integer)
RUN TITLE = (72 character literal)

The diagnostics level defines the level of error information to be printed. A diagnostics level of 1 prints warning and error messages. A level of 2 suppresses warning messages. A level of 0 dumps status information and the information manager tables, and prints error messages upon abnormal termination. The page length defines the number of lines to be printed on each page of output, while the page width defines the number of characters to be printed on each line. The run title is the title that will be placed on the opening and closing banner pages as well as at the top of each page of output. If any single record is omitted, GRASP will use the default value specified

31

in User Data (section 2.1.1). If the User Input file is omitted completely, all of the values will be defaulted.

3. RUNNING GRASP

Solving problems using GRASP is accomplished by running two programs. First, the Bu..d GRASP Input (BLDINP) program is run to convert a user-defined User Data file into the computer-oriented Input Data file. Then the GRASP program is run to produce the solutions to the problems specified by the user. The Input Data file must be the same in both progran.s and must not exist prior to its creation in BLDINP or be modified between its creation in BLDINP and its use in GRASP.

The procedures for invoking BLDINP and GRASP are installation dependent. The information supplied in this subsection applies specifically to the environment at the Aeroflightdymanics Directorate, which is collocated with the NASA Ames Research Center. In this environment, BLDINP, GRASP, and the associated libraries are resident on a Cray X-MP 48 computer operating under COS 1.14, which communicates with the Aeroflightdynamics Directorate VAX 11-785. A sample runstream for sequentially executing BLDINP and GRASP is included in Appendix C.

3.1. Invoking Build GRASP Input

The first step in invoking BLDINP is to gain access to the subroutines that are needed to run the program. The subroutines are resident in Cray libraries named BINPLIB (BLDINP main program and subroutines), ERRMLIB (Error Manager), INFMLIB (Information Manager), and UTILLIB (Utilities). The COS commands required to access these libraries are:

```
ACCESS,DN=BINPLIB,PDN=BINPLIB,ID=GRASPLIB.
ACCESS,DN=ERRMLIB,PDN=ERRMLIB,ID=GRASPLIB.
ACCESS,DN=INFMLIB,PDN=INFMLIB,ID=GRASPLIB.
ACCESS,DN=UTILLIB,PDN=UTILLIB,ID=GRASPLIB.
```

Next, the User Data file, which is stored on a remote VAX, must be obtained and, if desired, copied onto the output file. If User Data is copied, it must also be rewound. The COS commands required to get the file, then copy and rewind it are:

```
FETCH,DN=USRDAT,TEXT='$DISK filename.DAT'.
COPYS,I=USRDAT,O=$OUT.
REWIND,DN=USRDAT.
```

where filename.DAT is the complete VAX file specification with a .DAT extension.

After getting User Data, the FORTRAN units must be assigned. The file numbers and names used in BLDINP are:

BLDINP File Name	Unit Number	Cray File Name
User Data	1	USRDAT
Input Data	2	INPDAT
User Input	5	INPUT
User Output	6	RESULT

Since User Data (section 2.1) is provided as an external file, it must be assigned a unit number and name, as follows:

ASSIGN,DN=USRDAT,A=FT01.

Input Data is a mass storage file created by BLDINP for use by GRASP. Because it is a mass storage file, it must not be assigned in the runstream. User Input (section 2.2) is data that are included at the end of the COS commands. Since the Cray defaults the unit number and file name to those desired, it is not necessary to assign them in the runstream. User Output is the file to which the program writes using FORTRAN write statements. It includes any results and all warning and error information and is assigned as follows:

ASSIGN,DN=RESULT,A=FT06.

Having taken care of all of the preliminaries, BLDINP is ready to be run. The COS loader command identifies the data set containing program to be run and the data sets containing the libraries from which unsatisfied externals may be loaded.

LDR,DN=BINPLIB,LIB=INFMLIB:ERRMLIB:UTILLIB,^
    MAP=FULL,L=MAPFILE.

In this loader command, a full load map is created. The map may be useful if the program aborts, and need not be kept if the program terminates normally.

After the program has run, the program output must be returned to the user. For normal termination of the program, only the results and the logfile are needed. The following COS commands release User Data and the load map (they are no longer needed), rewind the results file and copy it into the logfile, and dispose that file back to the remote VAX.

RELEASE,DN=USRDAT.
RELEASE,DN=MAPFILE.
REWIND,DN=RESULT.
COPYF,I=RESULT,O=$OUT.
DISPOSE,DN=$OUT,DEFER,TEXT='$DISK filename.OUT'.

Should the program abort, it is often convenient to print out the results, log file, and map separately. The following COS commands perform this function:

EXIT.
REWIND,DN=RESULT.
DISPOSE,WAIT,DN=RESULT,TEXT='$DISK filename.RSL'.
REWIND,DN=MAPFILE.
DISPOSE,DEFER,DN=$OUT,TEXT='$DISK filename.LOG'.
DISPOSE,WAIT,DN=MAPFILE,TEXT='$DISK filename.MAP'.

Note that the EXIT command serves two purposes. If the program terminates normally, the operating system reads the command and terminates the job without performing the commands that come after it. If the program aborts, the operating system skips everything between the abort and the EXIT, then resumes executing COS commands.

If BLDINP is run in a separate job from GRASP, it also becomes necessary to save the Input Data file for later use. In such a case, the user should consult the Cray COS Reference Manual.

34

3.2. Invoking GRASP

The steps required to invoke GRASP are very similar to those required to invoke BLDINP. In fact, many of the commands required duplicate those used to run BLDINP. In the event that BLDINP and GRASP are run sequentially in a single job, the duplicate commands should not be repeated (see Appendix C).

The first step is again to gain access to the subroutines that are needed to run the program. The subroutines are resident in Cray libraries named EXECLIB (GRASP main program and subroutines), ERRMLIB (Error Manager), INFMLIB (Information Manager), MATHLIB (Managed Information Utilities), UTILLIB (Utilities), and the IMSL library. The COS commands required to access these libraries are:

```
ACCESS,DN=EXECLIB,PDN=EXECLIB,ID=GRASPLIB.
ACCESS,DN=ERRMLIB,PDN=ERRMLIB,ID=GRASPLIB.
ACCESS,DN=INFMLIB,PDN=INFMLIB,ID=GRASPLIB.
ACCESS,DN=MATHLIB,PDN=MATHLIB,ID=GRASPLIB.
ACCESS,DN=UTILLIB,PDN=UTILLIB,ID=GRASPLIB.
AMESLIB.
IMSLIB.
```

The data used by GRASP are stored in the Input Data file, which was created by BLDINP. If BLDINP and GRASP are run sequentially, Input Data is automatically available to GRASP. If, however, GRASP is using a previously created Input Data file, that file must also be obtained.

The FORTRAN units must be then assigned. The file numbers and names used in GRASP are:

GRASP File Name	Unit Number	Cray File Name
Input Data	2	INPDAT
User Input	5	INPUT
User Output	6	RESULT

Input Data is the mass storage file created by BLDINP and is not assigned in the runstream. User Input (section 2.2) is data that are included at the end of the COS commands. Since the Cray defaults the unit number and file name to those desired, it is not necessary to assign them in the runstream. User Output is the file which contains all of the results of the GRASP calculations, plus any warning and error information. It is assigned as follows:

```
ASSIGN,DN=RESULT,A=FT06.
```

Having taken care of all of the preliminaries, GRASP is ready to be loaded and executed. The COS loader command identifies the data set containing the program to be run and the data sets containing the libraries from which unsatisfied externals may be loaded.

```
LDR,DN=EXECLIB,LIB=MATHLIB:INFMLIB:ERRMLIB:UTILLIB:IMSL,^
    MAP=FULL,L=MAPFILE.
```

In this loader command, a full load map is created. The map may be useful if the program aborts, and need not be kept if the program terminates normally.

35

After the program has run, the only thing left to do is to return the program output to the user. For normal termination of the program, only the results and the logfile are needed. The following COS commands release User Data and the load map (they are no longer needed), rewind the results file and copy it into the logfile, and dispose that file back to the remote VAX.

```
RELEASE,DN=MAPFILE.
REWIND,DN=RESULT.
COPYF,I=RESULT,O=$OUT.
DISPOSE,DN=$OUT,DEFER,TEXT='$DISK filename.OUT'.
```

Should the program abort, it is often convenient to print out the results, log file, and map separately. The following COS commands perform this function:

```
EXIT.
REWIND,DN=RESULT.
DISPOSE,WAIT,DN=RESULT,TEXT='$DISK filename.RSL'.
REWIND,DN=MAPFILE.
DISPOSE,DEFER,DN=$OUT,TEXT='$DISK filename.LOG'.
DISPOSE,WAIT,DN=MAPFILE,TEXT='$DISK filename.MAP'.
```

Note that the EXIT command serves two purposes. If the program terminates normally, the operating system reads the command and terminates the job without performing the commands that come after it. If the program aborts, the operating system skips everything between the abort and the EXIT, then resumes executing COS commands.

4. INTERPRETING OUTPUT

4.1. User Output

User Output consists of all of the results produced by GRASP and any error or w‹ rning messages issued by the program. Upon normal termination, User Output is combined with the Log File (section 4.2), and appears just before the system messages. If GRASP terminates abnormally, User Output is sent to a separate .RSL file (section 3.2).

The contents of User Output depend largely on the directives issued by the user under "Control Data" (section 2.1.2). Through these directives, any data structure named at the directive level may be sent to User Output. In the following sections, the data structures used and produced by the Steady-state Solution and the Asymmetric Eigensolution will be described.

4.1.1. Steady-state Problem

Solution of the steady-state problem produces several different data structures that the user may wish to examine. These structures include the system model, the steady-deformations and residuals, and the steady-state solution. Each of these can be named by the user under the "SOLVE STEADY-STATE" directive (section 2.1.2). The following sections describe the contents and format of each data structure as it is displayed in User Output.

4.1.1.1. System Model

The system model for the steady-state problem consists of all of the data that GRASP uses internally to describe the model defined by the user in User Data. The only circumstance under which the system model is printed is if the user defines the model name to be something other than the default model name under the "SOLVE STEADY-STATE" directive (section 2.1.2). Documentation for the system model may be found in the GRASP Programmer's Manual.

The data contained in the system model are organized heirarchically as follows:

```
System Model (mdlnam)
    Fields Model (FLmdlnam)
        Air Flow Model (AFFLmdlnam)
            Aeroelastic Beam Air Flow Model (AEBAFFLmdlnam)
            Air Mass Air Flow Model (AEBAFFLmdlnam)
        Gravity Field Model (GRVFLDmdlnam)
        Inertial Motion Model (INRMTNmdlnam)
    Elements Model (ELmdlnam)
        Aeroelastic Beams Model (ABELmdlnam)
            AEB Parameters Model (PRABELmdlnam)
                AEB Information (INPRABELmdlnam)
                AEB Orders (ORPRABELmdlnam)
                AEB Gaussian Quadrature Points (QPPRABELmdlnam)
                AEB Gaussian Quadrature Weights (QPPRABELmdlnam)
                AEB ZXSSQ Parameters (ZXPRABELmdlnam)
```

AEB Geometric Properties Model (GPABELmdlnam)
    AEB Shape Function Identifiers (SIGPABELmdlnam)
    AEB Shape Functions (SFGPABELmdlnam)
    AEB Geometric Function Identifiers (GIGPABELmdlnam)
    AEB Geometric Functions (GFGPABELmdlnam)
AEB Material Properties Model (MPABELmdlnam)
    AEB Elastic Section Properties Identifiers (EIMPABELmdlnam)
    AEB Elastic Section Properties (ESMPABELmdlnam)
    AEB Inertial Section Properties Identifiers (IIMPABELmdlnam)
    AEB Inertial Section Properties (ISMPABELmdlnam)
AEB Aerodynamic Properties Model (APABELmdlnam)
    AEB Aerodynamic Properties Identifiers (AIAPABELmdlnam)
    AEB Aerodynamic Coefficient Tables (ACAPABELmdlnam)
    AEB Aerodynamic Section Properties (ASAPABELmdlnam)
Air Masses Model (AMELmdlnam)
Direct Input Model (DIELmdlnam)
Rigid-body Masses Model (RMELmdlnam)
Constraints Model (CNmdlnam)
Constraints Table (CNSTABCNmdlnam)
Constraints Parameters (CNSPARCNmdlnam)
    Constraint Parameters for CAM (CPRCAMCNmdlnam)
    Constraint Parameters for CPY (CPRCPYCNmdlnam)
    Constraint Parameters for FFR (CPRFFRCNmdlnam)
    Constraint Parameters for PAM (CPRPAMCNmdlnam)
    Constraint Parameters for PFR (CPRPFRCNmdlnam)
    Constraint Parameters for PGC (CPRPGCCNmdlnam)
    Constraint Parameters for PND (CPRPNDCNmdlnam)
    Constraint Parameters for PRE (CPRPRECNmdlnam)
    Constraint Parameters for RAM (CPRRAMCNmdlnam)
    Constraint Parameters for RFR (CPRRFRCNmdlnam)
    Constraint Parameters for RGC (CPRRGCCNmdlnam)
    Constraint Parameters for RND (CPRRNDCNmdlnam)
    Constraint Parameters for SCR (CPRSCRCNmdlnam)
    Constraint Parameters for SND (CPRSNDCNmdlnam)
    Constraint Parameters for SSR (CPRSSRCNmdlnam)
Constraint Vectors (CNSVECCNmdlnam)
Constraint Direction Cosine Arrays (CNSDCACNmdlnam)
Variables Model (VBmdlnam)
Lengths of State and Residual Vectors (LNSTRSVBmdlnam)
Lengths of AEB Internal State and Residual Vectors (LABISRVBmdlnam)
Variable Identifiers (VRBLIDVBmdlnam)
    Subsystem Variable Identifiers (SBSVBLVBmdlnam)
    AEB Internals Variable Identifiers (ABIVBLmdlnam)

As displayed in User Output, the heirachical structure is not evident, but the order of the constituent data structures conforms to the order of data structures in the list above.

4.1.1.2. Steady-state Deformations

The Steady-state Deformations data structure contains the steady-state values for all of ιe degrees of freedom in the model. To print out this data structure, its name (sspdef) must be defined under the "SOLVE STEADY-STATE" directive (section 2.1.2) to be something other than the default name, and a "PRINT DATA SET" directive (section 2.1.2) must be give ι.

Within the Steady-state Deformations data structure are two data structures: the Steady-state Deformations for Subsystems (SDSSsspdef), and the Steady-state Deformations for Aeroelastic Beam Internals (SDAIsspdef). Each of these two structures consists of a set of single-column matrices. The Steady-state Deformations for Subsystems data structure contains one matrix for each subsystem, while the Steady-state Deformations for Aeroelastic Beam Internals data structure contains one matrix for each aeroelastic beam element. If a beam element has no internal degrees of freedom, the corresponding Aeroelastic Beam Internals Steady-state Deformations matrix has a length of zero.

The degree of freedom values for each subsystem are contained in the individual single-column matrices that are the constituents of the Steady-state Deformations for Subsystems data structure. These matrices are printed in order of ascending subsystem pre-order number. The names of the Subsystem Steady-state Deformations matrices can be used to identify the subsystem names, since they are all of the form "SSSDsbsnam", where sbsnam is the subsystem name. The values of the degrees of freedom in each matrix are identified only by the degree of freedom number. The degree of freedom names can be found in the System Model under the Subsystem Variables Identifiers (section 4.1.1.1).

The internal degree of freedom values for each aeroelastic beam are contained in the individual single-column matrices that are the constituents of the Steady-state Deformations for Aeroelastic Beam Internals data structure. These matrices are printed in order of ascending aeroelastic beam number. The names of the Aeroelastic Beam Internals Steady-state Deformations matrices can be used to identify the aeroelastic beam elements, since they are all of the form "AISDsbsnam", where sbsnam is the subsystem name. The values of the degrees of freedom in each matrix are identified only by the degree of freedom number. The degree of freedom names can be found in the System Model under the Aeroelastic Beam Internals Variables Identifiers (section 4.1.1.1).

4.1.1.3. Steady-state Residuals

The Steady-state Residuals data structure is the data structure that contains the values of the steady-state residuals for the degrees of freedom for each element in the model. To print out this data structure, its name (ssprsd) must be defined under the "SOLVE STEADY-STATE" directive (section 2.1.2) to be something other than the default name, and a "PRINT DATA SET" directive (section 2.1.2) must be given.

Within the Steady-state Residuals data structure are four data structures: the Steady-state Aeroelastic Beam Residuals (ABELMRSD), the Steady-state Air Mass Residuals (AMELMRSD), the Steady-state Direct Input Residuals (DIELMRSD), and the Steady-state Rigid-body Mass Residuals (RMELMRSD). Each of these structures consists of a set of single-column matrices that contain the residuals for all of the elements of that type in the model. If, for example, a model included three rigid-body mass elements, there would be three matrices contained in the Steady-state Rigid-body Mass Residuals data structure.

The degree of freedom residual values for each element are contained in the individual single-column matrices that are the constituents of the Steady-state Aeroelastic Beam Residuals, the Steady-state Air Mass Residuals, the Steady-state Direct Input Residuals, and the Steady-state Rigid-body Mass Residuals data structures. For each element type, these matrices are printed in order of ascending subsystem number. The names of the matrices can be used to identify the subsystem names, since they are all of the form "sbsnamABELMRSD", "sbsnamAMELMRSD", "sbsnamDIELMRSD", or "sbsnamRMELMRSD", where sbsnam is the subsystem name. The residual values for the degrees of freedom in each matrix are identified only by the degree of freedom number. The degree of freedom names can be found in the System Model under the Subsystem Variables Identifiers (section 4.1.1.1).

4.1.1.4. Steady-state Solution

The Steady-state Solution data structure displays the primary results of a steady-state problem in a format that is convenient to the user. The data structure is printed by default if its name is not specified under the "SOLVE STEADY-STATE" directive (section 2.1.2). If the name is specified, however, a "PRINT DATA SET" directive (section 2.1.2) must be given in order to have the data structure printed.

For each subsystem in the structure, the values of the steady-state deformations are displayed along with the degree of freedom numbers and the variable identifiers. Preceding the deformations for each subsystem is a header which identifies the subsystem pre-order number, name, and type. The results are displayed in three columns: degree of freedom numbers, steady-state deformation values, and degree of freedom variable identifiers. The variable identifiers consist of four words which uniquely identify each degree of freedom: subsystem name, node name, and degree of freedom name (two words).

4.1.2. Asymmetric Eigenproblem

Solution of the asymmetric eigenproblem produces several different data structures that the user may wish to examine. These structures include the system model, the reference deformations and residuals, the asymmetic eigenproblem matrices, the asymmetric eigenproblem intermediate solution, and the asymmetric eigenproblem solution. Each of these data structures can be named by the user under the "SOLVE ASYMMETRIC EIGEN-PROBLEM" directive (section 2.1.2). The following sections describe the contents and format of each data structure as it is displayed in User Output.

4.1.2.1. System Model

The system model for the asymmetric eigenproblem consists of all of the data that GRASP uses internally to describe the model defined by the user in User Data. The only circumstance under which the system model is printed is if the user defines the model name to be something other than the default model name under the "SOLVE ASYMMETRIC EIGEN-PROBLEM" directive (section 2.1.2). Documentation for the system model may be found in the GRASP Programmer's Manual.

The data contained in the system model are organized heirarchically as follows:

System Model (mdlnam)
    Fields Model (FLmdlnam)
        Air Flow Model (AFFLmdlnam)
            Aeroelastic Beam Air Flow Model (AEBAFFLmdlnam)
            Air Mass Air Flow Model (AEBAFFLmdlnam)
        Gravity Field Model (GRVFLDmdlnam)
        Inertial Motion Model (INRMTNmdlnam)
    Elements Model (ELmdlnam)
        Aeroelastic Beams Model (ABELmdlnam)
            AEB Parameters Model (PRABELmdlnam)
                AEB Information (INPRABELmdlnam)
                AEB Orders (ORPRABELmdlnam)
                AEB Gaussian Quadrature Points (QPPRABELmdlnam)
                AEB Gaussian Quadrature Weights (QPPRABELmdlnam)
                AEB ZXSSQ Parameters (ZXPRABELmdlnam)
            AEB Geometric Properties Model (GPABELmdlnam)
                AEB Shape Function Identifiers (SIGPABELmdlnam)
                AEB Shape Functions (SFGPABELmdlnam)
                AEB Geometric Function Identifiers (GIGPABELmdlnam)
                AEB Geometric Functions (GFGPABELmdlnam)
            AEB Material Properties Model (MPABELmdlnam)
                AEB Elastic Section Properties Identifiers (EIMPABELmdlnam)
                AEB Elastic Section Properties (ESMPABELmdlnam)
                AEB Inertial Section Properties Identifiers (IIMPABELmdlnam)
                AEB Inertial Section Properties (ISMPABELmdlnam)
            AEB Aerodynamic Properties Model (APABELmdlnam)
                AEB Aerodynamic Properties Identifiers (AIAPABELmdlnam)
                AEB Aerodynamic Coefficient Tables (ACAPABELmdlnam)
                AEB Aerodynamic Section Properties (ASAPABELmdlnam)
        Air Masses Model (AMELmdlnam)
        Direct Input Model (DIELmdlnam)
        Rigid-body Masses Model (RMELmdlnam)
    Constraints Model (CNmdlnam)
        Constraints Table (CNSTABCNmdlnam)
        Constraints Parameters (CNSPARCNmdlnam)
            Constraint Parameters for CAM (CPRCAMCNmdlnam)
            Constraint Parameters for CPY (CPRCPYCNmdlnam)
            Constraint Parameters for FFR (CPRFFRCNmdlnam)
            Constraint Parameters for PAM (CPRPAMCNmdlnam)
            Constraint Parameters for PFR (CPRPFRCNmdlnam)
            Constraint Parameters for PGC (CPRPGCCNmdlnam)
            Constraint Parameters for PND (CPRPNDCNmdlnam)
            Constraint Parameters for PRE (CPRPRECNmdlnam)
            Constraint Parameters for RAM (CPRRAMCNmdlnam)
            Constraint Parameters for RFR (CPRRFRCNmdlnam)

Constraint Parameters for RGC (CPRRGCCNmdlnam)
Constraint Parameters for RND (CPRRNDCNmdlnam)
Constraint Parameters for SCR (CPRSCRCNmdlnam)
Constraint Parameters for SND (CPRSNDCNmdlnam)
Constraint Parameters for SSR (CPRSSRCNmdlnam)
Constraint Vectors (CNSVECCNmdlnam)
Constraint Direction Cosine Arrays (CNSDCACNmdlnam)
Variables Model (VBmdlnam)
Lengths of State and Residual Vectors (LNSTRSVBmdlnam)
Lengths of AEB Internal State and Residual Vectors (LABISRVBmdlnam)
Variable Identifiers (VRBLIDVBmdlnam)
Subsystem Variable Identifiers (SBSVBLVBmdlnam)
AEB Internals Variable Identifiers (ABIVBLmdlnam)

As displayed in User Output, the heirarchical structure is not evident, but the order of the constituent data structures conforms to the order of data structures in the list above.

For a single User Data file, the System Models for the steady-state problem (section 4.1.1.1) and the asymmetric eigenproblem are very similar, but they are not identical. For example, the Lengths of Aeroelastic Beam Internal State and Residual Vectors are always zero for the asymmetric eigenproblem because all of the internal degrees of freedom are combined with the subsystem degrees of freedom. Thus, for any aeroelastic beam element, the sum of the subsystem and internal degrees is the same for both the steady-state problem and the asymmetric eigenproblem. The entries in the Subsystem and Aeroelastic Beam Internals Variable Identifiers also reflect this difference in the models.

4.1.2.2.   Reference Deformations

The Reference Deformations data structure contains the steady-state values for all of the degrees of freedom in the model. Since the Reference Deformations are calculated by a steady-state problem, it will be identical to one of the Steady-state Deformations data structures (section 4.1.1.2) created in the same GRASP run. To print out this data structure, its name (refdef) must be defined under the "SOLVE ASYMMETRIC EIGEN-PROBLEM" directive (section 2.1.2) to be something other than the default name, and a "PRINT DATA SET" directive (section 2.1.2) must be given.

Within the Reference Deformations data structure are two data structures: the Reference Deformations for Subsystems (SDSSrefdef), and the Reference Deformations for Aeroelastic Beam Internals (SDAIrefdef). Each of these two structures consists of a set of single-column matrices. The Reference Deformations for Subsystems data structure contains one matrix for each subsystem, while the Reference Deformations for Aeroelastic Beam Internals data structure contains one matrix for each aeroelastic beam element. If a beam element has no internal degrees of freedom, the corresponding Aeroelastic Beam Internals Reference Deformations matrix has a length of zero.

The degree of freedom values for each subsystem are contained in the individual single-column matrices that are the constituents of the Reference Deformations for Subsystems data structure. These matrices are printed in order of ascending subsystem pre-order number. The names of the

Subsystem Reference Deformations matrices can be used to identify the subsystem names, since they are all of the form "SSSDsbsnam", where sbsnam is the subsystem name. The values of the degrees of freedom in each matrix are identified only by the degree of freedom number. The degree of freedom names can be found in the System Model for the corresponding steady-state problem under the Subsystem Variables Identifiers (section 4.1.1.1).

The internal degree of freedom values for each aeroelastic beam are contained in the individual single-column matrices that are the constituents of the Reference Deformations for Aeroelastic Beam Internals data structure. These matrices are printed in order of ascending aeroelastic beam number. The names of the Aeroelastic Beam Internals Reference Deformations matrices can be used to identify the aeroelastic beam elements, since they are all of the form "AISDsbsnam", where sbsnam is the subsystem name. The values of the degrees of freedom in each matrix are identified only by the degree of freedom number. The degree of freedom names can be found in the System Model for the corresponding steady-state problem under the Aeroelastic Beam Internals Variables Identifiers (section 4.1.1.1).

4.1.2.3. Reference Residuals

The Reference Residuals data structure is the data structure that contains the values of the steady-state residuals for the degrees of freedom for each element in the model. Since the Reference Residuals are calculated by the steady-state problem, it will be identical to one of the Steady-state Residuals data structures (section 4.1.1.3) created in the same GRASP run. To print out this data structure, its name (refrsd) must be defined under the "SOLVE ASYMMETRIC EIGEN-PROBLEM" directive (section 2.1.2) to be something other than the default name, and a "PRINT DATA SET" directive (section 2.1.2) must be given.

Within Reference Residuals data structure are four data structures: the Reference Aeroelastic Beam Residuals (ABELMRSD), the Reference Air Mass Residuals (AMELMRSD), Reference Direct Input Residuals (DIELMRSD), and the Reference Rigid-body Mass Residuals (RMELMRSD). Each of these structures consists of a set of single-column matrices that contain the residuals for all of the elements of that type in the model. If, for example, a model included three rigid-body mass elements, there would be three matrices contained in the Reference Rigid-body Mass Residuals data structure.

The degree of freedom residual values for each element are contained in the individual single-column matrices that are the constituents of the Reference Aeroelastic Beam Residuals, the Reference Air Mass Residuals, the Reference Direct Input Residuals, and Reference Rigid-body Mass Residuals data structures. For each element type, these matrices are printed in order of ascending subsystem number. The names of the matrices can be used to identify the subsystem names, since they are all of the form "sbsnamABELMRSD", "sbsnamAMELMRSD", "sbsnamDIELMRSD", or "sbsnamRMELMRSD", where sbsnam is the subsystem name. The residual values for the degrees of freedom in each matrix are identified only by the degree of freedom number. The degree of freedom names can be found in the System Model under the Subsystem Variables Identifiers (section 4.1.1.1).

43

4.1.2.4. Asymmetric Eigenproblem Matrices

The Asymmetric Eigenproblem Matrices data structure contains the mass, damping, and stiffness matrices for the highest-level subsystem in the system model. To print these matrices, the name of the data structure must be defined in the "SOLVE ASYMMETRIC EIGEN-PROBLEM" directive (section 2.1.2) to be different from the default name. Then, a "PRINT DATA SET" directive (section 2.1.2) must also be given.

Each matrix is displayed in standard matrix format and can be identified by its name. The names for the mass, damping, and stiffness matrices are "Maepmtx", "Caepmtx", and "Kaepmtx" respectively, where aepmtx is the name of the Asymmetric Eigenproblem Matrices data structure. The degree of freedom names that correspond to the row and column numbers in the matrices can be found in the System Model under the Subsystem Variables Identifiers (section 4.1.1.1) for the highest-level subsystem.

4.1.2.5. Asymmetric Eigenproblem Intermediate Solution

The Asymmetric Eigenproblem Intermediate Solution data structure contains the eigenvalues and eigenvectors obtained from the system of equations defined by the Asymmetric Eigenproblem Matrices (section 4.1.2.4). To print this data structure its name must be defined under the "SOLVE ASYMMETRIC EIGEN-PROBLEM" directive (section 2.1.2) to be different from the default name. Then a "PRINT DATA SET" directive (section 2.1.2) must also be given.

The data structure itself consists of two matrices: the Asymmetric Linear System Eigenvalues and the Asymmetric Linear System Eigenvectors. The names of the eigenvalue and eigenvector data structures are "EVALaepisl" and "EVECaepisl" respectively, where aepisl is the name of the Asymmetric Eigenproblem Intermediate Solution.

The Asymmetric Linear System Eigenvalues data structure is a single-column matrix. Each row entry in the matrix is one of the eigenvalues of the linear system. The eigenvalues are not displayed in any particular order.

The Asymmetric Linear System Eigenvectors data structure is a matrix that has as many rows as there are degrees of freedom, an as many columns as there are eigenvalues. Each column of the matrix represents an eigenvector that corresponds to the eigenvalue in the eigenvalues matrix having a row number that is the same as the eigenvector column number. The eigenvector row numbers correspond to the degree of freedom names in the System Model under the Subsystem Variable Identifiers (section 4.1.1.1) for the highest-level subsystem.

4.1.2.6. Asymmetric Eigenproblem Solution

The Asymmetric Eigenproblem Solution data structure displays the primary results of an asymmetric eigenproblem in a format that is convenient to the user. The data structure is printed by default if its name is not specified under the "SOLVE ASYMMETRIC EIGEN-PROBLEM" directive (section 2.1.2). If the name is specified, however, a "PRINT DATA SET" directive (section 2.1.2) must be given in order to have the data structure printed.

For each subsystem in the structure, the eigenvectors calculated for that subsystem are displayed along with their associated eigenvectors. Preceding the eigenvectors for each subsystem

is a header which identifies the subsystem pre-order number, name, and type. Preceding each eigenvector is a header that identifies the mode number, the modal frequency, and the modal damping. The eigenvectors are displayed in three columns: degree of freedom numbers, complex eigenvector values, and degree of freedom variable identifiers. The variable identifiers consist of four words which uniquely identify each degree of freedom: subsystem name, node name and degree of freedom name (two words).

The Asymmetric Eigenproblem Solution will display only the number of eigenvectors specified under "Complex Eigenproblem Parameters Definitions" (section 2.1.3.1.2) or the maximum number of eigenvectors that can be calculated for the model, whichever is smaller. If the specified number of eigenvalues (section 2.1.3.1.2) is greater than the specified number of eigenvectors, but smaller than the maximum number of eigenvalues, the eigenvalues and eigenvectors will be displayed for the lower modes and only the eigenvalues will be displayed for the higher modes.

4.2. Log File

The log file consists of three parts: an optional echo of the User Data file, Cray system messages, and the job history. Using the COS commands outlined in section 3 and Appendix C, the log file is merged with User Output when the the job terminates normally. If the job does not terminate normally, the log file is returned separately.

The User Data echo displays the User Data file exactly as it was obtained from its source. If User Data is not explicitly copied to $OUT in the COS command runstream, the echo will not be produced. Also, since there is no User Data required to run the program GRASP, a GRASP run that does not first invoke BLDINP does not produce an echo.

Immediately in front of the job history are the system messages. These messages announce operating hours, help messages, and other useful information for users of the Cray.

The job history records the COS commands issued by the user and the system responses to those commands. Also, there may be messages issued during a run either by the computer or by the user program. For example, GRASP sends a message to the log file whenever it begins executing a new directive. One message that often appears after "Model System" is:

UT009 - NAMED RECORD NOT FOUND, NAME = xxxxxx UNIT = FT02

Although this message appears to indicate an error, it is really a result of a system-type subsystem having neither structural nor air nodes. It therefore may be ignored.

4.3. Map File

The map file is a complete Cray load map for GRASP. It is only generated if GRASP terminates abnormally. For most users, it will be of little help in debugging problems that arise in running GRASP, and may be suppressed by removing the appropriate COS commands from the job runstream. For information on how to read and interpret the load map, Consult the Cray COS Reference Manual.

GLOSSARY

A	acceleration of the root subsystem reference frame relative to an inertial reference frame in the root subsystem reference frame basis
ABC	aeroelastic beam connectivity
AC	aerodynamic coefficient
AE	aeroelastic
AEB	aeroelastic beam
AEBEAM	aeroelastic beam
AEPISL	asymmetric eigenproblem intermediate solution name
AEPMTX	asymmetric eigenproblem matrices name
AEPPAR	asymmetric eigenproblem parameters name
AEPSLN	asymmetric eigenproblem solution name
AIRMAS	air mass
AMC	air mass connectivity
AMS	air mass
AMSID	air mass properties identifier
ANDNAM	air node name
AS	aerodynamic section
ASP	aerodynamic section properties
ASPID	aerodynamic section properties identifier
ASSNAM	air subsystem name
BLDINP	build GRASP input
B1	section integral B1
B2	section integral B2
B3	section integral B3
CALPRI	calculate performance index flag
CDTID	drag coefficient table identifier
CDX	drag coefficient lengthwise variation
CHD	chord width
CLTID	lift coefficient table identifier
CLX	lift coefficient lengthwise variation
CMTID	moment coefficient table identifier
CMX	moment coefficient lengthwise variation
CNSID	constraint identifier
CNSSID	subsystem constraint set identifier
CNSTYP	constraint type
DCA	direction cosines array
DELTA	delta convergence criterion
DIRCOS	direction cosines
DNDNAM	dependent node name
DOF	degree of freedom
DOFNUM	degree of freedom number
DOFS	degrees of freedom
DOFSET	degree of freedom set name
DOFTYP	degree of freedom type

DOFVAL	degree of freedom value
DRCSID	direction cosines identifier
DSNAME	data-set name
DSSNAM	dependent subsystem name
D0	section integral D0
D1	section integral D1
D2	section integral D2
D3	section integral D3
D4	section integral D4
EPS	epsilon convergence criterion
ERP	Euler-Rodrigues parameters
ES	elastic section
ESP	elastic section properties
ESPID	elastic section properties identifier
EUR	Euler rotation
E0	axial stiffness
E1	first area moment of axial stiffness about the 1-axis
E2	first area moment of axial stiffness about the 2-axis
FFR	fixed frame
FRDCID	frame orientation direction cosines identifier
FROFID	frame position identifier
G	geometric
GEOM	geometric
GEOMID	geometric properties identifier
GRAV	gravity vector
IM	inertial motion
INDNAM	independent node name
INPDAT	input data
IOPT	input options parameter
IS	inertial section
ISP	inertial section property
ISPID	inertial section properties identifier
ISSNAM	independent subsystem name
I1	bending stiffness about the 1-axis
I2	bending stiffness about the 2-axis
J	torsion stiffness
L	length
MAXFN	maximum number of function evaluations
MDLID	model identifier
MDLNAM	model name
MM1	first area moment of mass about 1-axis
MM2	first area moment of mass about 2-axis
MOI	moments of inertia
MOI1	moment of inertia about 1-axis
MOI2	moment of inertia about 2-axis

MSS	mass per unit length
N	number of subsystems
NDDCID	node direction cosine identifier
NDPSID	node position identifier
NGAUSS	number of Gauss quadrature points
NODDEF	node definition
NODNAM	node name
NSIG	number of significant digits convergence criterion
NVAL	number of eigenvalues
NVEC	number of eigenvectors
RQPZ2	offset of the aerodynamic center from the elastic axis
OFFSID	position identifier
OMEGA	angular velocity
ORD	orders
ORDID	orders identifier
ORDT3	order of the T3 expansion
ORDU1	order of the U1 expansion
ORDU2	order of the U2 expansion
ORDU3	order of the U3 expansion
OUTRSL	output results
PARM	input parameters
PFR	periodic frame
PRE	prescribed
PST	periodic structure
PSSNAM	parent subsystem name
RB	rigid-body
RBC	rigid-body connection
RBM	rigid-body mass
RBMASS	rigid-body mass
REFDEF	reference deformations name
REFRSD	reference residuals name
RFR	rotating frame
RHO	density
RMC	rigid-body mass connectivity
RNDNAM	root node name
RST	rotating structure
RSSNAM	root subsystem name
RTDCID	root orientation direction cosines identifier
RTOFID	root position identifier
SBSID	subsystem identifier
SBSNAM	subsystem name
SBSTYP	subsystem type
SCAUVI	screw axis unit vector identifier
SCOFID	screw axis position identifier
SCR	screw connection

SCRROT	screw rotation flag
SCRTRA	screw translation flag
SGNAA	sign of the angle of attack
SS	subsystem
SSNNAM	subsystem node name
SSPDEF	steady-state problem deformations name
SSPPAR	steady-state problem parameters name
SSPRSD	steady-state problem residuals name
SSPSLN	steady-state problem solution name
STR	structural degree of freedom
SYSTEM	system
TBA	Tait-Bryan angles
THETA	pretwist angle
TNDNAM	tip node name
TSSNAM	tip subsystem name
TZL	angle of the zero-lift line
T0	screw rotation offset
USRDAT	user data
U0	screw translation offset
V	velocity of the root subsystem reference frame relative to an inertial reference frame in the root subsystem reference frame basis
ZXPID	ZXSSQ parameters identifier
ZXSSQ	IMSL steady-state equation solver

A. USER DATA SUMMARY

The hierarchical structure of the User Data file is summarized in this appendix. Since blanks are ignored, indentation is arbitrary but useful in reading the file. Within any level of the hierarchy, the order of records is arbitrary, except that the definition of the number (or length) of a set of records must precede the definition of that record set. For example, NUMBER OF MODELS must precede BEGIN MODELS, and LENGTH OF SS NODE DEFINITION SET must precede BEGIN SUB-SYSTEM NODE DEFINITIONS.

Character constants consist of a string of characters enclosed in apostrophes, with the maximum length as indicated. Arrays consist of a series of real or integer numbers separated by commas. A character constant or an array may either be on the same line as the keyword for that record, or on the following line. Except for direction cosine arrays (section 2.1.3.3.3a), continuations from one line to the next are not allowed.

USER DATA

```
BEGIN DATA
    BEGIN CONFIGURATION DATA
        DIAGNOSTICS LEVEL = (integer)
        PAGE LENGTH = (integer)
        PAGE WIDTH = (integer)
        RUN TITLE = (character*72)
    END CONFIGURATION DATA
    BEGIN CONTROL DATA
        NUMBER OF CASE DEFINITIONS = (integer)
        BEGIN DEFINITIONS OF CASES
            BEGIN CASE DEFINITION
                CASE IDENTIFIER = (character*8)
                CASE TITLE = (character*72)
                NUMBER OF DIRECTIVES = (integer)
                BEGIN DIRECTIVES
                    SOLVE STEADY-STATE
                        MDLID = (character*8)
                        SSPPAR = (character*8)
                        MDLNAM = (character*8)
                        SSPDEF = (character*8)
                        SSPRSD = (character*8)
                        SSPSLN = (character*8)
                        ;
                    SOLVE ASYMMETRIC EIGEN-PROBLEM
                        MDLID = (character*8)
                        AEPPAR = (character*8)
                        MDLNAM = (character*8)
                        REFDEF = (character*8)
                        REFRSD = (character*8)
                        AEPMTX = (character*8)
                        AEPISL = (character*8)
                        AEPSLN = (character*8)
                        ;
                    PRINT DATA SET
                        DSNAME = (character*8)
                        ;
                    RELEASE DATA SET
                        DSNAME = (character*8)
                        ;
                END DIRECTIVES
            END CASE DEFINITION
        END DEFINITIONS OF CASES
    END CONTROL DATA
    BEGIN ANALYSIS DATA
        NUMBER OF CASES OF DATA = (integer)
        BEGIN DATA FOR CASES
            BEGIN CASE DATA = (character*8)
                BEGIN SOLUTION PARAMETERS
                    NUMBER OF ZXSSQ PARAMETERS = (integer)
                    BEGIN ZXSSQ PARAMETERS
                        BEGIN ZXSSQ DEFINITION = (character*8)
                            NSIG = (integer)
                            EPS = (REAL)
                            DELTA = (REAL)
                            MAXFN = (integer)
                            IOPT = (integer)
                            PARM = (integer array of length 4)
                        END ZXSSQ DEFINITION
                    END ZXSSQ PARAMETERS
```

```
NUMBER OF COMPLEX EIGEN PARAMETERS = (integer)
BEGIN COMPLEX EIGEN PARAMETERS
    BEGIN COMPLEX EIGEN DEFINITION = (character*8)
        NVAL = (integer)
        NVEC = (integer)
        CALPRI = ('TRUE' or 'FALSE')
    END COMPLEX EIGEN DEFINITION
END COMPLEX EIGEN PARAMETERS
END SOLUTION PARAMETERS
NUMBER OF MODELS = (integer)
BEGIN MODELS
    BEGIN MODEL DEFINITION = (character*8)
        AIR DENSITY IDENTIFIER = (character*8)
        GRAVITY FIELD IDENTIFIER = (character*8)
        INERTIAL MOTION IDENTIFIER = (character*8)
        SYSTEM ORGANIZATION IDENTIFIER = (character*8)
    END MODEL DEFINITION
END MODELS
BEGIN MODEL PARAMETERS
    BEGIN SYSTEM PARAMETERS
        NUMBER OF SYSTEMS = (integer)
        BEGIN SYSTEM ORGANIZATION DATA
            BEGIN SYSTEM DEFINITION = (character*8)
                NUMBER OF SUB-SYSTEMS = (integer)
                BEGIN DEFINITIONS OF SUB-SYSTEMS
                    BEGIN SUB-SYSTEM DEFINITION
                        PSSNAM = (character*8)
                        SBSNAM = (character*8)
                        SBSTYP = ('SYSTEM', 'AEBEAM',
                        'AIRMAS', or 'RBMASS')
                        SBSID = (character*8)
                        CNSSID = (character*8)
                    END SUB-SYSTEM DEFINITION
                END DEFINITIONS OF SUB-SYSTEMS
            END SYSTEM DEFINITION
        END SYSTEM ORGANIZATION DATA
        NUMBER OF SS NODDEF SETS = (integer)
        BEGIN NODE DEFINITIONS FOR SS
            BEGIN SS NODE DEFINITION SET = (character*8)
                LENGTH OF SS NODE DEFINITION SET = (integer)
                BEGIN SUB-SYSTEM NODE DEFINITIONS
                    BEGIN DEFINITION OF SS NODE
                        SSNNAM = (character*8)
                        NDPSID = (character*8)
                        NDDCID = (character*8)
                    END DEFINITION OF SS NODE
                END SUB-SYSTEM NODE DEFINITIONS
            END SS NODE DEFINITION SET
        END NODE DEFINITIONS FOR SS
        NUMBER OF SS AIR NODE SETS = (integer)
        BEGIN SS AIR NODE DEFINITION SETS
            BEGIN DEFINITION OF SS AIR NODE SET = (character*8)
                LENGTH OF SS AIR NODE SET = (integer)
                BEGIN DEFINITIONS OF AIR NODES
                    BEGIN AIR NODE DEFINITION
                        ANDNAM = (character*8)
                        OFFSID = (character*8)
                        DRCSID = (character*8)
                    END AIR NODE DEFINITION
                END DEFINITIONS OF AIR NODES
```

```
            END DEFINITION OF SS AIR NODE SET
    END SS AIR NODE DEFINITION SETS
    NUMBER OF SS CONSTRAINT SETS = (integer)
    BEGIN CONSTRAINTS FOR SUB-SYSTEMS
        BEGIN SUB-SYSTEM CONSTRAINT SET = (character*8)
            LENGTH OF SS CONSTRAINT SET = (integer)
            BEGIN SS CONSTRAINT DEFINITIONS
                BEGIN DEFINITION OF SS CONSTRAINT
                    CNSTYP = (character*8)
                    CNSID = (character*8)
                END DEFINITION OF SS CONSTRAINT
            END SS CONSTRAINT DEFINITIONS
        END SUB-SYSTEM CONSTRAINT SET
    END CONSTRAINTS FOR SUB-SYSTEMS
    LENGTH OF AEB CONNECTIVITY TABLE = (integer)
    BEGIN AEB CONNECTIVITY TABLE
        BEGIN ABC DEFINITION = (character*8)
            RSSNAM = (character*8)
            RNDNAM = (character*8)
            TSSNAM = (character*8)
            TNDNAM = (character*8)
            ASSNAM = (character*8)
            ANDNAM = (character*8)
            RTOFID = (character*8)
            RTDCID = (character*8)
        END ABC DEFINITION
    END AEB CONNECTIVITY TABLE
    LENGTH OF AMS CONNECTIVITY TABLE = (integer)
    BEGIN AMS CONNECTIVITY TABLE
        BEGIN AMC DEFINITION = (character*8)
            SBSNAM = (character*8)
            ANDNAM = (character*8)
        END AMC DEFINITION
    END AMS CONNECTIVITY TABLE
    LENGTH OF FFR CONSTRAINT TABLE = (integer)
    BEGIN FFR CONSTRAINT TABLE
        BEGIN FFR CONSTRAINT DEFINITION = (character*8)
            FROFID = (character*8)
            FRDCID = (character*8)
        END FFR CONSTRAINT DEFINITION
    END FFR CONSTRAINT TABLE
    LENGTH OF PFR CONSTRAINT TABLE = (integer)
    BEGIN PFR CONSTRAINT TABLE
        BEGIN PFR CONSTRAINT DEFINITION = (character*8)
            N = (integer)
            FROFID = (character*8)
            FRDCID = (character*8)
        END PFR CONSTRAINT DEFINITION
    END PFR CONSTRAINT TABLE
    LENGTH OF PST CONSTRAINT TABLE = (integer)
    BEGIN PST CONSTRAINT TABLE
        BEGIN PST CONSTRAINT DEFINITION = (character*8)
            ISSNAM = (character*8)
            INDNAM = (character*8)
            DSSNAM = (character*8)
            DNDNAM = (character*8)
        END PST CONSTRAINT DEFINITION
    END PST CONSTRAINT TABLE
    LENGTH OF PRE CONSTRAINT TABLE = (integer)
    BEGIN PRE CONSTRAINT TABLE
```

```
            BEGIN PRE CONSTRAINT DEFINITION = (character*8)
                DOFTYP = ('AIR' or 'STR')
                DOFSET = (character*8)
                DOFNUM = (integer)
                DOFVAL = (real)
            END PRE CONSTRAINT DEFINITION
        END PRE CONSTRAINT TABLE
        LENGTH OF RBC CONSTRAINT TABLE = (integer)
        BEGIN RBC CONSTRAINT TABLE
            BEGIN RBC CONSTRAINT DEFINITION =
            (character*8)
                ISSNAM = (character*8)
                INDNAM = (character*8)
                DNDNAM = (character*8)
            END RBC CONSTRAINT DEFINITION
        END RBC CONSTRAINT TABLE
        LENGTH OF RBM CONNECTIVITY TABLE = (integer)
        BEGIN RBM CONNECTIVITY TABLE
            BEGIN RMC DEFINITION = (character*8)
                SBSNAM = (character*8)
                NODNAM = (character*8)
                OFFSID = (character*8)
                DRCSID = (character*8)
            END RMC DEFINITION
        END RBM CONNECTIVITY TABLE
        LENGTH OF RFR CONSTRAINT TABLE = (integer)
        BEGIN RFR CONSTRAINT TABLE
            BEGIN RFR CONSTRAINT DEFINITION = (character*8)
                OMEGA = (real)
                FROFID = (character*8)
                FRDCID = (character*8)
            END RFR CONSTRAINT DEFINITION
        END RFR CONSTRAINT TABLE
        LENGTH OF RST CONSTRAINT TABLE = (integer)
        BEGIN RST CONSTRAINT TABLE
            BEGIN RST CONSTRAINT DEFINITION = (character*8)
                ISSNAM = (character*8)
                INDNAM = (character*8)
                DSSNAM = (character*8)
                DNDNAM = (character*8)
            END RST CONSTRAINT DEFINITION
        END RST CONSTRAINT TABLE
        LENGTH OF SCR CONSTRAINT TABLE = (integer)
        BEGIN SCR CONSTRAINT TABLE
            BEGIN SCR CONSTRAINT DEFINITION = (character*8)
                ISSNAM = (character*8)
                INDNAM = (character*8)
                DNDNAM = (character*8)
                SCOFID = (character*8)
                SCAUVI = (character*8)
                SCRROT = ('FIX' or 'FREE')
                SCRTRA = ('FIX' or 'FREE')
                CROT = (real)
                CTRA = (real)
                KROT = (real)
                KTRA = (real)
                T0 = (real)
                U0 = (real)
            END SCR CONSTRAINT DEFINITION
        END SCR CONSTRAINT TABLE
```

54

```
END SYSTEM PARAMETERS
BEGIN ELEMENT PARAMETERS
    NUMBER OF AE BEAM DEFINITIONS = (integer)
    BEGIN DEFINITIONS OF AE BEAMS
        BEGIN AE BEAM DEFINITION = (character*8)
            ASPID = (character*8)
            ESPID = (character*8)
            GEOMID = (character*8)
            ISPID = (character*8)
            ORDID = (character*8)
            ZXPID = (character*8)
        END AE BEAM DEFINITION
    END DEFINITIONS OF AE BEAMS
    NUMBER OF ASP DEFINITIONS = (integer)
    BEGIN DEFINITIONS OF AS PROPERTIES
        BEGIN AS PROPERTY DEFINITION = (character*8)
            SGNAA = (integer)
            CDTID = (character*8)
            CLTID = (character*8)
            CMTID = (character*8)
            CDX = (real array of length 4)
            CLX = (real array of length 4)
            CMX = (real array of length 4)
            TZL = (real array of length 4)
            CHD = (real array of length 4)
            RQPZ2 = (real array of length 4)
        END AS PROPERTY DEFINITION
    END DEFINITIONS OF AS PROPERTIES
    NUMBER OF AC TABLE DEFINITIONS = (integer)
    BEGIN DEFINITIONS OF AC TABLES
        BEGIN AC TABLE DEFINITION = (character*8)
            LENGTH OF AC TABLE = (integer)
            BEGIN AC TABLE ENTRIES
                BEGIN ENTRY FOR AC TABLE = (character*8)
                    ENTRY = (real array of length 3)
                END ENTRY FOR AC TABLE
            END AC TABLE ENTRIES
        END AC TABLE DEFINITION
    END DEFINITIONS OF AC TABLES
    NUMBER OF ESP DEFINITIONS = (integer)
    BEGIN DEFINITIONS OF ES PROPERTIES
        BEGIN ES PROPERTY DEFINITION = (character*8)
            E0 = (real array of length 4)
            E1 = (real array of length 4)
            E2 = (real array of length 4)
            I1 = (real array of length 4)
            I2 = (real array of length 4)
            J = (real array of length 4)
            B1 = (real array of length 4)
            B2 = (real array of length 4)
            B3 = (real array of length 4)
            D0 = (real array of length 4)
            D1 = (real array of length 4)
            D2 = (real array of length 4)
            D3 = (real array of length 4)
            D4 = (real array of length 4)
        END ES PROPERTY DEFINITION
    END DEFINITIONS OF ES PROPERTIES
    NUMBER OF G PROPERTY DEFINITIONS = (integer)
    BEGIN DEFINITIONS OF G PROPERTIES
```

```
                BEGIN GEOM PROPERTY DEFINITION = (character*8)
                    L = (REAL)
                    THETA = (real array of length 4)
                END GEOM PROPERTY DEFINITION
            END DEFINITIONS OF G PROPERTIES
            NUMBER OF ISP DEFINITIONS = (integer)
            BEGIN DEFINITIONS OF IS PROPERTIES
                BEGIN IS PROPERTY DEFINITION = (character*8)
                    MSS = (real array of length 4)
                    MM1 = (real array of length 4)
                    MM2 = (real array of length 4)
                    MOI1 = (real array of length 4)
                    MOI2 = (real array of length 4)
                END IS PROPERTY DEFINITION
            END DEFINITIONS OF IS PROPERTIES
            NUMBER OF ORD DEFINITIONS = (integer)
            BEGIN DEFINITIONS OF AEB ORDERS
                BEGIN AEB ORDER DEFINITION = (character*8)
                    ORDU1 = (integer)
                    ORDU2 = (integer)
                    ORDU3 = (integer)
                    ORDT3 = (integer)
                    NGAUSS = (integer)
                END AEB ORDER DEFINITION
            END DEFINITIONS OF AEB ORDERS
            NUMBER OF AIR MASS DEFINITIONS = (integer)
            BEGIN DEFINITIONS OF AIR MASSES
                BEGIN AIR MASS DEFINITION = (character*8)
                    AMSID = (character*8)
                END AIR MASS DEFINITION
            END DEFINITIONS OF AIR MASSES
            NUMBER OF AMS PROPERTY DEFINITIONS = (integer)
            BEGIN DEFINITIONS OF AMS PROPERTIES
                BEGIN AMS PROPERTY DEFINITION = (character*8)
                    TIP RADIUS = (real)
                    ROOT CUTOUT RADIUS = (real)
                    NGAUSS = (integer)
                END AMS PROPERTY DEFINITION
            END DEFINITIONS OF AMS PROPERTIES
            NUMBER OF RB MASS DEFINITIONS = (integer)
            BEGIN DEFINITIONS OF RB MASSES
                BEGIN RB MASS DEFINITION = (character*8)
                    MASS = (real)
                    MOI = (real array of length 3)
                END RB MASS DEFINITION
            END DEFINITIONS OF RB MASSES
        END ELEMENT PARAMETERS
        BEGIN ARRAY DATA
            NUMBER OF VECTOR DEFINITIONS = (integer)
            BEGIN DEFINITIONS OF VECTORS
                BEGIN VECTOR DEFINITION = (character*8)
                    VECTOR = (real array of length 3)
                END VECTOR DEFINITION
            END DEFINITIONS OF VECTORS
            NUMBER OF DIRCOS DEFINITIONS = (integer)
            BEGIN DEFINITIONS OF DIRCOS
                BEGIN DIRCOS DEFINITION = (character*8)
                    DCA = (real array of length 9)
                    ;
                        or
```

```
                        ERP = (real array of length 3)
                            or
                        EUR = (real array of length 3), (real)
                            or
                        TBA = (real array of length 3), (integer)
                    END DIRCOS DEFINITION
                END DEFINITIONS OF DIRCOS
            END ARRAY DATA
            BEGIN FIELD PARAMETERS
                NUMBER OF AIR DENSITY DEFINITIONS = (integer)
                BEGIN DEFINITIONS OF AIR DENSITY
                    BEGIN AIR DENSITY DEFINITION = (character*8)
                        RHO = (REAL)
                    END AIR DENSITY DEFINITION
                END DEFINITIONS OF AIR DENSITY
                NUMBER OF GRAVITY DEFINITIONS = (integer)
                BEGIN DEFINITIONS OF GRAVITY
                    BEGIN GRAVITY DEFINITION = (character*8)
                        GRAV = (real array of length 3)
                    END GRAVITY DEFINITION
                END DEFINITIONS OF GRAVITY
                NUMBER OF IM DEFINITIONS = (integer)
                BEGIN DEFINITIONS OF IM
                    BEGIN INERTIAL MOTION DEFINITION = (character*8)
                        V = (real array of length 3)
                        A = (real array of length 3)
                        OMEGA = (real array of length 3)
                    END INERTIAL MOTION DEFINITION
                END DEFINITIONS OF IM
            END FIELD PARAMETERS
        END MODEL PARAMETERS
    END CASE DATA
  END DATA FOR CASES
 END ANALYSIS DATA
END DATA
```

B. AEROELASTIC BEAM SECTION PROPERTY INTEGRALS

This appendix contains the definitions of the integrals in the "Aeroelastic Beam Elastic Section Properties" (section 2.1.3.3.2d) and the "Aeroelastic Beam Inertial Section Properties" (section 2.1.3.3.2.f). In these definitions, ξ_1 and ξ_2 are the beam cross section coordinates, λ_1 and λ_2 are the derivatives of the warp function along cross section axes, and A is the cross sectional area. The constants E, G, and ρ_s are Young's modulus, shear modulus, and structural density, respectively.

B1   aeroelastic beam cross section integral, $B_1 = \iint E\xi_1(\xi_1^2 + \xi_2^2)dA$

B2   aeroelastic beam cross section integral, $B_2 = \iint E\xi_2(\xi_1^2 + \xi_2^2)dA$

B3   aeroelastic beam cross section integral, $B_3 = \iint E(\xi_1^2 + \xi_2^2)^2 dA$

D0   aeroelastic beam cross section integral, $D_0 = \iint E(\xi_2\lambda_1 - \xi_1\lambda_2)dA$

D1   aeroelastic beam cross section integral, $D_1 = \iint E\xi_1(\xi_2\lambda_1 - \xi_1\lambda_2)dA$

D2   aeroelastic beam cross section integral, $D_2 = \iint E\xi_2(\xi_2\lambda_1 - \xi_1\lambda_2)dA$

D3   aeroelastic beam cross section integral, $D_3 = \iint E(\xi_1^2 + \xi_2^2)(\xi_2\lambda_1 - \xi_1\lambda_2)dA$

D4   aeroelastic beam cross section integral, $D_4 = \iint E(\xi_2\lambda_1 - \xi_1\lambda_2)^2 dA$

E0   aeroelastic beam axial rigidity, $E_0 = \iint E\,dA$

E1   aeroelastic beam first flexural moment about the local b_1 axis, $E_1 = \iint E\xi_1 dA$

E2   aeroelastic beam first flexural moment about the local b_2 axis, $E_2 = \iint E\xi_2 dA$

I1   aeroelastic beam bending rigidity about the local b_1 axis, $I_1 = \iint E\xi_2^2 dA$

I2   aeroelastic beam bending rigidity about the local b_2 axis, $I_2 = \iint E\xi_1^2 dA$

J   aeroelastic beam torsion rigidity, $J = \iint G((\xi_1 + \lambda_2)^2 + (\xi_2 - \lambda_1)^2)dA$

MSS   aeroelastic beam mass per unit length, $m = \iint \rho_s dA$

MM1   aeroelastic beam first mass moment of cross section about the local b_1 axis, $m_1 = \iint \rho_s \xi_1 dA$

MM2   aeroelastic beam first mass moment of cross section about the local b_2 axis, $m_2 = \iint \rho_s \xi_2 dA$

MOI1   mass moment of inertia about the local b_1 axis, $i_1 = \iint \rho_s \xi_2^2 dA$

MOI2   mass moment of inertia about the local b_2 axis, $i_2 = \iint \rho_s \xi_1^2 dA$

C. BLDINP AND GRASP RUNSTREAM SUMMARY

The Cray Operating System commands required to run BLDINP and GRASP back-to-back are summarized in this appendix. All of the data-set names and identifiers are those used on the Cray X-MP 48 at NASA Ames Research Center. Other installations may use different names and identifiers. No attempt is made in this summary to show how individual routines can be added, deleted or modified for a particular job. This information may be obtained by consulting the Cray UPDATE reference manual.

```
JOB,JN=TEST,T=10.
ACCOUNT,AC=xxxxxxx,US=xxxxxx,UPW=xxxx.
*
* Get the libraries for BLDINP
*
ACCESS,DN=BINPLIB,PDN=BINPLIB,ID=GRASPLIB.
ACCESS,DN=ERRMLIB,PDN=ERRMLIB,ID=GRASPLIB.
ACCESS,DN=INFMLIB,PDN=INFMLIB,ID=GRASPLIB.
ACCESS,DN=UTILLIB,PDN=UTILLIB,ID=GRASPLIB.
*
* Get User Data from remote VAX
*
FETCH,DN=USRDAT,TEXT='$DISK filename.DAT'.
COPYS,I=USRDAT,O=$OUT.
REWIND,DN=USRDAT.
*
* Assign FORTRAN units
*
ASSIGN,DN=USRDAT,A=FT01.
ASSIGN,DN=RESULT,A=FT06.
*
* Load and execute BLDINP
*
LDR,DN=BINPLIB,LIB=INFMLIB:ERRMLIB:UTILLIB,^
    MAP=FULL,L=MAPFILE.
RELEASE,DN=MAPFILE.
RELEASE,DN=USRDAT.
*
* Get the extra libraries for GRASP
*
ACCESS,DN=EXECLIB,PDN=EXECLIB,ID=GRASPLIB.
ACCESS,DN=MATHLIB,PDN=MATHLIB,ID=GRASPLIB.
AMESLIB.
IMSLIB.
*
* Load and execute GRASP
*
LDR,DN=EXECLIB,LIB=MATHLIB:INFMLIB:ERRMLIB:UTILLIB:IMSL,^
    MAP=FULL,L=MAPFILE.
REWIND,DN=RESULT.
COPYF,I=RESULT,O=$OUT.
RELEASE,DN=MAPFILE.
DISPOSE,DN=$OUT,DEFER,TEXT='$DISK filename.OUT'.
*
* Print results, load map, and logfile if job aborts
*
EXIT.
REWIND,DN=RESULT.
```

```
DISPOSE,WAIT,DN=RESULT,TEXT='$DISK filename.RSL'.
REWIND,DN=MAPFILE.
DISPOSE,WAIT,DN=MAPFILE,TEXT='$DISK filename.MAP'.
DISPOSE,DEFER,DN=$OUT,TEXT='$DISK filename.LOG'.
/EOF
BEGIN USER INPUT
END USER INPUT
/EOF
BEGIN USER INPUT
END USER INPUT
/EOF
```

Report Documentation Page

1. Report No. NASA TM-100043 USAAVSCOM TR 87-A-12	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle General Rotorcraft Aeromechanical Stability Program (GRASP) Version 1.03–User's Manual		5. Report Date February 1988
		6. Performing Organization Code
7. Author(s) A. Stewart Hopkins and Donald L. Kunz		8. Performing Organization Report No. A-88028
		10. Work Unit No. 992-21-01
9. Performing Organization Name and Address Ames Research Center, Moffett Field, CA 94035 and Aeroflightdynamics Directorate, U.S. Army Aviation Research and Technology Activity, Ames Research Center, Moffett Field, CA 94035-1099		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration, Washington, DC 20546-0001 and U.S. Army Aviation Systems Command, St. Louis, MO 63120-1798		
		14. Sponsoring Agency Code

15. Supplementary Notes

Point of Contact: Donald L. Kunz, Ames Research Center, MS 215-A/215-1, Moffett Field, CA 94035
(415) 694-5891 or FTS 464-5891

16. Abstract

The Rotorcraft Dynamics Division, Aeroflightdynamics Directorate, U.S. Army Aviation Research and Technology Activity has developed the General Rotorcraft Aeromechanical Stability Program (GRASP) to perform calculations that will assess the stability of rotorcraft in hovering flight and ground contact conditions. The program is designed to be a state-of-the-art, hybrid, finite-element/ multibody code that can be applied to all existing and future helicopter configurations. While GRASP was specifically designed to solve rotorcraft stability problems, its innovative structure and formulation allow for applications to a wide range of structures. This manual describes the preparation of the input file required by Version 1.03 of GRASP, the procedures used to invoke GRASP on the NASA Ames Research Center Cray X-MP 48 computer, and the interpretation of the output produced by GRASP. The parameters used by the input file are defined, and summaries of the input file and the job control language are included.

17. Key Words (Suggested by Author(s)) Finite elements Aeroelasticity Rotary wings		18. Distribution Statement Unclassified – Unlimited Subject category – 61	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 62	22. Price A04