NASA Contractor Report 2848

Modal Interpolation Program, L 215 (INTERP)

Volume II: Supplemental System Design and Maintenance Document

M. Y. Hirayama, R. I. Kroll, and R. E. Clemmons

CONTRACT NAS1-13918
SEPTEMBER 1979

NASA

NASA Contractor Report 2848

Modal Interpolation Program, L 215 (INTERP)

Volume II: Supplemental System Design and Maintenance Document

M. Y. Hirayama, R. I. Kroll, and R. E. Clemmons
Boeing Commercial Airplane Company
Seattle, Washington

Prepared for
Langley Research Center
under Contract NAS1-13918

National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

1979

CONTENTS

TABLES

FIGURES

1.0 SUMMARY

The Modal Interpolation program L215 (INTERP) is structured as ten overlays: one main, one primary and eight secondary overlays. Input into the program is made via cards and/or magnetic files (tapes or disks). Output from the program consists of printed results and magnetic files containing sorted modal data and arrays of interpolation coefficients.

Although L215 serves as a module of the DYLOFLEX system, it can be operated as a standalone program. Subroutines used by L215 include routines embedded in the program code, routines obtained from the standard FORTRAN library, and routines obtained from the DYLOFLEX library.

2.0 INTRODUCTION

The computer program L215 (INTERP) was developed for use as either a standalone program or as a module of the program system called DYLOFLEX (see fig. 1) developed for NASA under the contract NAS1-13918.(ref. 1). The Modal Interpolation program (L215) was designed to meet the DYLOFLEX contract requirements as defined in reference 2. These requirements specify the need for a program capable of using modal data to calculate displacements at several points on an aerodynamic surface. The program was developed using existing BCAC/BCS interpolation subroutines[1].

The objective of this volume is to aid those persons who will maintain and/or modify the program in the future. To meet this objective, the following items are defined:

- Program design and structure
- Overlay purpose and description
- Input, output, and internal data base descriptions
- Test cases used for program checkout

---

[1] *ATLAS - An Integral Structural Analysis and Design System - System Document., Boeing Document D6-25400-0002TN 1974.*

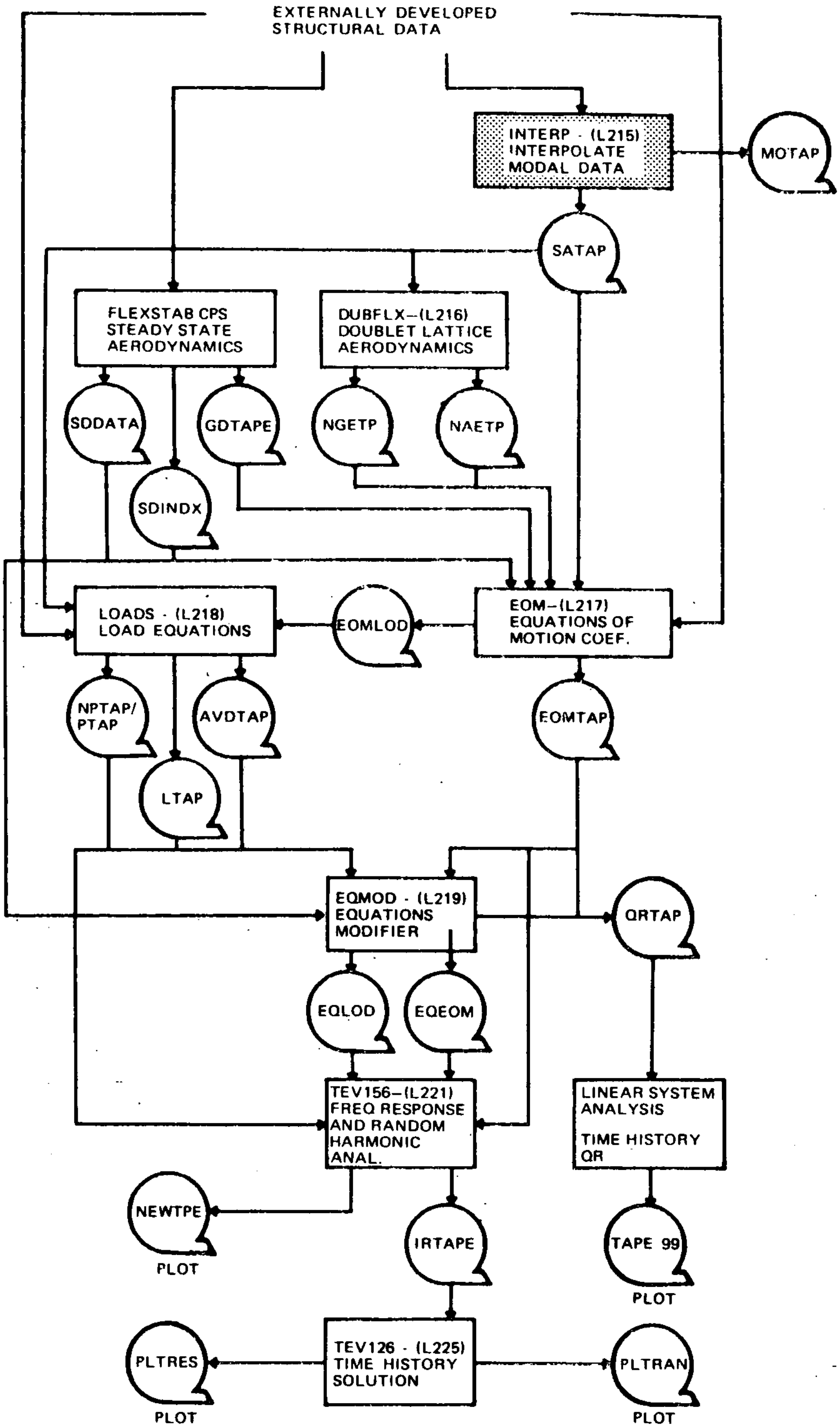


Figure 1. —DYLOFLEX Flow Chart

The program design, code, testing, and documentation were prepared according to the DYLOFLEX programming specifications[2]. The document describes the flow of each overlay in a macro sense in order to provide an overall understanding of the program's operation. The program's internal documentation (code comments) includes step-by-step descriptions of the analysis subtasks, plus complete lists of variable definition, input and output data, and routines called.

[2]Clemmons, R. E. : *Programming Specifications for Modules of the Dynamic Loads System to Interface with FLEXSTAB* NASA contract NAS1-13918, BCS-G0701, July 1975. This reference is not generally available.

3.0 PROGRAM DESIGN AND STRUCTURE

The program is constructed as a system of overlays (fig. 2). The data input and output from each overlay is displayed in figure 3.

Main overlay (L215,0,0)	L215vc
Primary overlay (L215,1,0)	INTERP
Secondary overlay (L215,1,1)	RDEDIT
Secondary overlay (L215,1,2)	BEAM
Secondary overlay (L215,1,3)	MOTA
Secondary overlay (L215,1,4)	MOTP
Secondary overlay (L215,1,5)	POLY
Secondary overlay (L215,1,6)	SURF
Secondary overlay (L215,1,7)	RESULT
Secondary overlay (L215,1,10)†	INTEMD

The main overlay L215vc simply calls the primary overlay INTERP into execution.

The 1,0 primary overlay, INTERP, controls the execution of the secondary overlays according to keywords extracted from card input data. It also aids communication between overlays via labelled common blocks.

The 1,1 secondary overlay, RDEDIT, reads and edits general card input data used for all types of modal interpolation.

The 1,2 through 1,6 secondary overlays read data and generate modal interpolation information arrays for the five different methods of interpolation that are available. The overlay number and name plus the method of interpolation are listed below:

Overlay number	Name	Method of interpolation
1,2	BEAM	Beam spline
1,3	MOTA	Motion axis

†All overlays are identified by octal numbers.

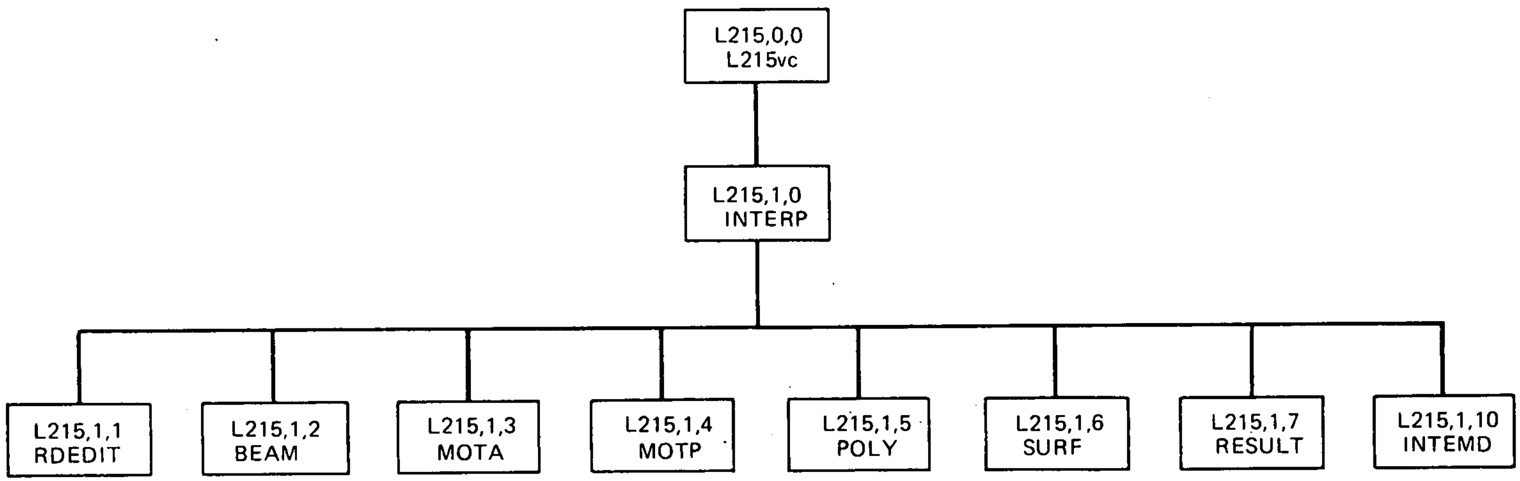
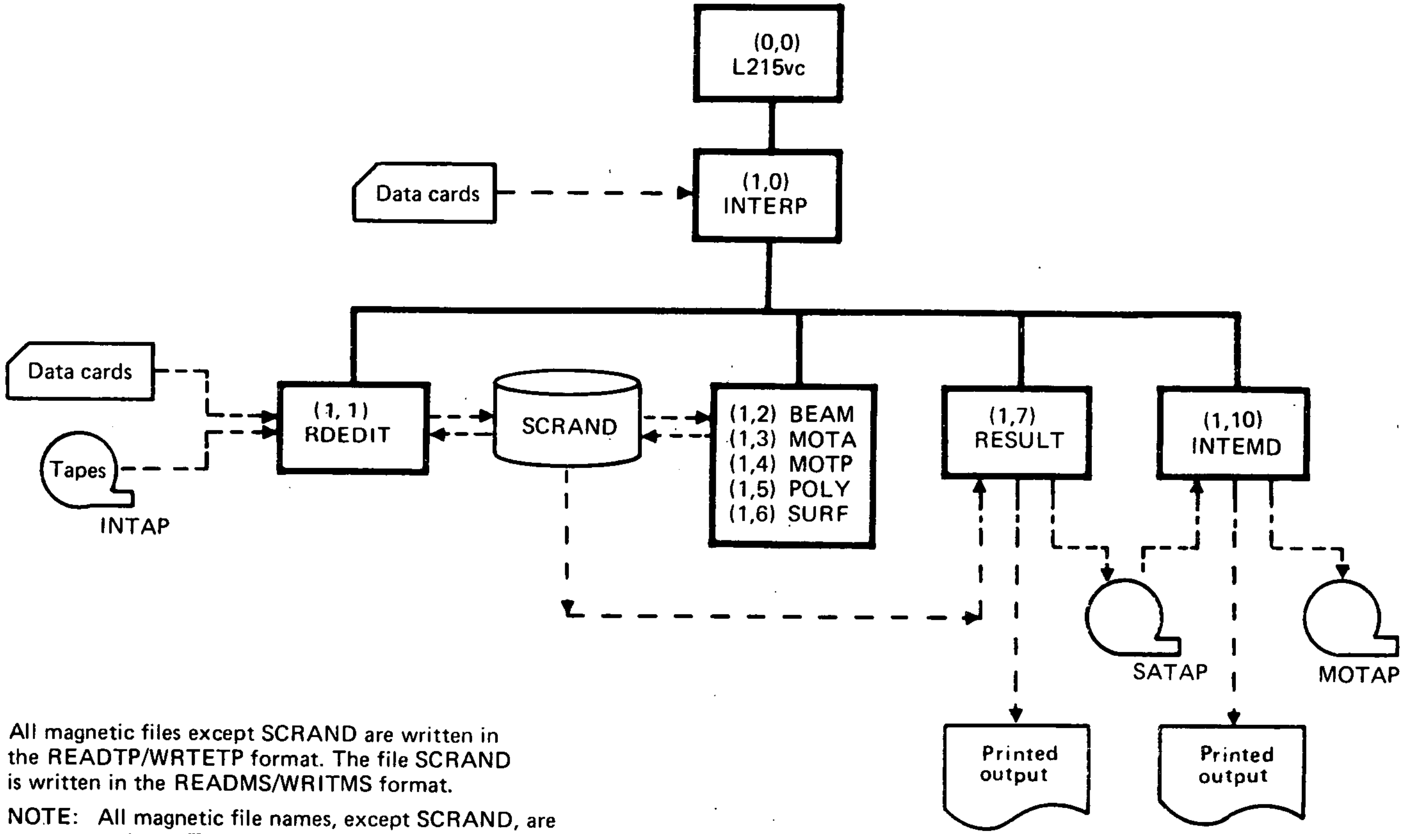


Figure 2.—Overlay Structure of L215 (INTERP)



All magnetic files except SCRAND are written in the READTP/WRTETP format. The file SCRAND is written in the READMS/WRITMS format.

NOTE: All magnetic file names, except SCRAND, are variable. The file names under the tape symbols are the ones most commonly used.

Figure 3.—Input/Output of L215's Overlays

1,4	MOTP	Motion point
1,5	POLY	Polynominal
1,6	SURF	Surface spline

The 1,7 secondary overlay, RESULT, prints the calculated results and saves them on a magnetic file if requested.

The last secondary overlay, INTEMD (1,10), interpolates for displacements at specific aerodynamic control points.

Each overlay is discussed in detail in succeeding sections. Included for each overlay are:

1. The overlay's purpose
2. The overlay's analytical steps
3. The input/output devices used
4. A macro flow chart
5. Table of subroutines called (NOTE: All subroutines have only one entry point.)

3.1 OVERLAY (L215,0,0) - L215vc

The main overlay of L215 is L215vc where v is a letter indicating the program version, and c is an integer number indicating the number of the last correction set applied to the v version.

Purpose of L215vc

L215vc simply calls the primary level overlay of the modal interpolation module into execution.

Analytical Steps of L215vc

Call overlay (L215,1,0)

I/O Devices of L215vc

None. All data communication is done in lower level (primary and secondary) overlays.

3.2 OVERLAY (L215,1,0) - INTERP

Purpose of INTERP

INTERP is the primary level overlay. Its function is to control the execution of the secondary overlays according to card input instructions. The overlay also performs certain bookkeeping tasks and aids communication between overlays via labelled common blocks.

Analytical Steps of INTERP

INTERP reads data cards and checks for the keywords contained in the list below. Opposite each keyword is its meaning to INTERP or the action it causes the program to take.

Keyword	Meaning or action caused
$INTerpolation	Indicates that the card data following is meant for L215 (INTERP)
$TITle	This card is printed before the interpolated results are printed
SATAp	Specifies the tape name where SA arrays will be stored
TMODe	Specifies the total number of output modes
$SURface	Causes the execution of RDEDIT (L215,1,1,) which will read and edit all the input data required for one of the interpolation schemes. In addition to the reading and editing of the input data, the appropriate overlay is called to generate the SA array for the selected interpolation method. The overlay called for each interpolation method is displayed below: BEAM(L215,1,2) Beam spline MOTA(L215,1,3) Motion axis MOTP(L215,1,4) Motion point POLY(L215,1,5) Polynominal SURF(L215,1,6) Surface Spline After the SA array is generated, RESULT (L215,1,7) is called to print and save the input mode shapes, input locations, and SA array
MOTApe	Specifies the tape name on which interpolated mode shapes will be stored
$MODe	Causes the execution of INTEMD (L215,1,10) which reads and edits output aerodynamic control points and generates interpolated mode shapes at those points

CHECKOUT	Causes the program to print intermediate results for debugging purposes
FLUSh	Causes program to call the subroutine FLUSH(1) when fatal errors are detected during execution of the program
$QUIt $EOR	Cause the modal interpolation program to terminate execution

The macro flow chart of INTERP is shown in figure 4. The routines called by INTERP are displayed in table 1.

I/O Devices of INTERP

INTERP reads card sets 1 through 5, 24, 25, and 30.

The overlay prints the cards it reads and diagnostics for any errors detected.

3.2.1 OVERLAY (L215,1,1) - RDEDIT

Purpose of RDEDIT

RDEDIT establishes default values of options and constants, reads and edits the general input data, and diagnoses errors. The valid input data are written onto the scratch random file SCRAND for use by the modules generating the SA arrays.

Analytical Steps of RDEDIT

RDEDIT reads data cards and checks the first few characters for keywords indicating action to be taken. The following list displays the keywords that are recognized by RDEDIT. Beside each keyword is its meaning to the program.

Keyword	Meaning or action caused
TRANsformation	Introduces coordinate transformation data rotation angles and translation values. The transformation may be from reference to local axis or from local to reference axis
NODEs	Indicates that nodal coordinates are to be input from tape or on cards in the reference or local axis system. The nodes may be reordered according to the user's specification if the keyword card MAPNodes is input
MODEs	Declares that mode shapes are to be input from tape or on cards, or that mode shapes are to be generated from a previous surface. Also, the mode shapes may be selected and reordered into single freedoms as required by the modules generating the SA arrays. Rigid surface modes

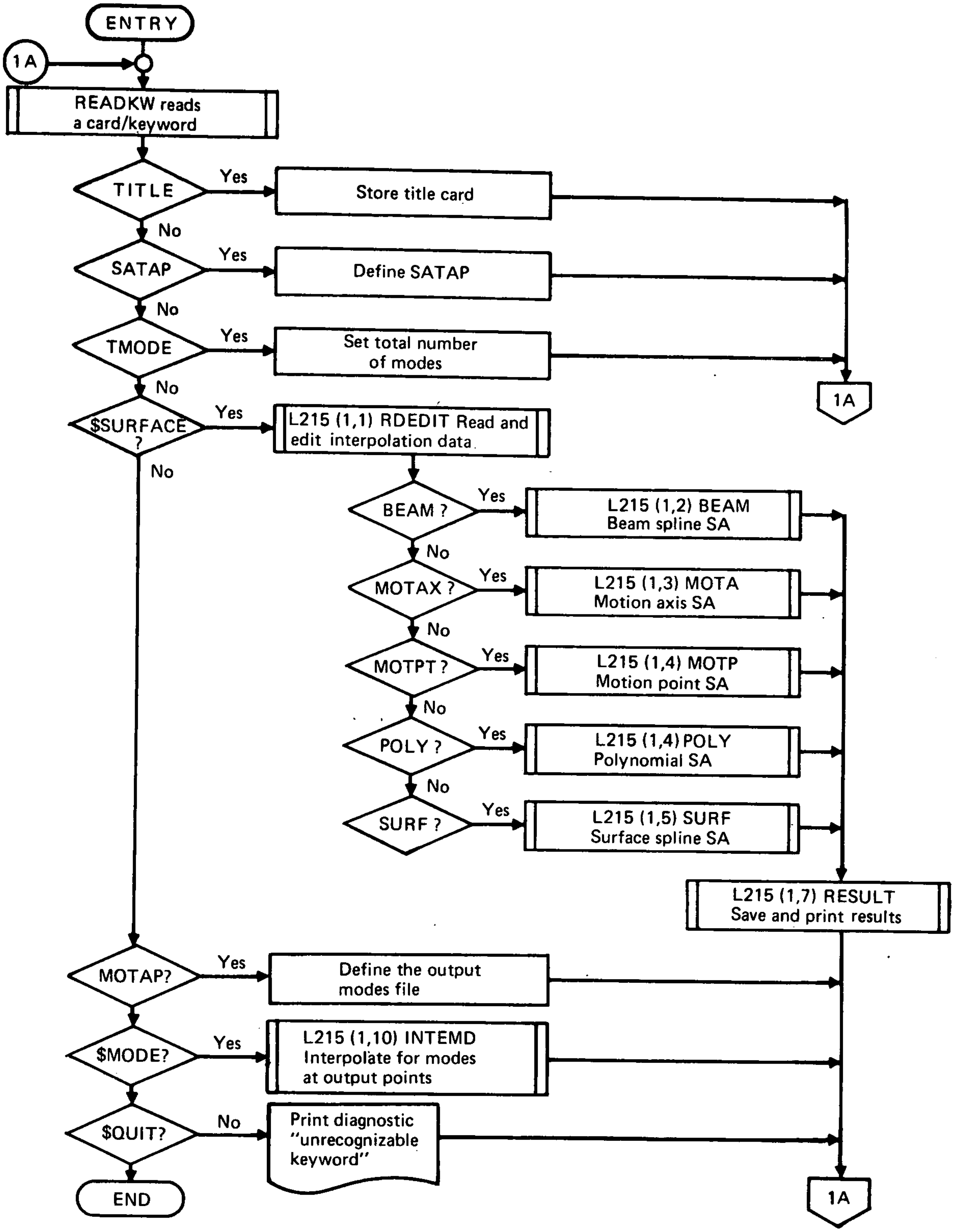


Figure 4.—Macro Flow Chart of Overlay (L215,1,0)—INTERP

Table 1.—Routines Called by Overlay (L215,1,0)—INTERP

```
OVERLAY (L215,1,0)

PROGRAM INTERP

-----------------------------------------------------------

INTERP calls { RDEDIT (Overlay L215,1,1)
               BEAM (Overlay L215,1,2)
               MOTA (Overlay L215,1,3)
               MCTP (Overlay L215,1,4)
               POLY (Overlay L215,1,5)
               SURF (Overlay L215,1,6)
               RESULT (Overlay L215,1,7)
               INTEMD (Overlay L215,1,10)
               FETAD+
               FETDEL+
               FLUSH+
               NAMFIL+
               OPENMS*
               PRGBEG+
               PRGEND+
               READKW    calls KWSRCH
               RETURNF+
---------------
```

\+ DYLOFLEX Library Routine

* CDC 6600 Library Routine

	also may be added to the sorted mode shapes at a user's request
SA	Selects the freedoms to be used in generating the SA array
BEAM	Introduces beam spline scheme data
MOTIONAxis	Introduces motion axis scheme data
MOTIONPoint	Introduces motion point scheme data
POLYnomial	Introduces polynomial scheme data
SURFace	Introduces surface spline scheme data
PRINt	Requests optional printing of matrices
$END	End of surface data, which causes RDEDIT to return to calling program

I/O Devices of RDEDIT

RDEDIT reads card sets 6 through 23. If required by the options chosen, RDEDIT will read data from the following files:

File Name (user-specified)	Data Read
NODETP	Node locations, [X,Y,Z]
MODETP	Combined freedoms and/or single freedoms
SATAP	SA array for a parent surface

RDEDIT writes the edit input data onto the scratch random file SCRAND. The edited data includes node locations, single freedoms, and instructions to the overlays generating the SA arrays with the methods:

- Beam spline
- Motion axis
- Polynomial
- Surface spline

Figure 5 is the macro flow chart of RDEDIT. The routines called by RDEDIT are displayed in table 2.

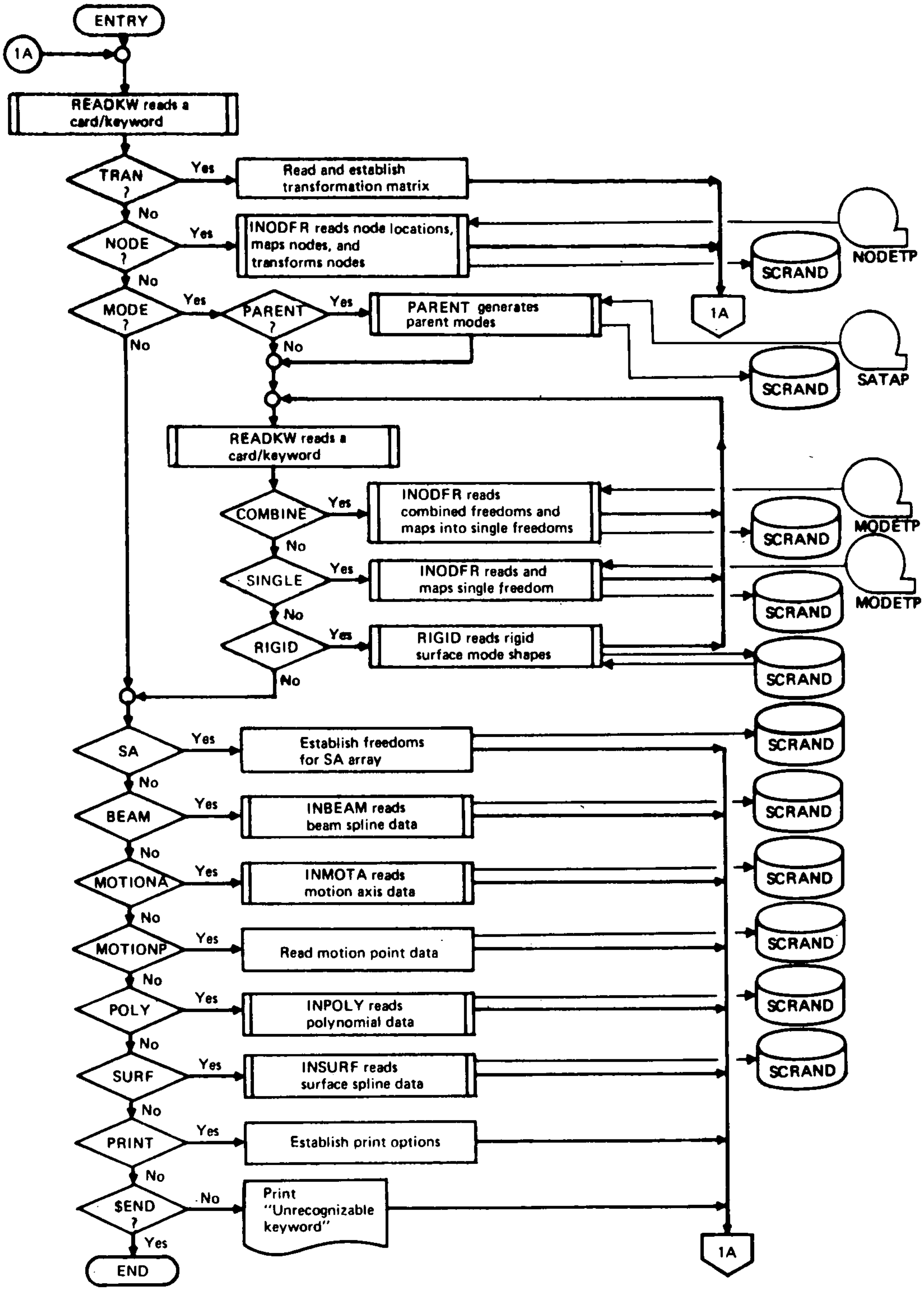


Figure 5.–Macro Flow Chart of Overlay (L215,1,1)–RDEDIT

Table 2.—Routines Called by RDEDIT

```
OVERLAY (L215,1,1)

PROGRAM RDEDIT

----------------------------------------------------------

RDEDIT calls   { INBEAM calls { WRITMS*
               | INMOTA calls { WRITMS*
               | INODFR calls { READKW   calls   KWSRCH
               |              | NAMFIL+
               |              | FETEDIT+
               |              | READTP+
               |              | READMS*
               | INPOLY calls { WRITMS*
               | INSURF calls { WRITMS*
               | ISCAN+
               | LOCF*
               | PARENT calls { RQL+
               |              | READTP+
               |              | AINTL+
               |              | READMS*
               |              | WRITMS*
               | READKW calls   KWSRCH
               | READMS*
               | REQFL+
               | RIGID calls  { READKW   calls   KWSRCH
               |              | WRITMS*
               | WRITMS*
```

\+ DYLOFLEX Library routine

* CDC 6600 Library routine

3.2.2 OVERLAY (L215,1,2) - BEAM

Purpose of BEAM

BEAM generates an array of interpolation functional coefficients (SA array) using the beam spline scheme.

Analytical Steps of BEAM

1. Read node coordinates and mode shapes from SCRAND

2. Reorder nodes on each beam to provide monotonic increasing Y-values

3. Prepare mode shapes in the reordered node sequence

4. Establish an array of values pointing to the beam positions (rows) on the node coordinate matrix

5. Call subroutine BEAMI, which generates the SA array using, as the interpolating function, cubic splines in arc length along each beam axis

6. Append the coordinate transformation matrix to the SA array

7. Write the SA array on the scratch random file SCRAND

I/O Devices of BEAM

BEAM reads from SCRAND (scratch random file) the node locations, single freedoms, and the edited beam spline input data. BEAM writes the SA array on SCRAND. Figure 6 is the macro flow chart of BEAM. The routines called by BEAM are displayed in table 3.

3.2.3 OVERLAY (L215,1,3) - MOTA

Purpose of MOTA

MOTA generates an array of interpolation functional coefficients (SA array) using the motion axis scheme.

Analytical Steps of MOTA

1. Read node coordinates and mode shapes from SCRAND

2. Reorder nodes monotonic to provide increasing Y-values

3. Prepare mode shapes in the reordered node sequence

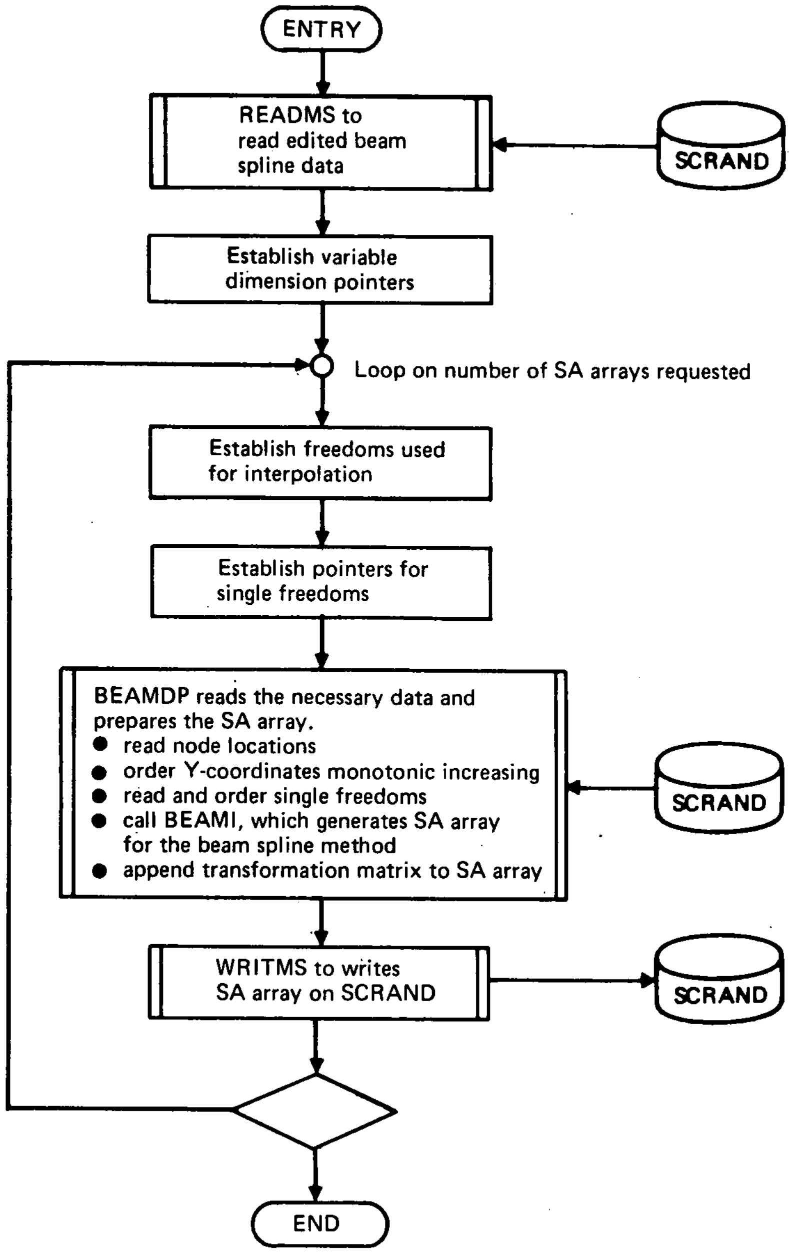


Figure 6.—Macro Flow Chart of Overlay (L215,1,2)—BEAM

Table 3.—Routines Called by BEAM

```
OVERLAY (L215,1,2)

PROGRAM BEAM

------------------------------------------------------------

BEAM calls  { BEAMDP    calls  { BEAMI+
            {                  { ORDER+
            {                  { READMS*
            {
            { LOCF*
            { READMS*
            { REQFL+
            { WRITMS*
```

\+ DYLOFLEX Library routine

* CDC 6600 Library routine

4. Call subroutine MOTAXI, which generates the SA array using as the interpolating function a cubic spline along a continuous planar curve

5. Append the coordinate transformation matrix to the SA array

6. Write the SA array on the scratch random file SCRAND

I/O Devices of MOTA

MOTA reads from SCRAND (scratch random file) the node locations, single freedoms, and the edited motion axis input data. MOTA writes the SA array on SCRAND and prints diagnostics when necessary. Figure 7 is the macro flow chart of MOTA. The subroutines called by MOTA are displayed in table 4.

3.2.4 OVERLAY (L215,1,4) - MOTP

Purpose of MOTP

MOTP generates an array of interpolation functional coefficients (SA array) using the motion point scheme.

Analytical Steps of MOTP

1. Read node coordinate and mode shapes from SCRAND

2. Call subroutine MOTPTI to generate the SA array using as the interpolation function the equations for the small angle rigid body transformation of motions from a single point

3. Append the coordinate transformation matrix to the SA array

4. Write the SA array on the scratch random file SCRAND

I/O Devices of MOTP

MOTP reads from SCRAND (scratch random file) the node locations and single freedoms. MOTP writes the SA array on SCRAND and prints diagnostics when necessary. Figure 8 is the macro flow chart of MOTP. The subroutines called by MOTP are displayed in table 5.

3.2.5 OVERLAY (L215,1,5) - POLY

Purpose of POLY

POLY generates an array of interpolation functional coefficients (SA array) using the polynomial scheme.

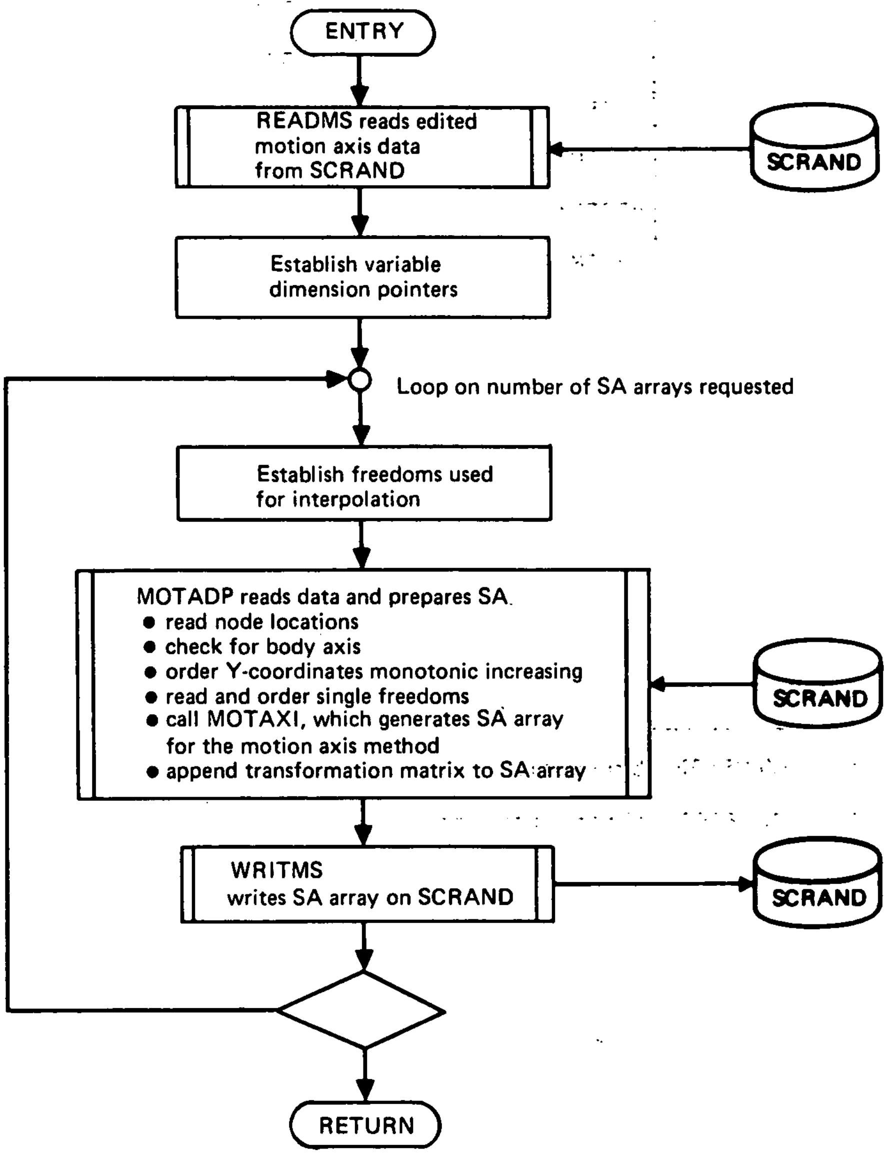


Figure 7.—Macro Flow Chart of Overlay (L215,1,3)—MOTA

Table 4.—Routines Called by MOTA

```
OVERLAY (L215,1,3) PROGRAM MOTA

-----------------------------------------------------------

MOTA calls   { MOTADP    calls   { MOTAXI+
             |                   | ORDER+
             |                   { READMS*
             | LOCF*
             | READMS*
             | REQFL+
             { WRITMS*
```

\+ DYLOFLEX Library routine

* CDC 6600 Library routine

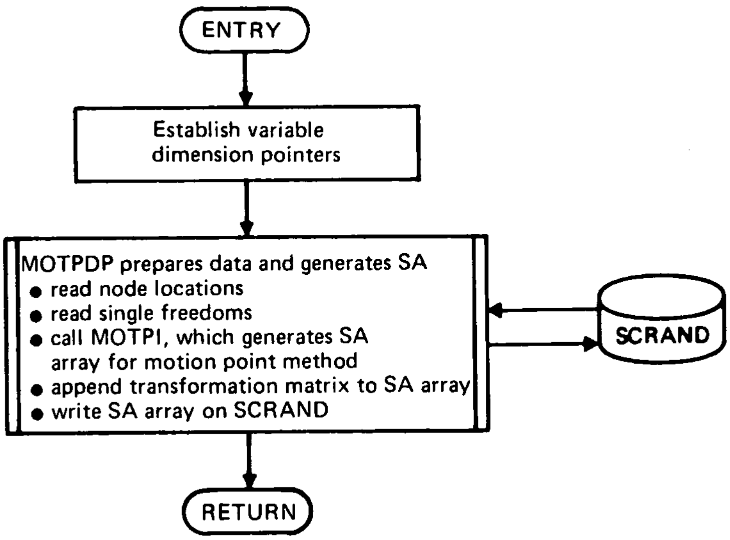


Figure 8.—Macro Flow Chart of Overlay (L215,1,4)—MOTP

Table 5.—Routines Called by MOTP

```
OVERLAY (L215,1,4) PROGRAM MOTP

------------------------------------------------------------

MOTP calls  { MOTPDP    calls   { READMS*
            {                   { WRITMS*
            {                   { MOTPTI+
            { LOCF*
            { REQFL+

---------------
```

\+ DYLOFLEX Library routine

* CDC 6600 Library routine

Analytical Steps of POLY

1. Read the polynomial coefficients from SCRAND

2. Call the subroutine POLYI to generate the SA array using as the interpolating function a set of polynomial coefficients

3. Append the coordinate transformation matrix to the SA array

4. Write the SA array on the scratch random file SCRAND

I/O Devices of POLY

POLY reads from SCRAND (scratch random file) the edited polynomial data. POLY writes the SA array on SCRAND and prints diagnostics when necessary. Figure 9 is the macro flow chart of POLY. The subroutines called by POLY are displayed in table 6.

3.2.6 OVERLAY (L215,1,6) - SURF

Purpose of SURF

SURF generates an array of interpolation functional coefficients (SA array) using the surface spline scheme.

Analytical Steps of SURF

1. Read node coordinates and mode shapes from SCRAND

2. Call subroutine PLATEI to generate the SA array using as the interpolating function the small deflection equation of an infinite pinned plate

3. Append the coordinate transformation matrix to the SA array

4. Write the SA array on the scratch random file SCRAND

I/O Devices of SURF

SURF reads from SCRAND the edited surface spline data, node locations, and single freedoms. SURF writes the SA array onto SCRAND and prints diagnostics when necessary. Figure 10 is the macro flow chart of SURF. The subroutines called by SURF are displayed in table 7.

3.2.7 OVERLAY (L215,1,7) - RESULT

Purpose of RESULT

RESULT writes the input mode shapes, input node locations (reference axis), and the SA arrays onto the file SATAP. Optionally, it prints the same information.

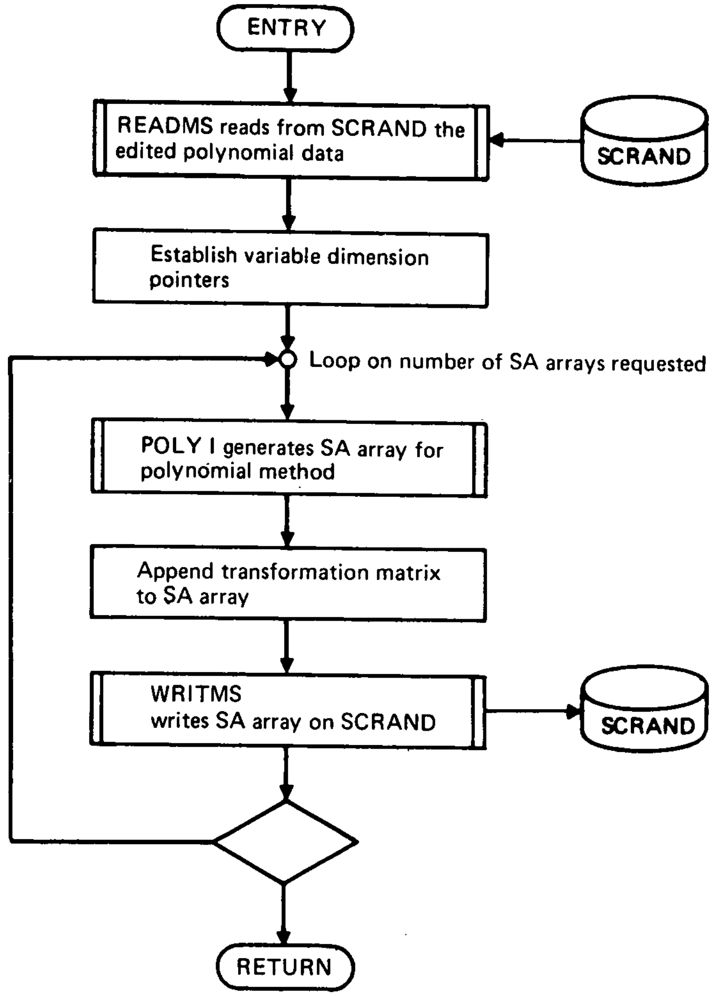


Figure 9.—Macro Flow Chart of Overlay (L215,1,5)—POLY

Table 6.—Routines Called by POLY

```
OVERLAY (L215,1,5) PROGRAM POLY
------------------------------------------------------------
POLY calls  { POLYI+
              LOCF*
              READMS*
              REQFL+
              WRITMS*
```

\+ DYLOFLEX Library routine

* CDC 6600 Library routine

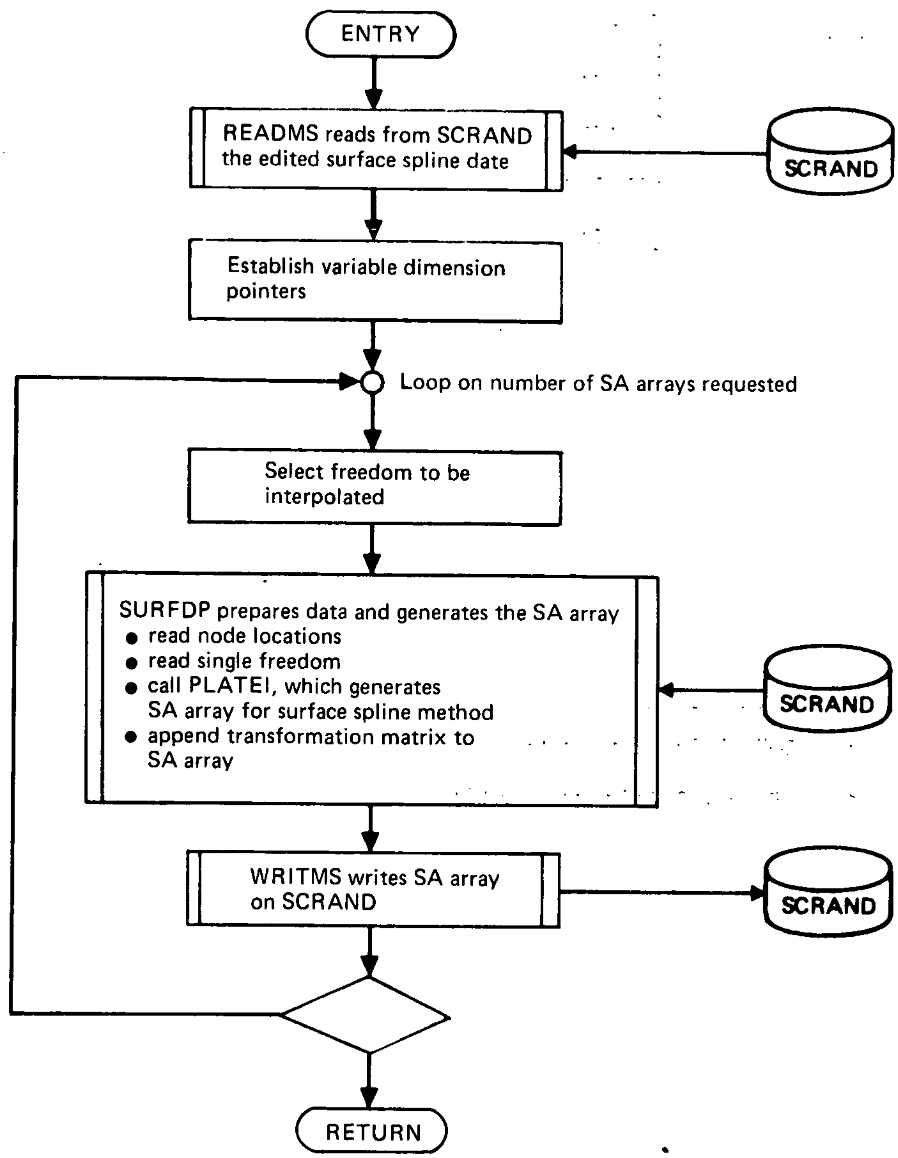


Figure 10.—Macro Flow Chart of Overlay (L215,1,6)—SURF

Table 7.—Routines Called by SURF

```
OVERLAY (L215,1,6) PROGRAM SURF

-----------------------------------------------------------------

SURF calls  { SURFCP    calls  { PLATEI+
                               { READMS*
            { LOCF*
            { READMS*
            { REQFL+
            { WRITMS*
```

\+ DYLOFLEX Library routine

* CDC 6600 Library routine

Analytical Steps of RESULT

1. Read single freedoms from SCRAND, write them onto SATAP, and optionally print them

2. Read node locations from SCRAND, write them onto SATAP, and optionally print them

3. Read the SA arrays from SCRAND, write them onto SATAP, and optionally print them

I/O Devices of RESULT

RESULT calls READMS to read from SCRAND the single freedoms, node locations, and SA arrays. RESULT calls WRTETP to write on SATAP the single freedoms, node locations, and SA arrays. Optionally, RESULT prints the same information written onto SATAP. Figure 11 is the macro flow chart of RESULT. The subroutines called by RESULT are displayed in table 8.

3.2.8 OVERLAY (L215,1,10) - INTEMD

Purpose of INTEMD

INTEMD transforms the information in the SA array into translational displacements and slopes at the given aerodynamic control points.

Analytical Steps of INTEMD

1. Read and edit aerodynamic transformation and output the aerodynamic control point location data

2. Read the selected SA array for the given aerodynamic points

3. Call AINTL for local axis or AINTG for reference axis to generate translational displacements and slopes at the given aerodynamic control points

4. Print and save the aerodynamic control points and translational displacements and slopes

I/O Devices of INTEMD

INTEMD reads card sets 26 through 29. It also reads the SA array(s) from SATAP. Optionally, INTEMD reads the aerodynamic control point locations from the file AEROTP. INTEMD writes the aerodynamic control point locations and interpolated translational displacements and slopes onto the file named MOTAP. Optionally, the same information is printed. Figure 12 is the macro flow chart of INTEMD. The subroutines called by INTEMD are displayed in table 9.

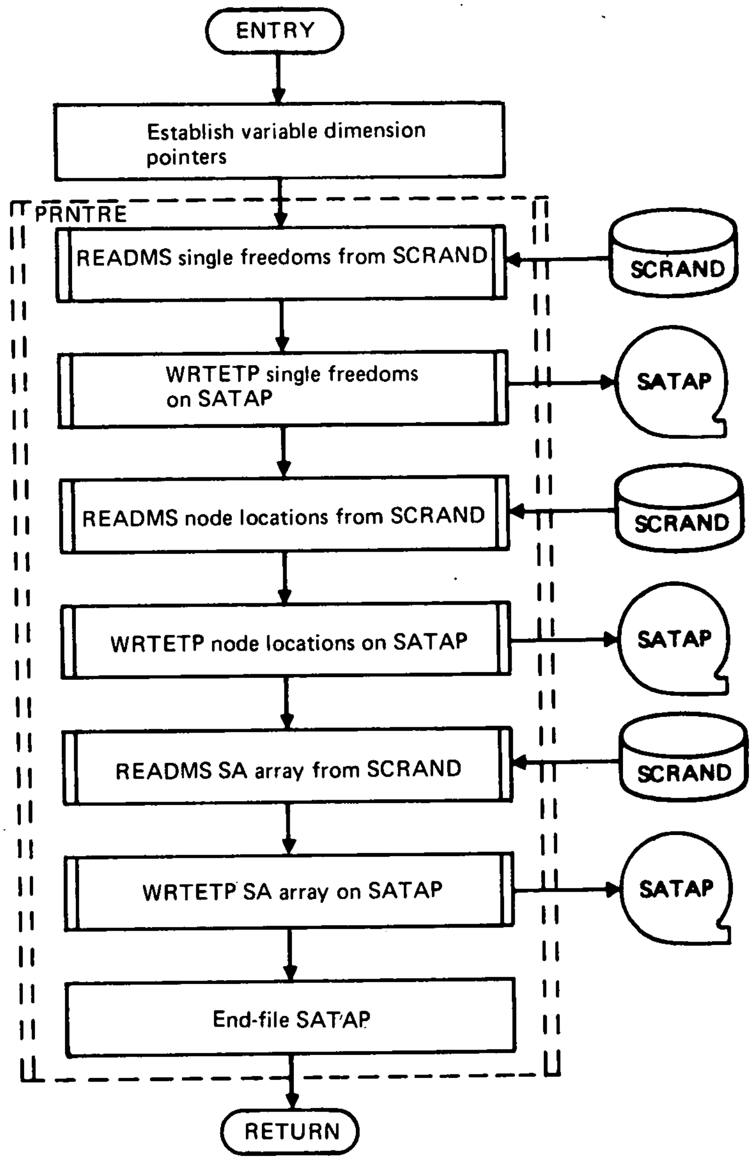


Figure 11.—Macro Flow Chart of Overlay (L215,1,7)—RESULT

Table 8.—Routines Called by RESULT

```
OVERLAY (L215,1,7) PROGRAM RESULT

-----------------------------------------------------

RESULT calls { PRNTRE    calls   { RQL+
             {                   { PRNTFR
             {                   { PRNTSA
             {                   { PRNTXY
             {                   { WRTETP+
             {                   { READMS*
             { LOCF+
             { REQFL+
```

\+ DYLOFLEX Library routine

* CDC 6600 Library routine

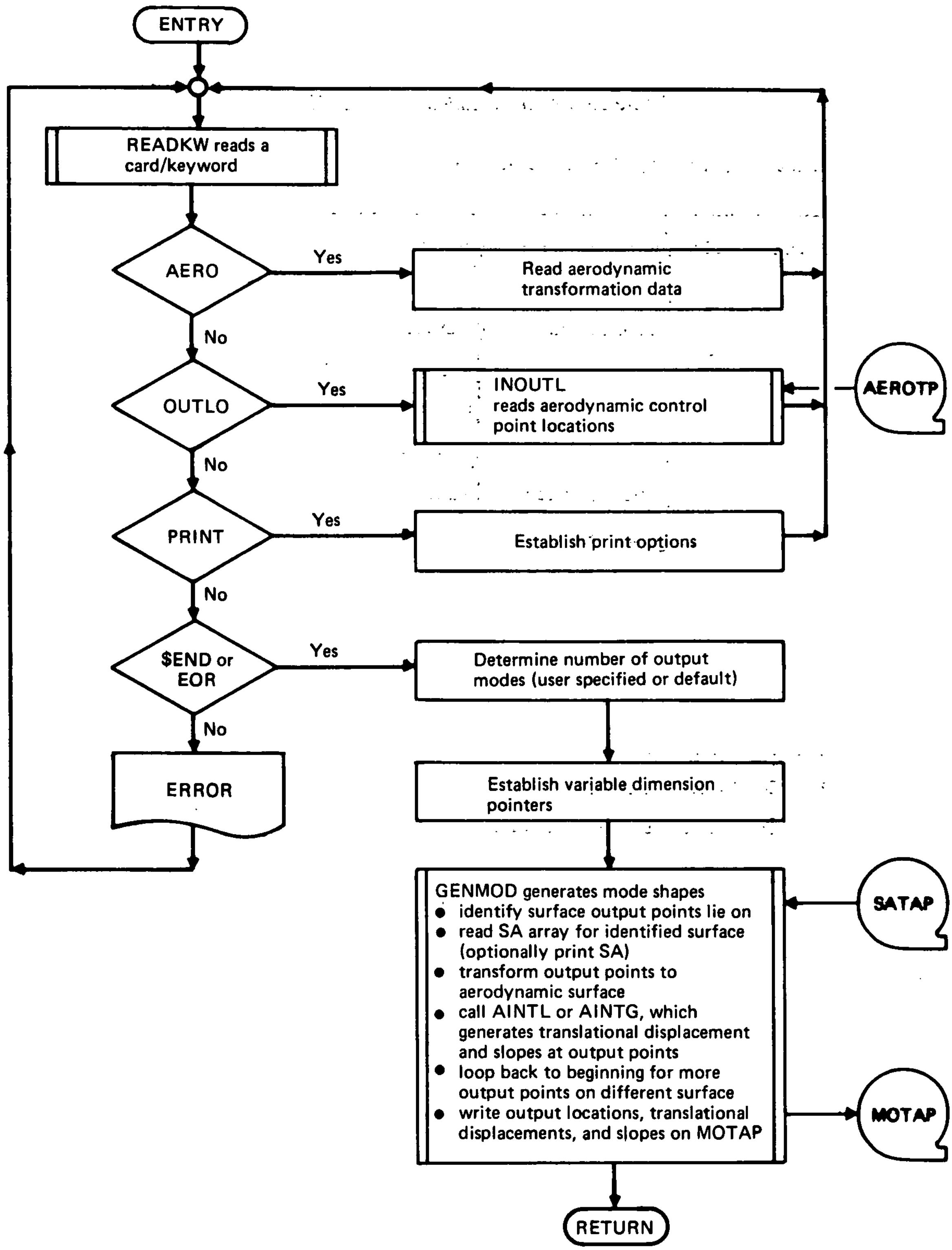


Figure 12.—Macro Flow Chart of Overlay (L215,1,10)—INTEMD

Table 9.—Routines Called by INTEMD

```
OVERLAY (L215,1,10 ) PROGRAM INTEMD

-----------------------------------------------------------

INTEMD calls { READKW calls { KWSRCH
             {
             { GENMOD calls { READTP+
             {              { WRTETP+
             {              { AINTG+
             {              { AINTL+
             { INOUTL calls { READTP+
             { READTP+
```

\+ DYLOFLEX Library routine

* CDC 6600 Library routine

3.3 DATA BASES

The Modal Interpolation program (L215) data bases include input and output files plus internal random scratch (temporary) storage files and labelled common blocks.

3.3.1 INPUT

Data is input to the program in two forms: data cards and/or magnetic files.

Card Input Data

For a complete description of L215 card input data, see section 6.5.1 in volume I of this document (Engineering and Usage).

Magnetic File Input Data

For a complete description of L215 disk or tape input data, see section 6.5.2 in volume I of this document (Engineering and Usage). The files read have user-specified names.

3.3.2 OUTPUT

The results of an L215 execution will be printed and written on magnetic files.

Printed Output Data

For a complete description of the printed output data, see section 6.6.1 in volume I of this document (Engineering and Usage).

Magnetic File (Tape or Disk)

For a complete description of the magnetic file output data, see section 6.6.1. in volume I of this document (Engineering and Usage).

3.3.3 INTERNAL

Two methods are used to pass data from one portion of the program to another: a scratch (temporary) random file and labelled common blocks.

Magnetic File (Scratch Random File)

L215 uses a scratch random file for temporary storage of data. The name of the file is SCRAND. All data is written onto SCRAND with the subroutine WRITMS. Later, the data is read with the subroutine READMS. Table 10 shows the matrices which are stored temporarily on SCRAND. Following the table, the contents of each matrix is described.

Table 10.—Matrices Written on SCRAND

MATRIX DESCRIPTION	INDEX NAME
Edited Input Data for Beam Spline	BS
Edited Input Data for Motion Axis	MA
Edited Input Data for Polynomial	PL
Edited Input Data for Surface Spline	SS
Coordinates Local Axis	XYL
Coordinates Reference Axis	XYR
Translation-X Freedoms	TX
Translation-Y Freedoms	TY
Translation-Z Freedoms	TZ
Rotation-X Freedoms	RX
Rotation-Y Freedoms	RY
Rotation-Z Freedoms	RZ
SA Arrays	SA_i (i=1,6)

Edited Input Data for Beam Spline

<u>File:</u> SCRAND

<u>Index Name:</u> BS

<u>Dimensions:</u> N*1 where N is the total number of elements (see following list)

<u>Elements:</u>

Items 1: NBEAM, number of beams

Item 2 through (NBEAM+1):

Number of points on each beam, NPB_i (i = 1,NBEAM)

Items (NBEAM + 2) through (2* NBEAM+2):

Extrapolation indicator for each beam,

$IEXTRP_i$ (i = 1,NBEAM)

Items (2*NBEAM + 3) through (2*NBEAM + 2+NPTS):

Node locations on each beam, IP_j (j=1,NPTS)

$$NPTS = \sum_{i=1}^{NBEAM} NPB_i$$

<u>Generation:</u> Subroutine INBEAM

Edited Input Data for Motion Axis

<u>File:</u> SCRAND

<u>Index Name:</u> MA

<u>Dimensions:</u> N*1 where N is the total number of elements (see following list)

<u>Elements:</u>

Items 1: NDEF, number of definition points

Items 2: IORIEN, indicator for rotation R_X orientation

Items 3 through (2+NDEF):

Slopes at reference line at definition points, $DYDXRL_i$ (i=1,NDEF)

Items (3+NDEF) through (2+2*NDEF):

X-coordinate of definition points,
XRL_i (i=1,NDEF)

Items(3+2*NDEF) through (2+3*NDEF):

Y-coordinate of definition points,

YRL_i (i=1,NDEF)

<u>Generation:</u> Subroutine INMOTA

Edited Input Data for Polynomial

File: SCRAND

Index Name: PL

Dimensions: N*1 where N is the total number of elements (see following list)

Elements:

Item 1: IORD, order (degree) of polynomial

Items 2 through (1+NPCOEF):

Polynomial coefficients, C_i(i=1,NPCOEF)

$$NPCOEF = \frac{(IORD+1)\ (*IORD+2)}{2}$$

Generation: Subroutine INPOLY

Edited Input Data for Surface Spline

File: SCRAND

Index Name: SS

Dimensions: N*1 where N is the total number of elements (see following list).

Elements:

Item 1: NSMTH, number of smoothing values

Items 2 through (1+NSMTH):

Smoothing values, $SMTH_i$(i+1,NSMTH)

Generation: Subroutine INSURF

Coordinates in Local Axis

File: SCRAND

Index Name: XYL

Dimensions: NNODES*6 where NNODES is the number of nodes

Elements:

A typical row of the coordinates in local axes contains:

Item 1: Node X-coordinate local axis

Item 2: Node Y-coordinate local axis

Item 3: Node Z-coordinate local axis

Items 4 through 6: Zeros

Generation: Subroutine RDEDIT

Coordinates in Reference Axis

File: SCRAND

Index Name: XYR

Dimensions: NNODES*6 where NNODES is the number of nodes

Elements:

A typical row of the coordinates in reference axes contains:

Item 1: Node X-coordinate reference axis

Item 2: Node Y-coordinate reference axis

Item 3: Node Z-coordinate reference axis

Item 4: Rotation angle about X-axis in degrees

Item 5: Rotation angle about Y-axis in degrees

Item 6: Rotation angle about Z-axis in degrees

Generation: Subroutine RDEDIT,

Translation-X Freedoms

File: SCRAND

Index Name: TX

Dimensions: NNODES*NTMODE

where: NNODES = the number of nodes

NTMODE = the number of modes

Elements:

Item (i, j): Contains the translation-x freedom of the ith node for the jth mode

Generation: Subroutine RDEDIT

Translation-Y Freedoms

File: SCRAND

Index Name: TY

Dimensions: NNODES*NTMODE

where: NNODES = the number of nodes

NTMODE = the number of modes

Elements:

Item (i, j): Contains the translation-Y freedom of the ith node for the jth mode

Generation: Subroutine RDEDIT

Translation -Z Freedoms

File:	SCRAND
Index Name:	TZ
Dimensions:	NNODES*NTMODE
where:	NNODES = the number of nodes
	NTMODE = the number of modes
Elements:	
Item (i, j):	Contains the translation-Z freedom of the ith node for the jth mode
Generation:	Subroutine RDEDIT

Rotation-X Freedoms

File:	SCRAND
Index Name:	RX
Dimensions:	NNODES*NTMODE
where:	NNODES = the number of nodes
	NTMODE = the number of modes
Elements:	
Item (i, j):	Contains the rotation-X freedom of the ith node for the jth mode
Generation:	Subroutine RDEDIT

Rotation-Y Freedoms

File:	SCRAND
Index Name:	RY
Dimensions:	NNODES*NTMODE
where:	NNODES = the number of nodes
	NTMODE = the number of modes
Elements:	
Item (i, j):	Contains the rotation-Y freedom of the ith node for the jth mode
Generation:	Subroutine RDEDIT

Rotation-Z Freedoms

File:	SCRAND
Index Name:	RZ
Dimensions:	NNODES*NTMODE
where:	NNODES = the number of nodes
	NTMODE = the number of modes

Elements:

Item (i, j): Contains the rotation-Z freedom of the ith node for the jth mode

Generation: Subroutine RDEDIT

SA Array for Beam Spline

File: SCRAND

Index Name: SA_i (where i may be from one to six)

Dimensions: M*1

where:

M = 17*6*NNODES+MAX2 + (INDC+3)/2)
*2*NNODES*NTMODE

NNODES = Number of nodes

NBEAM = Number of beams

INDC = Indicator for rotation motions
= 1, x-rotation
= 2, y-rotation
= 3, both x- and y-rotations

NTMODE = Number of modes (MCOLN, see following, item 6)

MAX2 = maximum of 8* (largest number of points on a beam) and 13*NBEAM

Elements:

Item 1: M – number of elements in this matrix

Item 2: 10HBEAMSPLINE

Item 3: IPOINT = pointer to transformation matrix; if IPOINT = 0, no transformation matrix

Item 4: MCOLS; total number of modes

Item 5: MCOL1; modes up to MCOL1 but not including MCOL1 are zeros

Item 6: MCOLN; (MCOLN-MCOL1+1) is the number of input modes; modes MCOLN+1 through MCOLS are zeros

Item 7: NPTS, the sum of the number of points defining all beams; NPTS≥4

Item 8: NBMS, the total number of beams defined for the analysis; NBMS≥2

Item 9: INDC, indicator for retained freedoms present in this array

INDC = 0,TZ only
= 1,TZ and RX
= 2,TZ and RY
= 3,TZ, RX, and RY

Items 10 through 15:

Reserved for future use

Items 16 through (15 + NBMS +1):

Beam pointer array; the ith element of this array points to elements of other arrays corresponding to the first point of the ith beam specified

Items (15 + NBMS +2) through (15 + 2*NBMS + 1):

Beam extrapolation code array; the ith element of this array contains the extrapolation code for the ith beam

Items (15 + 2*NBMS + 2) through (15 + 2*NBMS + NPTS + 1):

Input point Y-coordinates

Items (15 + 2*NBMS + NPTS + 2) through (15 + 2*NBMS + 2*NPTS + 1):

Arc length array

Items (15 + 2*NBMS + 2*NPTS + 2) through (15 + 2*NBMS + 2*NPTS + NSEG + 1):

C_0, the first cubic coefficient for the cubic splines defined on the beam segments

where: NSEG = NPTS-NBMS

Items (15 + 2*NBMS + 2*NPTS + NSEG + 2) through (15 +2*NBMS + 2*NPTS + 2*NSEG + 1):

C_1, the second cubic coefficient for the cubic splines defined on the beam segments

Items (15 + 2*NBMS + 2*NPTS + 2*NSEG + 2) through (15 + 2*NBMS + 2*NPTS + 3*NSEG + 1):

C_2, the third cubic coefficent for the cubic splines defined on the beam segments

Items (15 + 2*NBMS + 2*NPTS + 3*NSEG + 2) through (15 +2*NBMS + 2*NPTS + 4*NSEG + 1):

C_3, the fourth cubic coefficient for the cubic splines defined on the beam segments

Items (15 + 2*NBMS + 2*NPTS + 4*NSEG + 2) through (15 + 2*NBMS + 2*NPTS + 4*NSEG +NDEF + 1):

Z-translation mode shapes

where: NDEF = NPTS * number of modes

Items (15 + 2*NBMS + 2*NPTS + 4*NSEG + 2*NDEF + 2) through (15 + 2*NBMS + 2*NPTS + 4*NSEG + 3*NDEF + 1)

Z-rotation (slopes) mode shapes

Items (15 + 2*NBMS + 2*NPTS + 4*NSEG + 2*NDEF + 2) through (15 + 2*NBMS + 2*NPTS + 4*NSEG + 3*NDEF + 1):

X-translation mode shapes

Items (15 + 2*NBMS + 2*NPTS + 4*NSEG + 3*NDEF + 2) through (15 + 2*NBMS + 2*NPTS + 4*NSEG + 4*NDEF + 1):

X-rotation mode shapes.

Items (15 +2*NBMS + 2*NPTS + 4*NSEG + 4*NDEF + 2) through (15 + 2*NBMS + 2*NPTS + 4*NSEG + 5*NDEF + 1):

Y-translation mode shapes

Items (15 + 2*NBMS + 2*NPTS + 4*NSEG + 5*NDEF + 2) through (15 + 2*NBMS + 2*NPTS + 4*NSEG + 6*NDEF + 1):

Y-rotation mode shapes

Items (15 + 2*NBMS + 2*NPTS + 4*NSEG + 6*NDEF + 2) through (15 + 2*NBMS + 2*NPTS + 4*NSEG + 6*NDEF + 15*NBMS + 1):

Scratch area for temporary storage.

Item (15 + 2*NBMS + 2*NPTS + 4*NSEG + 6*NDEF + 15*NBMS + 2):

10HSURFSPLINE

Generation: Subroutine BEAM

SA Array for Motion Axis

File: SCRAND

Index Name: SA (where i may be from one to six)

Dimension: M*1

where M = 9 + 10*NDEF + NNODES + 6*NNODES* NTMODE + 3*NTMODE

NDEF = Number of motion axis definition points

NNODES = Number of motion stations (nodes)

NTMODE = Number of modes (MCOLN; see the following, item 6)

Elements:

Item 1: M, the number of items in this matrix

Item 2: 10HMOTIONAXIS

Item 3: IPOINT = pointer to the transformation matrix

If IPOINT = 0, no transformation matrix

Item 4: MCOLS, total number of modes

Item 5: MCOL1; modes 1 through MCOL1 will be zero on output

Item 6: MCOLN; (MCOLN-MCOL1+1) is the number of input modes and modes MCOLN + 1 through MCOLS will be zero on output

Item 7: NMADP, number of motion axis definition points

Item 8: NMS, number of motion stations

Items 9 through (8 + NMADP):

XMA, x-coordinates of the motion axis definition points

Items (8 + NMADP + 1) through (8 + 2*NMADP):

XMA, y-coordinates of the motion axis definition points

Items (8 + 2*NMADP + 1) through (8 + 3*NMADP):

DYDXRL, slope dy/dx of the reference lines through the definition points

Items (8 + 3*NMADP + 1) through (8 + 4*NMADP):

SMA, arc length along motion axis for the ith definition points

Items (8 + 4*NMADP +1) through (8 + 4*NMADP + NSEG):

XMAP, x mapping point for the ith segment

Items (8 + 4*NMADP + NSEG + 1) through (8 + 4*NMADP + 2*NSEG):

YMAP, y mapping point for ith segment

Items (8 + 4*NMADP + 2*NSEG + 1) through (8 + 4*NMADP + 3*NSEG):

C_0, cubic coefficient for the ith segment

Items (8 + 4*NMADP + 3*NSEG + 1) through (8 + 4*NMADP + 4*NSEG):

C_1, cubic coefficient for the ith segment

Items (8 + 4*NMADP + 4*NSEG + 1) through (8 + 4*NMADP + 5*NSEG):

C_2, cubic coefficient for the ith segment

Items (8 + 4*NMASP + 5*NSEG + 1) through (8 + 4*NMADP + 6*NSEG):

C_3, cubic coefficient for the ith segment

Items (8 + 4*NMADP + 6*NSEG) through (8 + 4*NMADP + 6*NSEG + NMS):

Sms, arc length along motion axis for the ith motion station

The next block of data contains the modal displacements at the ith input point for the jth mode; (N1= 8 + 4*NMADP + 6*NSEG + NMS)

Items (N1 + 1) through (N1 + NMS*NCOLS):

TZ

Items (N1 + NMS*NCOLS + 1) through (N1 + 2*NMS*NCOLS):

RX

Items (N1 + 2*NMS*NCOLS + 1) through (N1 + 3*NMS*NCOLS):

RY

Items (N1 + 3*NMS*NCOLS + 1) through (N1 + 4*NMS*NCOLS):

dTz/ds

Items (N1 + 4*NMS*NCOLS + 1) through (N1 + 5*NMS*NCOLS):

dRx/ds

Items (N1 + 5*NMS*NCOLS + 1) through (N1 + 6*NMS*NCOLS):

dRy/ds

Items (M1 + 1) through (M1 + 3*NCOLS):

scratch area where M1 = N1 + 6*NMS*NCOLS

Items (M1 + 3*NCOLS + 1) through (M1 + 3*NCOLS + ITRAN):

transformation matrix location (if specified)
where ITRAN = 12 if matrix exists
= 0 if matrix does not exist

Item (M1 + 3*NCOLS + ITRAN + 1):

10HMOTIONAXIS

Generation: Subroutine MOTA

SA Array for Motion Point

File: SCRAND

Index Name: SA_i (where i may be from one to six)

Dimensions: M*1

where: M = 10 +6*NTMODE

NTMODE = Number of modes (MCOLN, see following, item 6)

Elements:

Item 1: M, number of items in this matrix

Item 2: 8HMOTIONPT

Item 3: IPOINT = pointer to the transformation matrix;
if IPOINT = 0, no transformation matrix

Item 4: MCOLS, total number of modes

Item 5: MCOL1; modes 1 through MCOL1 will be zero on output

Item 6: MCOLN; (MCOLN-MCOL 1 + 1) is the number of input modes and modes MCOLN + through MCOLS will be zero on output

Item 7: X, reference point x-coordinate

Item 8: Y, reference point y-coordinate

Item 9: Z, reference point z-coordinate

Item 10: TX, translation in X

Item 11: TY, translation in Y

Item 12: TZ, translation in Z

Item 13: RX, rotation in X

Item 14: RY, rotation in Y

Item 15: RZ, rotation in Z

Items 16 through (9 +6*NCOLS):

The translation and rotations are repeated for each mode; (NCOLS = NCOLN-MCOL1 + 1)

Items (9 + 6*NCOLS + 1) through (9 +6*NCOLS + ITRAN):

Transformation matrix location (if specified)

where: ITRAN = 12 if matrix exists
= 0 if matrix does not exist

Item (9 + 6*NCOLS + ITRAN + 1):

8HMOTIONPT

Generation: Subroutine MOTP

SA Array for Polynomial

File: SCRAND

Index Name: SA_i (where i may be from one to six)

Dimensions: M*1

where: $M = 8 + NTMODE * \frac{(IORD + 1)*(IORD + 2)}{2}$

IORD = Degree (order) of polynomial IDEG, see following item 7)

NTMODE = Number of modes (MCOLN, see below)

Elements:

Item 1: M, number of items in this matrix

Item 2: 10HPOLYNOMIAL

Item 3: IPOINT = pointer to the transformation matrix; if IPOINT = 0, no transformation matrix

Item 4: MCOLS, total number of modes

Item 5: MCOL1; modes 1 through MCOL1 will be zero on output

Item 6: MCOLN; (MCOLN-MCOL1 + 1) is the number of input modes, and modes MCOLN + 1 through MCOLS will be zero on output

Item 7: IDEG, the highest degree of polynomial

Items 8 through (7 + N):

Polynomial coefficients for mode 1 where N = { (IDEG + 1)*(IDEG + 2) } /2

Items (8 + N) through (7 + N*NCOLS):

The coefficients are repeated for each mode; (NCOLS = NCOLN-MCOL 1 + 1)

Item (7 + N*NCOLS + 1):

10HPOLYNOMIAL

Generation: Subroutine POLY

SA Array for Surface Spline

File: SCRAND

Index Name: SA_i (where i may be from one to six)

Dimensions: M*1

where: M = maximum of {17 + 2*NNODES + (NNODES + 3) * (NTMODE + 2)} and { (NNODES + 3)**2 + (NNODES + 3)}

NNODES = Number of nodes (NIPTS, see below)

NTMODE = Number of modes (MCOLN, see below)

Elements:

Item 1: M, the number of items in this matrix

Item 2: 10HSURFSPLINE

Item 3: IPOINT = pointer to the transformation matrix
if IPOINT =0, no transformation matrix

Item 4: MCOLS, total number of modes

Item 5: MCOL1; modes 1 through MCOL1 will be zero on output modes

Item 6: MCOLN; (MCOLN-NCOL1 + 1) is the number of defined input modes, and modes MCOLN + 1 through MCOLS will be zero on output

Item 7: NIPTS, number of input points

Item 8: 14 + NSK, pointer to input points x, y coordinates (NPCOOR)

Item 9: 14 + NSK + 2*NIPTS + 2 + (NIPTS + 3), pointer to first spline coefficient (NPCOEF)

Item 10: XBAR, x cg location

Item 11: YBAR, y cg location

Item 12: COST, cosine of the rotation angle

Item 13: SINT, sine of the rotation angle

Item 14: RGU, Ru (radius of gyration)

Item 15: RGV, Rv (radius of gyration)

Item 16: INDS, Smoothing indicator
INDS = 0–no smoothing
= 1–applies to all input points

Items 17 through NPCOOR:
SK values if present

Items (NPCOOR) through (NPCOOR + 2*NIPTS):
U, V transformed representation of input points

Items (NPCOOR + 2*NIPTS + 1) through (NPCOEF-1):
Scratch area of 2*(NIPTS + 3)

Items (NPCOEF) through (NPCOEF + N):
Spline coefficients where N = (NIPTS + 3)*NCOLS-1

Items (NPCOEF + N + 1) through (NPCOEF + N + ITRAN):
Transformation matrix location (if specified)
where ITRAN = 12 if matrix exists
= 0 if matrix does not exist

Item (NPCOEF + N + ITRAN + 1):
10HSURFSPLINE

Generation: Subroutine SURF

Common Blocks

Table 11 displays the common blocks used in the program and the overlays in which they are used.

The labelled common blocks are used for communication between the secondary overlays, and between the routines in a secondary overlay. The block names and contents are described in table 12; each of the common blocks is described on a separate page.

Blank common is used in most secondary overlays as a variable length working storage area. The main program of the overlay calculates the area required for arrays in the various subroutines and passes a first-word address and variable dimension for each array through the subroutine's calling sequence. A description of the core used by each overlay is given in Figure 13.

Table 11.—Common Blocks Defined in Each Overlay

OVERLAY	COMMON BLOCKS										
	CHKPRT	DEFMOD	DEFSRF	INOUT	KWRECD	MATNAM	NERROR	OUTAPE	RWBUFF	TITLE	Blank
L215,0,0 L215											
L215,1,0 INTERP	x	x	x	x	x	x	x	x	x	x	
L215,1,1 RDEDIT	x		x	x	x	x	x	x	x	x	x
L215,1,2 BEAM			x	x		x	x	x			x
L215,1,3 MOTA			x	x		x	x	x			x
L215,1,4 MOTP			x	x		x	x	x			x
L215,1,5 POLY			x	x		x	x	x			x
L215,1,6 SURF			x	x		x	x	x			x
L215,1,7 RESULT	x		x	x		x	x	x	x	x	x
L215,1,10_8 INTEMD	x	x	x	x	x		x	x	x		x

Table 12.—Contents of Common Blocks

LABELED COMMON NAME: CHKPRT

DESCRIPTION: Print and flush options

NO.	VARIABLE	T	DIMENSION	ENG. NOM.	DESCRIPTION
1	ICKPRT	I			Print options =0, Results not printed =1, Print SA array =2, Print locations =3, Print SA array and locations =4, Print mode shapes =5, Print SA array and mode shapes =6, Print locations and mode shapes =7, Print SA array, locations, and mode shapes =999, checkout print for debugging purposes only.
2	IFLUSH	I			Indicator to call FLUSH(1) when fatal error occurs. =0, Do not call FLUSH(1) =1, Call FLUSH(1) when fatal errors detected.

NOTE: The following abbreviations are used to describe the variables.

NO.	Indicates the variable number in the common block list
VARIABLE	Common block item name
T	Type of variable I = Integer R = Real C = Complex H = Hollerith
DIMENSION	Number of elements in variable
ENG. NOM.	Engineering nomenclature—symbolic

Table 12.—(Continued)

LABELED COMMON NAME: DEFMOD

DESCRIPTION: Define output mode shape parameters

NO.	VARIABLE	T	DIMENSION	ENG. NOM.	DESCRIPTION
1	IDM	H			Mneumonic identification for the interpolated mode shape matrix.
2	IAEROT	I			Indicator for aerodynamic surface transformation: =0, no transformation =1, transformation values read.
3	AEROX	R		Xoa	X-coordinate of the origin of the aerodynamic surface in the reference axis system.
4	AEROY	R		Yoa	Y-coordinate of the origin of the aerodynamic surface in the reference axis system.
5	AEROZ	R		Zoa	Z-coordinate of the origin of the aerodynamic surface in the reference axis system.
6	OFFX	R		ΔX_{sh}	X-offset of the aerodynamic surface in the local axis system.
7	OFFY	R		ΔY_{sh}	Y-offset of the aerodynamic surface in the local axis system.
8	NOUTLO	I			Number of output point locations.
9	IAXISO	I			Indicator for axis frame output points: = 5HLOCAL, local axis =9HREFERENCE, reference axis

Table 12.–(Continued)

LABELED COMMON NAME: DEFSRF

DESCRIPTION: Define surface parameters

NO.	VARIABLE	T	DIMENSION	ENG. NOM.	DESCRIPTION
1	NTMODE	I			Total number of output modes.
2	ISURF	I			Surface number working on. Also, the file position number of the SATAP where the surface results are written.
3	ID	H			Nneumonic identification of the surface.
4	IMETHD	I			Method (scheme) of interpolation selected for the surface given below: =1, Beam spline =2, Motion axis =3, Motion point =4, Polynomial =5, Surface spline
5	ITRANS	I			Transformation of axis indicator =0, No transformation =1, Local to reference axis =2, Reference to local axis
6	XR	R		$X_{o\ell}$	X-coordinate of the origin of the local axis in the reference axis system.
7	YR	R		$Y_{o\ell}$	Y-coordinate of the origin of the local axis in the reference axis system.
8	ZR	R		$Z_{o\ell}$	Z-coordinate of the origin of the local axis in the reference axis system.
9	XRANG	R		θ_x	Rotation angle about x-axis from reference axis to local axis (deg.).

Table 12.–(Continued)

LABELED COMMON NAME: DEFSRF (continued)

DESCRIPTION: Define surface parameters

NO.	VARIABLE	T	DIMENSION	ENG. NOM.	DESCRIPTION
10	YRANG	R		θ_y	Rotation angle about y-axis from reference axis to local axis (deg.).
11	ZRANG	R		θ_z	Rotation angle about z-axis from reference axis to local axis (deg.).
12	ROTRAN	R	(3,4)	$[R]\begin{Bmatrix} X_{o\ell} \\ Y_{o\ell} \\ Z_{o\ell} \end{Bmatrix}$	Rotation and translation transformation matrix. Transformation is from reference to local. The last column is the translation array.
13	IAXISI	I			Indicator for the axis frame input nodes are in: =1, local axis =2, reference axis
14	NNODES	I			Number of nodes on the surface.
15	INUNIT	I			Indicator for the unit system node locations are in: =6HMETRIC, metric units =7HENGLISH, English units
16	SCALE	R	(6)		Scalar values for single freedoms. TX = TX * SCALE(1) TY = TY * SCALE(2) TZ = TZ * SCALE(3) RX = RX * SCALE(4) RY = RY * SCALE(5) RZ = RZ * SCALE(6)
17	IISURF	I			Parent surface number. Used when modes are generated from a previous surface.

Table 12.–(Continued)

LABELED COMMON NAME: DEFSRF (continued)

DESCRIPTION: Define surface parameters

NO.	VARIABLE	T	DIMENSION	ENG. NOM.	DESCRIPTION
18	NSAFR	I			Number of SA arrays to be generated for the surface.
19	ITX	I	(6)		Translation-X indicator =0, freedom not requested =1, freedom requested
20	ITY	I	(6)		Translation-Y indicator =0, freedom not requested =1, freedom requested
21	ITZ	I	(6)		Translation-Z indicator =0, freedom not requested =1, freedom requested
22	IRX	I	(6)		Rotation-X indicator =0, freedom not requested =1, freedom requested
23	IRY	I	(6)		Rotation-Y indicator =0, freedom not requested =1, freedom requested
24	IRZ	I	(6)		Rotation-Z indicator =0, freedom not requested =1, freedom requested

Table 12.—(Continued)

LABELED COMMON NAME: INOUT

DESCRIPTION: File names for card input, print, and punch output

NO.	VARIABLE	T	DIMENSION	ENG. NOM.	DESCRIPTION
1	INFIL	H			Data card input file name (6LTAPE5, 5LINPUT)
2	IUTFIL	H			Printed output file name (6LTAPE6, 6LOUTPUT)
3	IPFIL	H			Punched output file name (6LTAPE7, 5LPUNCH)

Table 12.—(Continued)

LABELED COMMON NAME: KWRECD

DESCRIPTION: Keyword record 9card image and code)

NO.	VARIABLE	T	DIMENSION	ENG. NOM.	DESCRIPTION
1	ICARD	H	(8)		Card image of last input data read.
2	ICODE	I			Keyword code number associated with keyword table.
3	IREAD	I			Indicator for reading next card: =0, read next card =1, do not read next card

Table 12.–(Continued)

LABELED COMMON NAME: MATNAM

DESCRIPTION: Name and size of matrices for scratch random file (SCRAND)

NO.	VARIABLE	T	DIMENSION	ENG. NOM.	DESCRIPTION
1	NAMBS	H			Name index for beam spline data.
2	LNBS	I			Number of words in beam spline data record.
3	NAMMA	H			Name index for motion axis data.
4	LNMA	I			Number of words in motion axis data record.
5	NAMPL	H			Name index for polynomia data.
6	LNPL	I			Number of words in polynomia data record.
7	NAMSS	H			Name index for surface spline data.
8	LNSS	I			Number of words in surface spline data record.
9	NAMXYL	H			Name index for local coordinates.
10	LNXYL	I			Number of words in local coordinates matrix.
11	NAMXYR	H			Name index for reference coordinates.
12	LNXYR	I			Number of words in reference coordinates matrix.
13	NAMTX	H			Name index for TX freedoms.
14	LNTX	I			Number of words in TX matrix.
15	NAMTY	H			Name index for TY freedoms.
16	LNTY	I			Number of words in TY matrix.

Table 12.—(Continued)

LABELED COMMON NAME: MATNAM (continued)

DESCRIPTION: Name and size of matrices for scratch random file (SCRAND)

NO.	VARIABLE	T	DIMENSION	ENG. NOM.	DESCRIPTION
17	NAMTZ	H			Name index for TZ freedoms.
18	LNTZ	I			Number of words in TZ matrix.
19	NAMRX	H			Name index for RX freedoms.
20	LNRX	I			Number of words in RX matrix.
21	NAMRY	H			Name index for RY freedoms.
22	LNRY	I			Number of words in RY matrix.
23	NAMRZ	H			Name index for RZ freedoms.
24	LNRZ	I			Number of words in RZ matrix.
25	NAMSA	H	(6)		Name index for SA arrays.
26	LNSA	I	(6)		Number of words in SA arrays.

Table 12.—(Continued)

LABELED COMMON NAME: NERROR

DESCRIPTION: Error parameters

NO.	VARIABLE	T	DIMENSION	ENG. NOM.	DESCRIPTION
1	IRR	I			Error number returned from subroutines called.
2	NWARN	I			Number of warning errors accumulated.
3	NFATAL	I			Number of fatal errors accumulated.

Table 12.—(Continued)

LABELED COMMON NAME: OUTAPE

DESCRIPTION: Magnetic file name for input, output, and scratch

NO.	VARIABLE	T	DIMENSION	ENG. NOM.	DESCRIPTION
1	ISATP	H			Tape name where SA arrays are written.
2	IMOTP	H			Tape name where interpolated mode shapes are written.
3	ISCTP	H			Tape name of scratch random file where matrices are stored temporarily during execution.

Table 12.–(Continued)

LABELED COMMON NAME: RWBUFF

DESCRIPTION: Buffer area required by READTP/WRTETP

NO.	VARIABLE	T	DIMENSION	ENG. NOM.	DESCRIPTION
1	IBUFF	I	(2)		IBUFF(1) = 8HBUFFSIZE IBUFF(2) = Buffer size = 10000
2	BUFF	R	(10000)		Buffer area for READTP/WRTETP routines.

Table 12.—(Concluded)

LABELED COMMON NAME: TITLE

DESCRIPTION: Title cards

NO.	VARIABLE	T	DIMENSION	ENG. NOM.	DESCRIPTION
1	NTITLE	I			Number of title cards.
2	ITITLE	H	(8,4)		Title cards to be printed at top of output listing.

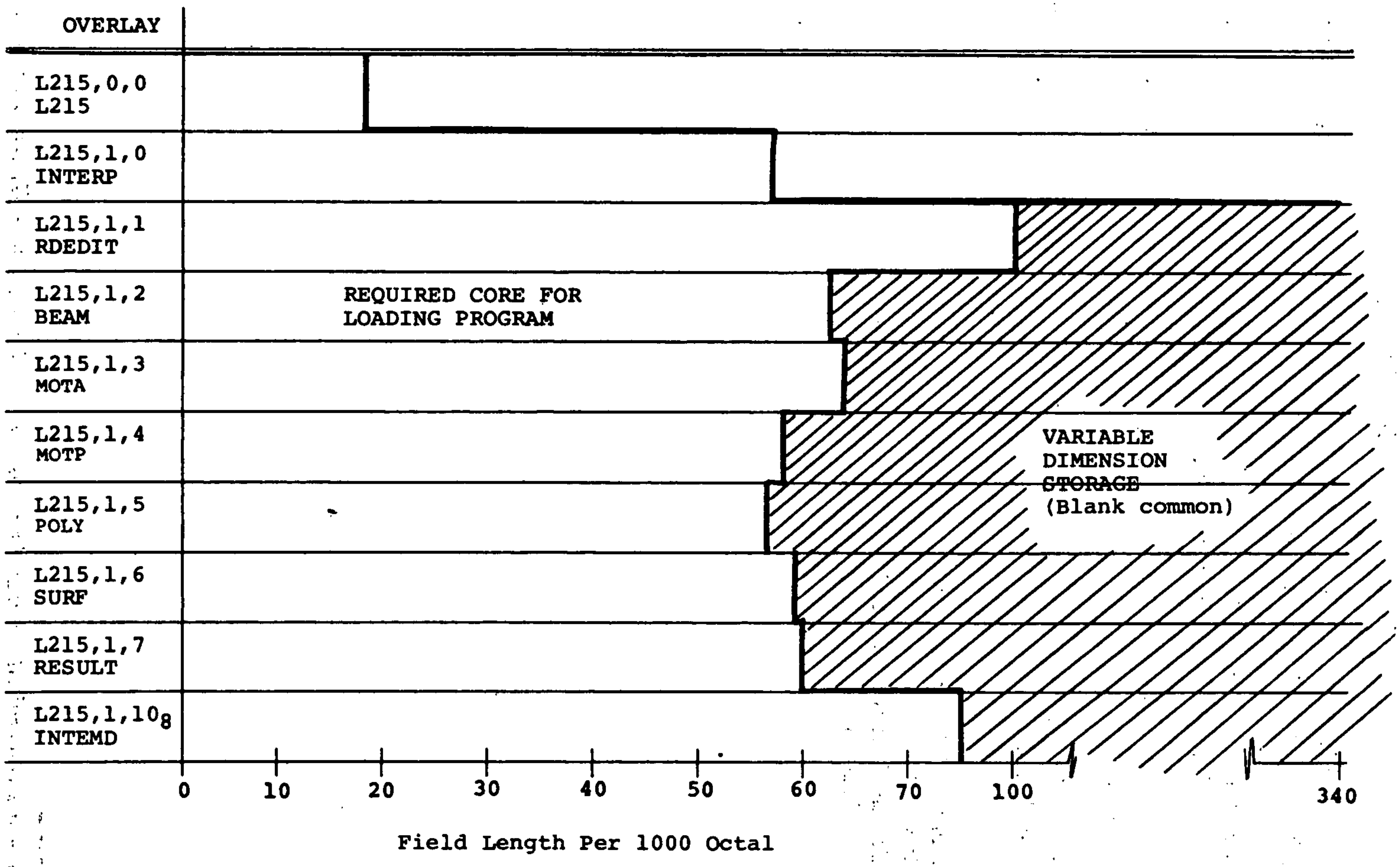


Figure 13.—Core Used by Each Overlay

4.0 EXTENT OF CHECKOUT

Four sample problems were developed as test cases to verify the correctness of L215 (INTERP). They are described in table 13.

Case one uses the basic ATLAS 3.1[1] test case for the Interpolation module. The results from L215 and ATLAS 3.1 Interpolation are identical.

Case two checks error diagnostics of the L215 program. Case one card input data was modified to construct this case.

Cases three and four were developed by the engineer, and results were checked against hand-calculated results.

Boeing Commercial Airplane Company
P.O. Box 3707
Seattle, Washington 98124
May 1977

[1] *ATLAS – An Integral Structural Analysis and Design System – System Document.* Boeing Document, 1974 D6-25400-0002TN.

Table 13.—Verification Test Cases

OPTIONS	CASE 1	CASE 2	CASE 3	CASE 4
1. Beam Spline	x	x		
2. Motion Axis	x	x	x	x
3. Motion Point	x	x		
4. Polynomial	x	x		
5. Surface Spline	x	x		
6. Local to Reference	x	x		
7. Reference to Local			x	x
8. Nodes from CARD	x	x	x	x
9. Nodes from TAPE			x	
10. SORT nodes			x	
11. PARENT Modes			x	
12. Combined freedoms from CARD				x
13. Combined freedoms from TAPE			x	
14. SORT combined freedoms			x	x
15. Single freedoms from CARD	x	x		
16 Single freedoms from TAPE				x
17. SORT single freedoms				x
18. RIGID Modes			x	
19. SA one			x	
20. SA two			x	
21. Aero Transformation			x	x
22. Out locations from CARD	x	x	x	
23. Out locations from TAPE				x
24. One surface	x	x	x	x
25. Control surface			x	
26. Tab-tab			x	
27. Diagnostic Print		x		

REFERENCES

1. Miller, R. D.; Kroll, R. I.; and Clemmons, R. E.: Dynamic Loads Analysis System (DYLOFLEX) Summary. NASA CR-2846-1, 1979.

2. Miller, R. D.; Richard, M.; and Rogers, J. T.: Feasibility of Implementing Unsteady Aerodynamics Into the FLEXSTAB Computer Program System. NASA CR-132530, 1974.

1. Report No.: NASA CR-2848

2. Government Accession No.:

3. Recipient's Catalog No.:

4. Title and Subtitle: MODAL INTERPOLATION PROGRAM, L215 (INTERP) Volume II: Supplemental System Design and Maintenance Document

5. Report Date: September 1979

6. Performing Organization Code:

7. Author(s): M. Y. Hirayama, R. I. Kroll, and R. E. Clemmons

8. Performing Organization Report No.: D6-44457

9. Performing Organization Name and Address:
Boeing Commercial Airplane Company
P.O. Box 3707
Seattle, Washington 98124

10. Work Unit No.:

11. Contract or Grant No.: NAS1-13918

12. Sponsoring Agency Name and Address:
National Aeronautics and Space Administration
Washington, DC 20546

13. Type of Report and Period Covered: May 1975 to May 1977

14. Sponsoring Agency Code:

15. Supplementary Notes:
Langley Technical Monitors: Robert C. Goetz and Boyd Perry III
Topical Report

16. Abstract

This document describes the usage of the Modal Interpolation program L215. (INTERP). The program uses modal data to form sets of arrays containing interpolation coefficients. The interpolation arrays then can be used to determine displacements at various aerodynamic control points. The displacements consist of translations normal to the aerodynamic surface and surface slopes that are parallel and perpendicular to the free-stream direction. Five different interpolation methods are available. A description of the data manipulation and the interpolation methods is presented in volume I of this document. Volume II contains a description of the design and structure of the program to aid those who will maintain and/or modify the program in the future.

17. Key Words (Suggested by Author(s)):
Modal Interpolation
Beam Spline
Surface Spline
Polynomial
Motion Axis
Motion Point

18. Distribution Statement:
Unclassified – Unlimited

Subject Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	66	$5.25

*For sale by the National Technical Information Service, Springfield, Virginia 22151