COPY NO. 8

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle CONSOLIDATION OF DATA BASE FOR ARMY GENERALIZED MISSILE MODEL		5. Report Date August 1980 6. Performing Organization Code 540/C
7. Author(s) Daniel J. Klenke and Michael J. Hemsch		8. Performing Organization Report No. NEAR TR-221 10. Work Unit No.
9. Performing Organization Name and Address Nielsen Engineering & Research, Inc. 510 Clyde Avenue Mountain View, CA 94043		11. Contract or Grant No. NAS2-10081 13. Type of Report and Period Covered Final Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Ames Research Center Moffett Field, CA 94035		14. Sponsoring Agency Code
15. Supplementary Notes Point of Contact: Gerald Malcolm, M/S 227-8, NASA Ames Research Center Moffett Field, CA. 95035 FTS 448-965-6266, 415-965-6266		

16. Abstract

A computer program has been written to consolidate the results from a series of tests on the Army Generalized Missile model and make them readily accessible for plotting, listing, and evaluation. The program is written in FORTRAN and will run on an ordinary minicomputer. It has the capability of retrieving any coefficient from the existing DATAMAN tapes and displaying it in tabular or plotted form. Comparisons of data taken in several wind tunnels and of data with the predictions of Program MISSILE2 are also presented.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Cruciform missiles Data base Aerodynamic characteristics Computer Program	Unlimited STAR Category 01

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	98	

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

CONSOLIDATION OF DATA BASE FOR ARMY GENERALIZED MISSILE MODEL

by

Daniel J. Klenke
and
Michael J. Hemsch

NEAR TR-221

August 1980

FINAL REPORT

Prepared under Contract NAS2-10081

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Ames Research Center
Moffett Field, CA

by

NIELSEN ENGINEERING & RESEARCH, INC.
510 Clyde Avenue, Mountain View, CA 94043
Telephone (415) 968-9457

TABLE OF CONTENTS

CONSOLIDATION OF DATA BASE FOR ARMY GENERALIZED MISSILE MODEL

SUMMARY

A computer program has been written to consolidate the results from a series of tests on the Army Generalized Missile model and make them readily accessible for plotting, listing, and evaluation. The program is written in FORTRAN and will run on an ordinary minicomputer. It has the capability of retrieving any coefficient from the existing DATAMAN tapes and displaying it in tabular or plotted form. Comparisons of data taken in several wind tunnels and of data with the predictions of Program MISSILE2 are also presented.

INTRODUCTION

Over the past several years a generalized missile model, designed and built for the Army Missile Command (MICOM), has been tested in several wind tunnels under a wide range of conditions. Some of the results of these tests have been recorded using Chrysler Corporation's DATAMAN system which produces formatted magnetic tapes.

The purpose of the work reported here was to develop a computer program which would consolidate the results from these tests and several others (references 1-8) and make them readily accessible for plotting, listing, and evaluation. The program is written in FORTRAN and will run on an ordinary minicomputer. It has the capability of retrieving any coefficient from the existing DATAMAN tapes and displaying it in tabular or plotted form.

The next section presents the considerations involved in designing the data consolidation program. Following that is a description of the program structure and its method of operation. Next a user's guide to the program is given which is followed by a set of comparisons of data taken in different wind tunnels and comparisons of data with the missile aerodynamics prediction program, MISSILE2 (reference 9). The report is concluded with an appendix which contains a listing of the program, a description of each of the subroutines, a dictionary of common block variables and the program flow charts.

PROGRAM DESIGN CONSIDERATION

The purpose of this project is to create a computer program which will read a certain set of magnetic tapes and, according to instructions from the user, produce plots and tabulated listings of data from those tapes. The magnetic tapes referred to here are the Chrysler DATAMAN tapes listed in Table 1. The program is adaptable to a typical minicomputer system and uses existing plotting routines and hardware.

One of the major factors in the design of the program, DATAMANIA, is the very large amount of data stored on magnetic tape. The main problem encountered in retrieving information from the DATAMAN tapes is that magnetic tapes are sequential access devices. This means that in order to read the second record on a magnetic tape, one must first read the first record. And, in order to read the last record on a tape, all preceding records must be read. If all the data on the tape is to be processed, this restriction is unimportant. However, if the user is searching for a particular piece of information, such as a dataset name, the problem becomes magnified. As each record is read a comparison of some sort must be made to determine whether the current record is the one being sought. Even at the speed of modern computers this process can be quite slow. To overcome this problem, DATAMANIA transfers selected information from magnetic tape to a random access disk storage device.

A second major factor in the design of DATAMANIA is that it must be adaptable to a typical minicomputer system. The system which has been used for the present work is described below and those features which are clearly typical of minicomputers are enumerated.

The minicomputer facility, located at NASA/Ames Research Center, is organized around a Digital Equipment Corporation (DEC) PDP-11/70 with 192K, 16-bit words of core memory. The PDP-11/70 runs under DEC's multiuser, multitasking operating system RSX-11M. The peripheral equipment includes a Versatec electrostatic printer/plotter, a nine-track tape drive, and two 1.2-million word disk drives. Access to the computer is by telephone using the resource code FAC11. Resident on FAC11 is software which allows access to the Research Center's CDC 7600, a large scale computer. This software and the associated communication interfaces which allow the PDP-11/70 to act as a remote job entry station for the 7600, are not typical of minicomputer systems. However, a typical minicomputer system will include a central processing unit with a limited amount of core storage, a tape drive, a disk storage unit, a FORTRAN compiler, and some type of output device capable of drawing plots.

To ensure that DATAMANIA is easily adaptable to other computer systems the program has been constructed in a modular fashion. Each subroutine performs one function and each common block includes one class of data.

A final requirement for the finished program is that it should be easy to use. This indicates an interactive mode of operation in which the user controls the direction of the program as it is running. This allows for a certain amount of trial-and-error by the user along with the benefit of nearly immediate results.

GENERAL STRUCTURE AND METHOD OF OPERATION

The program DATAMANIA does one task repeatedly. The task is to produce a frame of data which can be printed and/or plotted. Each frame consists of one or more sets of independent and dependent variable data and pertinent descriptive information. The user, working from a terminal, directs the production of each frame by responding to prompts from the program.

To construct a frame, the program completes the following steps. First, a DATAMAN tape is searched for a particular data set. Next, a random access file is created along with an index to that file. Finally, the information pertinent to the current frame is extracted from the random access file and stored in plotting arrays. If more than one dataset is required to complete the current frame, this process is repeated. Once a frame is complete it can be listed at the terminal or a plot file can be created for later use.

DATAMAN Tapes

The Chrysler DATAMAN tapes were written on a UNIVAC 1108/3G computer. Due to differences between equipment it is likely that the tapes will not be compatible with a typical minicomputer. Therefore, some preprocessing of the tapes will be necessary.

Each DATAMAN tape is partitioned into datasets. A DATAMAN dataset represents one or more wind tunnel runs. Each dataset includes a unique six character identifier, a description of the model configuration, and a list of parametric values which are constant for all runs in the dataset. The data recorded in the wind tunnel are presented as a function of two independent variables. The tests of Burt (Refs. 1,3,4), Hemsch (Ref. 8), and Schwind (Ref. 7) use Mach number and angle of attack as independent variables while Henderson's tests (Refs. 2,5,6) use either Mach number and jet coefficient or angle of attack and jet coefficient as the two independent variables. A description of the FORTRAN statements used to write a DATAMAN tape are given in Table 2 along with a dictionary of the variables used.

Random Access Data Base

In order to make efficient use of the information stored in each dataset and to minimize the time involved in searching a magnetic tape, a random access disk file is created each time a particular dataset is used. At the same time, the name of the dataset is added to a list of existing disk file datasets. In this way it is not necessary to perform another tape search when the dataset is used again.

The random access file created by DATAMANIA is a reorganized and abbreviated DATAMAN dataset. The term 'random access' implies that each record of the file can be found in about the same amount of time. The system accomplishes this by numbering each record as it is written. In order to retrieve a particular record it is necessary to know its record number. DATAMANIA makes use of this by saving the record numbers of key records in an index array.

The structure of the random access file is pictured in figure 1 and an example of a random access dataset is shown in figure 2. The first segment of the file consists of one record which contains identifiers for the dataset, a list of parameters, and a series of values which describe the size of the dataset. In particular, the variables NMN, MAXNAA, and NDV appear in this record. They represent respectively, the number of values of first independent variable tested, the maximum number of second independent variable values per first independent variable, and the number of dependent variables recorded. The record number of this record serves as the first index to this dataset.

The second segment of the file represents the values of first independent variable encountered during the test and consists of NMN + 1 records. The first record, whose record number is the second index, contains the name of the first independent variable and its minimum and maximum values. This is followed by NMN records each of which contains MAXNAA values of first independent variable. When the program refers to this dataset, all of these values are read into an array whose dimensions are NMN by MAXNAA.

Segment three, which contains second independent variable data, is identical in form and size to segment two. The first record of this segment contains the name of the variable and its minimum and maximum values. The third index for this dataset contains the record number of this record.

Segment four consists of only one record which contains the Reynolds numbers and run numbers for this dataset.

The fifth segment contains all of the dependent variable data. It is made up of NDV blocks of records, each of which are in the same form as the independent variable data. Each block begins with a dependent variable name and its minimum and maximum values. These record numbers are used as indices to the dataset and are numbered five through NDV + 4.

The dataset shown in figure 2 is taken from Burt's Canard Control test in the Ames 6- by 6-Foot Wind Tunnel (reference 3) and is named REZ048. By comparing figures 1 and 2 it may be seen that the two independent variables for REZ048 are MACH and ALPHA. The four numbers appearing at the end of the first line in figure 2 are the values of the variables NMN, MAXNAA, NDV, and NPARM. NMN equal to two means that two Mach numbers were tested while a value of seven for MAXNAA means that a maximum of seven values of angle of attack were tested for each Mach number. NDV equal to nine implies that nine dependent variables were recorded for this dataset. Their names, CN, CA, CM, CY, CYM, CRM, CRMB, CYB, and CNC, appear in figure 2 preceding the corresponding arrays of dependent variable data. A value of nine for the variable NPARM indicates that there are nine parameters for this dataset. Their names and values appear in the second and third lines respectively of figure 2.

The indexing scheme referred to above is implemented by storing the appropriate record numbers in an array named INDEX. Each dataset can have up to fourteen indices, depending on the number of dependent variables. The array INDEX has dimension 100 by 14 and can therefore serve as an index for up to 100 datasets. The first subscript refers to a particular dataset and the second to a particular record of that dataset. The datasets are numbered sequentially as they are created. Thus, the number stored in INDEX (3,1) is the record number of the first record of the third dataset created. In order to read the name of the first dependent variable of that dataset, one must reference the record whose record number is stored in INDEX (3,5). A separate array, named DSLIST, is used to store the names of the datasets indexed by the array INDEX.

Method of Operation

Once a random access file has been created for a particular dataset, the program can read the file and transfer information pertinent to the current frame to in-core storage areas. The data used by the program include all of the dataset descriptors and three arrays of numerical data. The three arrays, named A, B, and C, contain first independent variable, second independent variable, and dependent variable data respectively. At any one time, this is the only information available to the program. Once this information is in-core it is processed according to user requests. To see how this is done, consider the example illustrated by the following figure which shows what DATAMANIA knows about the hypothetical dataset AAA001. For this dataset, the first independent variable is Mach number, the second is angle of attack, and the dependent variable is CN.

Dataset name - AAA001

Configuration - BODY + CANARDS

Dependent Variable - CN

NMN = 3

MAXNAA = 8

Parameters

PHI = 0.0

PT = 7.0

RE/FT = 2.1

A — Mach number

0.79	0.80	0.81	0.80	0.80	0.79	0.81	0.80
1.20	1.22	1.21	1.21	1.20	1.19	1.20	1.21
1.29	1.28	1.28	1.30	1.30	1.29	1.28	1.28

B — Alpha

20.1	26.1	30.1	36.1	39.8	42.7	45.1	49.8
20.2	25.8	28.7	35.9	40.1	43.1	45.0	50.1
20.1	25.5	29.9	36.0	40.8	43.4	44.8	50.0

C — CN

5.1	6.3	7.4	8.6	9.3	10.1	11.3	12.6
6.2	7.5	8.1	9.2	10.2	11.4	12.7	13.1
7.9	8.3	9.4	10.5	11.7	12.9	13.6	14.8

If a plot of CN versus ALPHA is requested, the program will ask the user for a Mach number, or series of Mach numbers to be plotted. Suppose the user responds with Mach number = 0.8. DATAMANIA proceeds by searching the first column of A for a number equal to 0.8 ± 0.04. The first element satisfies this criterion so the program transfers the first row of B and the first row of C to plotting arrays X and Y respectively. Notice that DATAMANIA has assumed that every element of the first row of A is near 0.8.

To produce a plot of CN as a function of Mach number, the user must input a value of angle of attack as a parameter. Suppose a value of 30.0 is input. In this case the goal of the program is to put Mach number values in array X and corresponding values of CN in array Y. To achieve this end, the program begins by scanning the first row of B for values of angle of attack that bracket 30.0; in this case, the second and third elements of row one. Using these numbers and the corresponding elements from the array C, a linear interpolation is performed to find the value of CN for an angle of attack of 30.0. This value becomes the first element of the Y array. The corresponding Mach number is taken to be the average between the second and third numbers of row one in the Mach number array. Again, it is assumed that the Mach numbers will not vary significantly. DATAMANIA then repeats this process for each remaining row.

Since each dataset represents a test performed at one roll angle, PHI, it is necessary to use a series of datasets to produce a plot with roll angle as the independent variable. DATAMANIA depends on the user's good judgment in selecting these datasets. To produce a meaningful plot, the parameters from each dataset should be identical (excepting PHI, of course), as should the configurations. DATAMANIA processes these datasets one at a time, performing the same task each time. First, the parameter list is searched for the word PHI. The corresponding parametric value is placed in the X array. To find the value of CN which is associated with this roll angle, bilinear interpolation is used. The user will have specified a Mach number and an angle of attack for which a CN value is desired. The Mach number array and the angle of attack array are searched for bracketing values from which CN is to be determined. Then, CN is determined by first interpolating in Mach number and then in angle of attack. The resulting CN is stored in the Y array for later plotting or listing.

Plotting

An important aspect of the design of DATAMANIA is its adaptability to a variety of plotting packages. This was accomplished by passing all plot information in two common blocks; one containing the arrays to be plotted and the other containing textual data which is to appear on the plot.

For the present implementation of DATAMANIA the PLOTA/PLOTI (reference 10) routines were chosen. This package, developed at Nielsen Engineering & Research, Inc. (NEAR), was selected for several reasons. First, it is simple to use; a single FORTRAN CALL statement will generate an X,Y plot. Second, automatic scaling or user defined scales are available. Third, the code is readily available and the necessary host routines are already resident on the computer system which was used. Finally, the entire package is clearly and thoroughly documented.

Output from the PLOTA/PLOTI routines consists of plot information in the Simple Graphics Package (SGP) File format (reference 11). This intermediate file, which can be read and edited, can then be disposed to a device-specific post processor to create CRT displays or hard copy plots. Post processors have been written (reference 11) for the following devices:

1) Evans and Sutherland Picture System
2) Versatec Electrostatic Printer/Plotter
3) IMLAC - PDS1 Graphic Display Scope
4) Zeta Pen Plotter
5) SC4020 Hard Copy Device

The plots appearing in this report were originally produced on a Versatec printer/plotter.

Implementation on DEC PDP-11/70

The implementation of DATAMANIA on the DEC PDP-11/70 described above is governed by several factors. First, NEAR is physically separated from the computer by more than a mile. The system does not have a full-time operator to respond to requests for tape mounts or other auxiliary services. Also, the system supports many users with the result that public disk storage space is limited. Finally, the DATAMAN tapes created by Chrysler Corporation cannot be read by DEC machinery. Because of these limitations, it was decided to make use of the communicating link between the minicomputer and the large scale CDC 7600.

The first necessary step for running DATAMANIA on the PDP-11/70 is to translate the DATAMAN data to a format compatible with the DEC system. This is accomplished by reading the tapes on the 7600 according to the FORMAT statements in Table 2 and then writing the data to a 7600 disk file. In this process the arrays NAMDM, DM, and GD, which are not used on any of the DATAMAN tapes, are deleted from each dataset. The disk file created in this way can be sent to the PDP-11/70 using the communicating link between the two computers. Once this file is captured on the DEC system it is given the name FOR001.DAT. By DEC convention, this name is associated with logical unit 1 for FORTRAN programs; i.e., when DATAMANIA reads data from logical unit 1, it reads the file named FOR001.DAT.

The fundamental executable unit on the DEC minicomputer is a task. To make DATAMANIA a task, the source code, as shown in the appendix, is entered in the system as a text file by using one of the system's editors. Next, the text file is translated into an object module by using the FORTRAN compiler. The object module is then converted to a task image using DEC's Task Builder program. The program DATAMANIA is then ready to run.

The only nonstandard FORTRAN language features used in DATAMANIA concern the creation and use of a random access file. A random access file is created using the statement OPEN with appropriate parameters. Reading the file requires the use of a READ statement whose syntax is nonstandard. Implementation of this program on another type of machine will involve replacing each occurrence of these two statements with their machine dependent counterparts.

The PLOTA/PLOTI plot package used by DATAMANIA writes its output to logical unit seven. On the PDP-11/70 this means that a disk file named FOR007.DAT is created. A standard plot created by the PLOTA/PLOTI package has a bottom axis and a left side axis only. The plotting routines themselves have been left in standard form except for the addition of two subroutines, RITAX and TOPAX, which draw the top axis and right side axis. At the end of a DATAMANIA session, plots are generated by disposing file FOR007.DAT to the appropriate SGP post-processor.

USER'S GUIDE

Success in the use of DATAMANIA depends to a large degree on the preparation of the user. Since the program is intended to be a model for similar implementations on other computer systems, it was considered important to keep the code simple. As a result, many error detecting and processing devices have been omitted which would be valuable in a version oriented towards production.

To prepare for a DATAMANIA session the user should make a list of the plots to be generated and a list of the datsets necessary to make those plots. This information can be compiled from the wind tunnel reports, references 2 through 8. Once this is done, and the necessary DATAMAN tapes are ready for mounting, the user is ready to use DATAMANIA.

DATAMANIA builds frames of data by asking the user for descriptive information. The first prompt to the user is for independent variable name. He responds by typing either ALPHA, MACH, CRT, or PHI. This response determines the program's mode of operation until the current frame is complete.

If either ALPHA, MACH, or CRT are named as the independent variable, DATAMANIA will ask for the name of the dataset to be used and the name of the dependent variable being plotted. Next, the user is asked to input a series of parametric values. The parameter used will depend on the independent variable and the dataset being used. One curve will be produced for each parameter. When this data has been assembled the user has the option of putting more curves on the same frame using either the same dataset or another dataset. A maximum of five curves per frame is allowed.

When a frame is to be generated using PHI as the independent variable, the user must input a list of dataset names to be used. It is the responsibility of the user to ensure that these datasets are compatible with one another and that each represents a test performed at a different roll angle. After the dataset list is input, the program prompts for a dependent variable name. Now, the user has the option of producing either a series of curves each representing a different Mach number and one angle of attack or representing different alphas and one Mach number. DATAMANIA determines this by printing "ENTER NAME OF PARAMETER TO BE HELD CONSTANT (MACH OR ALPHA) - ".

As each frame is completed DATAMANIA asks the user if a tabulation is desired. If the response is "YES", the contents of the current frame are listed at the terminal. Next, the user is asked if a plot file should be created. If so, he is prompted for the type of scaling to be used. The scaling options are default scaling, same-as-last-plot scaling, and user defined scaling. The default scaling option will produce plots which include all data points within the smallest range which will allow reasonable labeling of the tick marks. To generate plots with user defined scales the user must input minimum and maximum values for both the independent variable and the dependent variable.

Sample Session

The following sample session with DATAMANIA will illustrate its use in generating three frames of data each using a different independent variable. Underlining has been used to show user responses. All three frames are tabulated as well as plotted.

```
>RUN DATAMANIA

           ********** WELCOME TO DATAMANIA **********


DO YOU WANT INSTRUCTIONS?  YES


    DATAMANIA IS DESIGNED TO READ DATAMAN TAPES AND
PRODUCE PLOTS OF COEFFICIENT DATA AS A FUNCTION OF
EITHER ANGLE OF ATTACK (ALPHA), MACH NUMBER, THRUST
```

COEFFICIENT (CRT), OR ROLL ANGLE (PHI). IN ORDER TO DO THIS, YOU NEED TO KNOW THE NAMES OF THE DATASETS WHICH CONTAIN THE INFORMATION TO BE PLOTTED.

AT THE BEGINNING OF EACH FRAME, DATAMANIA WILL PROMPT FOR THE INDEPENDENT VARIABLE NAME. YOUR RESPONSE OF ALPHA, MACH, CRT, OR PHI WILL DETERMINE THE MODE OF PROGRAM OPERATION. FOLLOWING THIS YOU WILL BE PROMPTED FOR INFORMATION NEEDED TO COMPLETE THE FRAME.

AS EACH FRAME IS COMPLETED YOU WILL BE ASKED FIRST IF YOU WANT A TABULATION OF THE CURRENT FRAME, AND THEN, IF YOU WANT TO CREATE A PLOT FILE OF THE FRAME.

ENTER INDEPENDENT VARIABLE NAME - <u>ALPHA</u>

ENTER DATASET IDENTIFIER - <u>RAW002</u>

DO YOU WANT A SUMMARY OF RAW002? <u>YES</u>

SUMMARY OF RAW002

```
**************************************************
*                                 CREATED 22 SEP 77
*     CONFIGURATION - BODY + TAILS
*
*     PARAMETERS
*     ----------
*    RE/FT = 2.100
*    PHI   = 0.000
*    PT    = 7.000
*        MACH NUMBERS
         ------------
*   INDEX        RANGE
*     1    FROM 0.790 TO 0.790
*     2    FROM 1.210 TO 1.220
*     3    FROM 1.280 TO 1.290
*
*      DEPENDENT VARIABLES
*      -------------------
*             CN
*             CY
*             CA
*             CLM
*             CYN
*             CBL
*             PHI-M
*             PHI-C
*             PHI-T
*             BETA
*
**************************************************
```

```
ENTER DEPENDENT VARIABLE NAME - CN

ENTER NUMBER OF MACH NUMBERS TO BE PLOTTED - 2

ENTER 2 MACH NUMBERS (SEPARATE WITH COMMAS) - .8,1.2

ANY MORE DATA ON THIS FRAME? NO

TABULATE DATA? YES



$$$$$$$$$$$$$ TABULATION OF CURRENT FRAME
$
$
$
$   DATASET - RAW002
$   CONFIGURATION - BODY + TAILS
$               PARAMETRIC VALUES
$               -----------------
$               RE/FT       2.100
$               PHI         0.000
$               PT          7.000
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
$
$
$      ALPHA        CN       FOR MACH = 0.8 FROM RAW002
$     20.0700     4.3457
$     21.9700     4.7439
$     23.8400     5.2159
$     26.8300     6.0867
$     29.7400     6.8728
$     32.7600     7.6546
$     34.6600     8.1298
$     38.5600     9.2253
$     41.6400    10.0630
$     44.7200    11.0890
$     47.5600    11.9860
$     49.2400    12.4710

$
$
$      ALPHA        CN       FOR MACH = 1.2 FROM RAW002
$     20.1300     5.2252
$     22.0600     5.8659
$     23.9500     6.4432
$     26.9800     7.5335
$     29.9400     8.6510
$     33.0000     9.9209
$     34.9300    10.7130
$     38.8800    12.2730
$     41.9800    13.3800
$     45.0600    14.3030
$     47.8900    14.9450
$     49.5700    15.2940
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
CREATE PLOT FILE? YES

USE DEFAULT SCALING? YES

WANT TO MAKE ANOTHER FRAME? YES

ENTER INDEPENDENT VARIABLE NAME - MACH

ENTER DATASET IDENTIFIER - RAW002

DO YOU WANT A SUMMARY OF RAW002? NO

ENTER DEPENDENT VARIABLE NAME - CN

ENTER 3 ALPHAS (SEPARATE WITH COMMAS) - 20.,30.,50.

ANY MORE DATA ON THIS FRAME? NO

TABULATE DATA? YES


$$$$$$$$$$$$$ TABULATION OF CURRENT FRAME
$
$
$
$   DATA SET - RAW002
$   CONFIGURATION - BODY + TAILS
$               PARAMETRIC VALUES
$               -----------------
$               RE/FT       2.100
$               PHI         0.000
$               PT          7.000
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

```
$
$
$      MACH          CN        FOR ALPHA=20.0 FROM RAW002
$     0.7900       4.3310
$     1.2150       5.1820
$     1.2900       5.1057

$
$
$      MACH          CN        FOR ALPHA=30.0 FROM RAW002
$     0.7900       6.9
$     1.2150       8.6759
$     1.2900       8.6621

$
$
$      MACH          CN        FOR ALPHA=50.0 FROM RAW002
$     0.7900      12.6904
$     1.2100      15.3833
$     1.2850      15.5535
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

CREATE PLOT FILE? YES

USE DEFAULT SCALING? YES

WANT TO MAKE ANOTHER FRAME? YES

ENTER INDEPENDENT VARIABLE NAME - PHI

ENTER NUMBER OF PHI VALUES TO BE PLOTTED - 4
```

ENTER 4 DATASET NAMES

EACH FOLLOWED BY CARRIAGE RETURN

RAW002

RAW003

RAW004

RAW005

ENTER DEPENDENT VARIABLE NAME - CN

ENTER PARAMETER TO BE HELD CONSTANT (MACH OR ALPHA) - MACH

ENTER VALUE OF CONSTANT PARAMETER - .8

ENTER NUMBER OF ALPHA VALUES TO BE PLOTTED - 3

ENTER 3 ALPHA'S (SEPARATE WITH COMMAS) - 20.,30.,50.

DO YOU WANT A SUMMARY OF RAW002? NO

DO YOU WANT A SUMMARY OF RAW003? NO

DO YOU WANT A SUMMARY OF RAW004? NO

DO YOU WANT A SUMMARY OF RAW005? NO

TABULATE DATA? YES

```
$$$$$$$$$$$$ TABULATION OF CURRENT FRAME
$
$ USING DATASETS
$                 RAW002
$                 RAW003
$                 RAW004
$                 RAW005
$
$ CONFIGURATION - BODY + TAILS
$
$                  PARAMETRIC VALUES
$                  -----------------
$                  RE/FT       2.100
$                  PT          0.000
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

                   MACH = 0.8

$
$
$       PHI          CN       FOR ALPHA=20.0
$      0.0000      4.3513
$     10.0000      4.3836
$     30.0000      4.1957
$     45.0000      4.2816
$
$
$       PHI          CN       FOR ALPHA=30.0
$      0.0000      6.9814
$     10.0000      6.8639
$     30.0000      6.7752
$     45.0000      6.5203
$
```

```
$
$        PHI          CN       FOR ALPHA=50.0
$       0.0000      12.7545
$      10.0000      -5.6740
$      30.0000      11.8958
$      45.0000      11.5471
$
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

CREATE PLOT FILE? YES

USE DEFAULT SCALING? YES

WANT TO MAKE ANOTHER FRAME? NO

*********** END OF SESSION ***********
```

DATA COMPARISONS

The consolidated data base for the Army Generalized Missile can be divided into three categories:

1. Henderson's plume interaction tests, references 2,5,6.
2. Burt's tests of nose mounted canard configurations, references 1,3,4.
3. NEAR high angle-of-attack tests, references 7,8.

Comparisons of data taken in different wind tunnels can be made for the first and third categories and are presented below. Also, comparisons of data and the predictions of Program MISSILE2 (reference 9) are presented below for some of the tests in categories 2 and 3.

Comparisons of Data Taken in Different Wind Tunnels for Plume Interaction Tests

A comprehensive series of plume interaction tests were run in the Calspan 8-Foot Transonic Wind Tunnel and the AEDC 16T and 16S Wind Tunnels for several body-alone and body-tail configurations. Some of the test parameters at the two facilities were identical and provide a means of checking the sensitivity of the results to test conditions. Figure 3 provides geometrical details of the Army Generalized Missile model. Comparisons are presented for the following configurations.

(a) Body-alone with Army nose for the following conditions:

Mach	Radial Thrust Coefficient, CRT	Figure
0.4	0, 12	4a
1.0	0, 6	4b
1.25	6	4c

(b) Body with Army nose and Army F1 tail fins mounted in the forward hinge-line position for the following conditions:

Mach	Radial Thrust Coefficient, CRT	Figure
0.4	0	5a
1.0	0, 12	5b
1.25	6	5c

(c) Body with Army nose and Army F2 tail fins mounted in the forward hinge-line position for the following conditions:

Mach	Radial Thrust Coefficient, CRT	Figure
1.0	0, 6	6

The pitching-moment coefficient is shown for comparison since it is usually a more sensitive variable than normal force. In general, except for zero shifts at $M_\infty = 0.4$ for the body alone and $M_\infty = 1.25$ for the Fin 1 case, the comparisons for all three configurations are good.

Comparisons of Data From Nose-Mounted Canard-Configuration Tests with Predictions of Program MISSILE2

In this section, comparisons of data with the prediction of Program MISSILE2 (reference 9) are presented for a particular canard configuration with inline and with interdigitated tail fins. The comparisons are for three Mach numbers ($M_\infty = 0.9$, 1.25, 2.00) and for roll angles of 0 and 45 degrees. The configuration consists of the Army Generalized Missile body with the Army nose, the Army canards mounted at the aft canard hinge-line position, and the Army F1 tail fins mounted at the aft tail hinge-line position.

Results for overall normal-force coefficient with zero roll angle and no fin deflection are presented in figure 7. While the Mach number and angle of attack trends are correct, the theory underpredicts the data by about 10-15%. The rest of the comparisons for this configuration are for the pitching-moment coefficient which is a sensitive indicator of the ability of the theory to predict longitudinal stability. The results are presented in figures 8, 9 and 10 for $M_\infty = 0.9$, 1.25 and 2.0 respectively. (Note the difference in moment center for the $M_\infty = 2.0$ case.) Except for the 15° canard fin deflection case, the agreement is good for $M_\infty = 0.9$ and 1.25. The worst agreement (excluding the 15° fin deflection cases) gives an error of roughly 0.4 body diameters in the position of the longitudinal center of pressure. For $M_\infty = 2.0$, the maximum error is nearly one body diameter.

Comparisons of Data from NEAR High Angle-of-Attack Tests With Prediction of Program MISSILE2

In this section, comparisons of data with the predictions of Program MISSILE2 (reference 9) are presented for another canard configuration with inline tail fins. The Navy nose, canards and tail fins shown in figure 3 were used. The comparisons are for subsonic and supersonic speeds ($M_\infty = 0.8$, 1.3), several roll angles and with and without fin deflection. Since Program MISSILE is especially useful for the determination of lateral stability and control-cross coupling effects, comparisons are presented for the induced rolling-moment coefficient.

Results for zero fin deflection at roll angles of 10° and 30° are presented in figure 11. Note that the data taken in the NASA/ARC 6- by 6-Foot Wind Tunnel and NASA/ARC 11-Foot Wind Tunnel overlap for angles of attack between 20 and 24°. The agreement for the two tunnels is satisfactory. The agreement for the roll angle case of 30° is excellent for angles of attack up to 40° even to the point of reproducing the Mach number effects between angles of attack of

15° and 35°. For 10° roll, the agreement is acceptable for angles of attack up to 40° for purposes of design.

Results for 15° deflection of the yaw control fins at roll angles of 0° and 30° are presented in figure 12. Again, agreement of data from the two wind tunnels is satisfactory. For this case, the theory predicts the overall level and trends of induced roll adequately for design purposes. However, definite improvement of the program is needed for the higher angle of attack range.

APPENDIX

Description of Subroutines

MAIN - This is the driver for the rest of the program. It begins by opening the random access file (logical unit 3) and initializing counters and flags. Then, for each frame, MAIN asks the user for the independent variable name and the variable MODE is set. Next, subroutine MYSTRO is called to produce a frame. When the frame is complete, control returns to MAIN which then responds to user instructions to tabulate and/or plot the frame. See flow chart in figure 13.

MYSTRO - This subroutine does most of the interaction with the user. Its job is to obtain a description of the desired frame and to direct its production. See flow chart in figure 14.

CREATO - Given a dataset name, this subroutine first searches a list of existing datasets for that name. If the dataset name appears on the list, the variable DSNUM is given an appropriate value and control is returned to MYSTRO. If the random access dataset does not exist, a DATAMAN tape is searched, a new random access dataset is created, and its name is added to the list of existing dataset names. See flow chart in figure 15.

FINDER - This subroutine reads the random access file created by CREATO. Given a dependent variable name, FINDER fills the arrays A, B, and C with data from the current dataset. Array A contains first independent variable data, B contains second independent variable data, and C is an array of dependent variable data. This routine also retrieves identifying information for the dataset such as the configuration, date of creation, and the names and values of the parameters. See flow chart in figure 16.

ABC - Subroutine ABC processes the arrays A, B, and C which are provided by subroutine FINDER. Depending on the independent variable of the current frame, this routine writes information into arrays X and Y which represent the abscissa and ordinate values to be plotted. Linear interpolation is done between the discrete elements of A, B, and C. See flow chart in figure 17.

SORTOR - This subroutine is called only if roll angle is the independent variable. Its purpose is to put the plotting ordinates in ascending order.

TABULA - This routine directs the formatting and listing of the current frame at the terminal.

PLOTOR - Subroutine PLOTOR is called if a plot file is to be created. After asking the user for scaling information, it directs the production of an SGP plot file.

Dictionary of Common Block Variables

Common block BETA contains information concerning the dataset currently being processed. Its variables are defined in subroutine FINDER and processed by subroutine ABC.

A	an array of first independent variables
B	an array of second independent variables

C	an array of data for a specific dependent variable
NBP	an array contaning the number of entries recorded in each row of A, B, and C
SETNAM	name of current dataset
CONFIG	configuration of current dataset
DVNAM	name of dependent variable under consideration
NMN	number of runs in dataset
MAXNAA	largest value in array NBP
NPARM	number of parameters in current dataset
NAMPAR	names of parameters
PVAL	parametric values
FEEVAL	value of roll angle in parameter list
TMACH	Mach number under consideration
TALPH	angle of attack under consideration

Common block COUNT contains counters and flags.

MODE	flag indicating independent variable for current frame (1 = first independent variable, 2 = second independent variable, and 3 = roll angle)

NCURV	number of curves on current frame
NSETS	number of datasets used to make current frame
NAPARM	number of active parameters
NFEE	number of datasets used when roll angle is the independent variable
MODP	flag indicating active parameter when roll angle is the independent variable (1 = first independent variable, 2 = second independent variable)
NPT	number of points plotted when roll angle is the independent variable
FLAG	flag indicating an error condition

Common block CURVS contains information defining the curves to be plotted. Its variables are set in subroutine ABC and used by PLOTOR.

X	array of abscissae to be plotted
Y	array of ordinates to be plotted
NP	number of points in each curve
AVAL	values of the active parameter for each curve
CPAR	value of parameter held constant

Common block FILE3 appears in all subroutines that read or write to the random access data base, logical unit 3.

NREC3	the variable associated with unit 3. It is incremented by one after each read or write operation on unit 3
NEXREC	after each data base is written on unit 3, this variable is set to the next available record number

Common block IDENTS contains identifiers for the frame currently being constructed.

SETID	the name of dataset input by the user
VAR	contains the names of both independent variables for the current dataset
IVID	independent variable name for the current frame
DVID	dependent variable name input by the user
ACPARM	values of the active parameters input by the user
NAMES	the names of the datasets to be used when roll angle is the independent variable

Common block NDEX contains data used to index the random access datasets which are created by subroutine CREATO.

INDEX	an array of indices for the random access data base

DSLIST	a list of existing random access datasets
DSNUM	the identifying number of the dataset currently being used
SETCNT	the number of datasets in DSLIST

Common block TEXT contains textual identifiers which appear on plots and tabulations of the current frame.

IDCURV	an array containing the dependent variable name and dataset name associated with each curve in the current frame
IDSETS	a list of datasets used in the current frame
IDCFIG	a list of configurations associated with the datasets in IDSETS
IDPARMS	an array containing a list of parameters for each dataset in IDSETS
NPARMS	the number of entirs in IDPARMS
PARVAL	the parametric values for each parameter in IDPARMS

Program Listing

```
C
C          PROGRAM DATAMANIA
C          540/C
C
      IMPLICIT INTEGER (A-Z)
C
      COMMON /COUNT/ MODE,NCURV,NSETS,NAPARM,NFEE,MODP,NPT,FLAG,
     *               IVFLAG,ACFLAG,MACP
      COMMON /IDENTS/ SETID(3),VAR(2,3),IVID(3),DVID(3),ACPARM(5),
     *                NAMES(10,3)
      COMMON /FILE3/ NREC3,NEXREC
      COMMON /NDEX/ INDEX(14,100),DSLIST(100,3),DSNUM,SETCNT
C
      DATA YES  /1HY/,
     *     MOCK /2HMA/,
     *     ALFA /2HAL/,
     *     FEE  /2HPH/,
     *     CRT  /2HCR/,
     *     ZERO /0/
C
      OPEN (UNIT=3,TYPE='NEW',ACCESS='DIRECT',FORM='UNFORMATTED',
     *      RECORDSIZE=64,INITIALSIZE=100,DISPOSE='DELETE',
     *      ASSOCIATEVARIABLE=NREC3)
C
      NEXREC=1
      SETCNT=0
      FLAG=0
      DSNUM=0
C
C          WRITE DATAMANIA HEADER AND PROMPT FOR INSTRUCTIONS
C
      WRITE(5,5001)
      READ (5,5002) ANSWER
      IF (ANSWER.EQ.YES) CALL NSTRUC
C
C          PROMPT FOR INDEPENDENT VARIABLE
C
   50 NSETS=0
      NCURV=0
      MODE=0
      WRITE(5,5003)
      READ (5,5009) IVID
      IF (IVID(1).EQ.ALFA .OR. IVID(1).EQ.MOCK .OR.
     *    IVID(1).EQ.CRT  .OR. IVID(1).EQ.FEE) GO TO 100
C
C
C          UNACCEPTABLE IND. VAR. - WRITE MESSAGE, TRY AGAIN
C
      WRITE(5,5004)
      GO TO 50
C
  100 CALL MYSTRO
      IF (FLAG.EQ.ZERO) GO TO 150
      FLAG=0
      WRITE(5,5010)
      GO TO 50
C
C          PROMPT FOR TABULATION
```

```
C
  150 CONTINUE
      WRITE(5,5005)
      READ (5,5002) ANSWER
      IF (ANSWER.EQ.YES) CALL TABULA
C
C           PROMPT FOR PLOT FILE
C
      WRITE(5,5006)
      READ (5,5002) ANSWER
      IF (ANSWER.EQ.YES) CALL PLOTOR
C
C           PROMPT FOR MORE
C
      WRITE(5,5007)
      READ (5,5002) ANSWER
      IF (ANSWER.EQ.YES) GO TO 50
C
C           END OF SESSION
C
      WRITE(5,5008)
C
      STOP
C
C************************FORMATS********************************
C
 5001 FORMAT(////1X,10(1H*),22H WELCOME TO DATAMANIA ,10(1H*)////
     *       27H$DO YOU WANT INSTRUCTIONS? )
 5002 FORMAT(A1)
 5003 FORMAT(/35H$ENTER INDEPENDENT VARIABLE NAME - )
 5004 FORMAT(/43H  INDEPENDENT VARIABLE NAME IS UNACCEPTABLE/
     *59H  THE FOUR ACCEPTABLE IND.VARIABLES ARE ALPHA, MACH NUMBER,/
     *59H  CRT, AND PHI.  DATAMANIA READS ONLY THE FIRST TWO LETTERS/
     *54H  OF YOUR RESPONSE SO YOU MAY ENTER AL, MA, CR, OR PH.)
 5005 FORMAT(/16H$TABULATE DATA? )
 5006 FORMAT(/19H$CREATE PLOT FILE? )
 5007 FORMAT(/29H$WANT TO MAKE ANOTHER FRAME? )
 5008 FORMAT(//1X,13(1H*),16H END OF SESSION ,13(1H*)//)
 5009 FORMAT(3A2)
 5010 FORMAT(//45H IND. VAR. NAME UNACCEPTABLE FOR THIS DATASET/)
C
C***************************************************************
C
      END
```

```
      SUBROUTINE ABC
C
C         THIS SUBROUTINE EXTRACTS DATA FROM THE ARRAYS
C         GENERATED BY FINDER AND PREPARES IT FOR USE
C         BY THE PLOTTING AND TABULATING ROUTINES
C
      COMMON /COUNT/ MODE,NCURV,NSETS,NAPARM,NFEE,MODP,NPT,FLAG,
     *               IVFLAG,ACFLAG,MACP
      COMMON /BETA/  A(10,20),B(10,20),C(10,20),NBP(10),
     *              SETNAM(3),CONFIG(24),DVNAM(3),NMN,MAXNAA,NPARM,
     *              NAMPAR(3,10),PVAL(10),FEEVAL,TMACH,TALPH
      COMMON /IDENTS/ SETID(3),VAR(2,3),IVID(3),DVID(3),ACPARM(5),
     *                NAMES(10,3)
      COMMON /CURVS/ X(20,5),Y(20,5),NP(5),AVAL(5),CPAR
      COMMON /TEXT/ IDCURV(6,5),IDSETS(3,10),IDCFIG(24,5),
     *             IDPARM(3,10,5),NPARMS(5),PARVAL(10,5)
C
      INTEGER SETNAM,CONFIG,DVNAM,SETID,DVID,ACFLAG,VAR
C
      YLINE(OR1,AB1,OR2,AB2,ORP)=AB1+(AB2-AB1)*(ORP-OR1)/(OR2-OR1)
C
      DATA EPS /0.04/
C
      NPARMS(NSETS)=NPARM
C
      DO 10  I=1,3
      IDSETS(I,NSETS)=SETNAM(I)
   10 CONTINUE
C
      DO 20  I=1,24
      IDCFIG(I,NSETS)=CONFIG(I)
   20 CONTINUE
C
      DO 30  M=1,NPARM
      PARVAL(M,NSETS)=PVAL(M)
      DO 30  I=1,3
      IDPARM(I,M,NSETS)=NAMPAR(I,M)
   30 CONTINUE
C
      IF (MODE.EQ.3) GO TO 500
C
      DO 400  N=1,NAPARM
      NCURV=NCURV+1
      AVAL(NCURV)=ACPARM(N)
C
      DO 40  I=1,3
      IDCURV(I,NCURV)=DVNAM(I)
      IDCURV(I+3,NCURV)=SETNAM(I)
   40 CONTINUE
C
C          USE 2ND IND. VAR.
C
      IF (MODE.EQ.1) GO TO 200
C
      DO 50  I=1,NMN
      IF (ABS(ACPARM(N)-A(I,1)).LT.EPS) GO TO 100
   50 CONTINUE
```

```
C
      I=I-1
  100 INC1=I
      NP(NCURV)=NBP(INC1)
C
      DO 150  J=1,NBP(INC1)
      X(J,NCURV)=B(INC1,J)
      Y(J,NCURV)=C(INC1,J)
  150 CONTINUE
C
      GO TO 400
C
C           USE 1ST IND. VAR.
C
  200 CONTINUE
C
      NP(NCURV)=NMN
C
         DO 330  M=1,NMN
            DO 300  I=2,MAXNAA
            II=I
            IF (ACPARM(N).LE.B(M,I)) GO TO 350
  300       CONTINUE
C
  350 I=II-1
         X(M,NCURV)=(A(M,II)+A(M,I))*0.5
         Y(M,NCURV)=YLINE(B(M,I),C(M,I),B(M,II),C(M,II),ACPARM(N))
  330   CONTINUE
  400 CONTINUE
C
      RETURN
C
C          IND.VAR. IS PHI
C
  500 CONTINUE
C
      DO 750  N=1,NAPARM
      NCURV=N
      AVAL(NCURV)=ACPARM(N)
      DO 530  I=1,3
      IDCURV(I,NCURV)=DVNAM(I)
  530 CONTINUE
C
      IF (MODP.EQ.1) TALPH=ACPARM(N)
      IF (MODP.EQ.2) TMACH=ACPARM(N)
C
         DO 550  I=2,NMN
         II=I
         IF (TMACH.LE.A(I,1)) GO TO 600
  550    CONTINUE
C
  600 I=II-1
C
         DO 650  J=2,MAXNAA
         JJ=J
         IF (TALPH.LE.B(I,J)) GO TO 700
  650    CONTINUE
```

```
C
  700 J=JJ-1
C
      X(NPT,NCURV)=FEEVAL
C
      C13=YLINE(B(I,J),C(I,J),B(I,JJ),C(I,JJ),TALPH)
      C24=YLINE(B(II,J),C(II,J),B(II,JJ),C(II,JJ),TALPH)
C
      Y(NPT,NCURV)=YLINE(A(I,1),C13,A(II,1),C24,TMACH)
C
  750 CONTINUE
C
      RETURN
C
      END
```

```
      SUBROUTINE CREATO
C
C          THIS ROUTINE CREATES A DATABASE AND AN INDEX
C          FOR THE DATASET UNDER CONSIDERATION.
C
      IMPLICIT INTEGER (A-Z)
C
      COMMON /COUNT/ MODE,NCURV,NSETS,NAPARM,NFEZ,MODP,NPT,FLAG,
     *               IVFLAG,ACFLAG,MACP
      COMMON /FILE3/ NREC3,NEXREC
      COMMON /IDENTS/ SETID(3),VAR(2,3),IVID(3),DVID(3),ACPARM(5),
     *                NAMES(10,3)
      COMMON /NDEX/ INDEX(14,100),DSLIST(100,3),DSNUM,SETCNT
C
      REAL DUM(30),XLIM(2),APLIM(2),DLIM(10,2),PARVAL(10),
     *     A(10,20),ACPARM
C
      DIMENSION DSNAM(3),DATE(6),CONFIG(24),DVAR(10,3),
     *          PARMS(10,3),RUNNO(10,2),NAAX(10)
C
      DATA YES /1HY/
C
      IF (SETCNT.EQ.0) GO TO 10
C
C          SEARCH ARRAY OF EXISTING DATASETS
C
C
      DO 5  N=1,SETCNT
      DSNUM=N
      IF (SETID(1).NE.DSLIST(N,1)) GO TO 5
      IF (SETID(2).NE.DSLIST(N,2)) GO TO 5
      IF (SETID(3).EQ.DSLIST(N,3)) GO TO 65
    5 CONTINUE
C
C          SEARCH FOR DATASET IDENTIFIER
C
   10 READ(1,1001,END=70) DSNAM
      IF (DSNAM(1).NE.SETID(1)) GO TO 10
      IF (DSNAM(2).NE.SETID(2)) GO TO 10
      IF (DSNAM(3).NE.SETID(3)) GO TO 10
C
C          YOU'VE FOUND IT - NOW CREATE
C
      BACKSPACE 1
      SETCNT=SETCNT+1
      DSNUM=SETCNT
C
      DO 20  I=1,3
   20 DSLIST(DSNUM,I)=DSNAM(I)
C
      READ(1,1002) DSNAM,DATE,CONFIG,NMN,MAXNAA,NDV,NPAR
C
      READ(1,1003) ((VAR(I,J),J=1,3),I=1,2),
     *             ((DVAR(I,J),J=1,3),I=1,NDV),
     *             ((PARMS(I,J),J=1,3),I=1,NPAR),
     *             (DUM(I),I=1,30)
C
      READ(1,1004) ((RUNNO(I,J),I=1,NMN),J=1,2),(NAAX(I),I=1,NMN)
```

```
C
      READ(1,1005) (DUM(I),I=1,2),(XLIM(I),I=1,2),(APLIM(I),I=1,2),
     *             ((DLIM(I,J),I=1,NDV),J=1,2),
     *             (PARVAL(I),I=1,NPAR),(DUM(I),I=1,7)
C
      NREC3=NEXREC
      INDEX(1,DSNUM)=NREC3
      WRITE(3'NREC3)DSNAM,DATE,CONFIG,NMN,MAXNAA,NDV,NPAR,
     *               ((PARMS(I,J),J=1,3),I=1,NPAR),(PARVAL(I),I=1,NPAR)
C
      INDEX(2,DSNUM)=NREC3
      WRITE(3'NREC3) (VAR(1,J),J=1,3),XLIM
C
      READ(1,1005) ((A(M,I),I=1,MAXNAA),M=1,NMN)
C
      DO 30  M=1,NMN
   30 WRITE (3'NREC3) (A(M,I),I=1,MAXNAA)
C
      INDEX(3,DSNUM)=NREC3
      WRITE(3'NREC3)(VAR(2,J),J=1,3),APLIM
C
      READ(1,1005) ((A(M,I),I=1,MAXNAA),M=1,NMN)
C
      DO 40  M=1,NMN
   40 WRITE(3'NREC3) NAAX(M),(A(M,I),I=1,MAXNAA)
C
      READ(1,1005) (DUM(I),I=1,NMN)
C
      INDEX(4,DSNUM)=NREC3
      WRITE(3'NREC3)(DUM(I),I=1,NMN),((RUNNO(I,J),I=1,NMN),J=1,2)
C
      DO 60  K=1,NDV
      KK=K+4
      INDEX(KK,DSNUM)=NREC3
      WRITE(3'NREC3) (DVAR(K,I),I=1,3),(DLIM(K,I),I=1,2)
C
      READ(1,1005) ((A(M,I),I=1,MAXNAA),M=1,NMN),(DUM(I),I=1,NMN)
C
         DO 50  M=1,NMN
   50    WRITE(3'NREC3)(A(M,I),I=1,MAXNAA)
   60 CONTINUE
      NEXREC=NREC3
      REWIND 1
C
C           SUMMARY?
C
   65 WRITE(5,5001) SETID
      READ (5,5002) ANSWER
      IF (ANSWER.EQ.YES) CALL SUMMRY
C
C            GET NAMES OF INDEPENDENT VARIABLES
C
      DO 68  I=1,2
      NREC3=INDEX(I+1,DSNUM)
      READ(3'NREC3) (VAR(I,J),J=1,3)
   68 CONTINUE
C
      RETURN
```

```
C
   70 CONTINUE
      FLAG=1
      REWIND 1
      RETURN
C
C*************************FORMATS*******************************
C
 1001 FORMAT(6X,3A2)
 1002 FORMAT(6X,33A2,12X,4I6)
 1003 FORMAT(66A2)
 1004 FORMAT(22I6)
 1005 FORMAT(8E15.8)
 5001 FORMAT(/26H$DO YOU WANT A SUMMARY OF ,3A2,2H? )
 5002 FORMAT(A1)
C
C**************************************************************
C
      END
```

```
      SUBROUTINE FINDER
C
C         THIS SUBROUTINE GETS BLOCKS OF DATA FROM
C         A DATABASE FOR USE BY SUBROUTINE ABC
C
      IMPLICIT INTEGER (A-Z)
C
      COMMON /IDENTS/ SETID(3),VAR(2,3),IVID(3),DVID(3),ACPARM(5),
     *                NAMES(10,3)
      COMMON /COUNT/ MODE,NCURV,NSETS,NAPARM,NFEE,MODP,NPT,FLAG,
     *               IVFLAG,ACFLAG,MACP
      COMMON /FILE3/ NREC3,NEXREC
      COMMON /NDEX/ INDEX(14,100),DSLIST(100,3),DSNUM,SETCNT
      COMMON /BETA/ A(10,20),B(10,20),C(10,20),NBP(10),
     *              SETNAM(3),CONFIG(24),DVNAM(3),NMN,MAXNAA,NPARM,
     *              NAMPAR(3,10),PVAL(10),FEEVAL,TMACH,TALPH
C
      REAL A,B,C,PVAL,FEEVAL,TMACH,TALPH,ACPARM
C
      DIMENSION FEE(3),DATE(6)
C
      DATA  FEE /2HPH,2HI ,2H  /
C
      NREC3=INDEX(1,DSNUM)
      READ (3'NREC3) SETNAM,DATE,CONFIG,NMN,MAXNAA,NDV,NPARM,
     *               ((NAMPAR(I,J),I=1,3),J=1,NPARM),(PVAL(J),J=1,NPARM)
C
      IF (MODE.NE.3) GO TO 200
C
      M=0
      DO 150  N=1,NPARM
      IF (NAMPAR(1,N).NE.FEE(1)) GO TO 50
      IF (NAMPAR(2,N).NE.FEE(2)) GO TO 50
      IF (NAMPAR(3,N).NE.FEE(3)) GO TO 50
C
      FEEVAL=PVAL(N)
      GO TO 150
   50 M=M+1
C
         DO 100  I=1,3
         NAMPAR(I,M)=NAMPAR(I,N)
  100    CONTINUE
C
  150 CONTINUE
C
      NPARM=M
C
  200 CONTINUE
      NREC3=INDEX(2,DSNUM)+1
      DO 250  I=1,NMN
      READ(3'NREC3) (A(I,J),J=1,MAXNAA)
  250 CONTINUE
C
      NREC3=INDEX(3,DSNUM)+1
      DO 300 I=1,NMN
      READ(3'NREC3) NBP(I),(B(I,J),J=1,NBP(I))
  300 CONTINUE
```

```
C
      DO 350  K=1,NDV
      NREC3=INDEX(K+4,DSNUM)
      READ(3'NREC3) DVNAM
         IF (DVNAM(1).NE.DVID(1)) GO TO 350
         IF (DVNAM(2).NE.DVID(2)) GO TO 350
         IF (DVNAM(3).EQ.DVID(3)) GO TO 400
  350 CONTINUE
C
C          NOT FOUND - SET FLAG AND RETURN
C
      FLAG=1
      RETURN
C
  400 CONTINUE
      DO 450  I=1,NMN
      READ(3'NREC3) (C(I,J),J=1,MAXNAA)
  450 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE MYSTRO
C
C          THIS SUBROUTINE PROMPTS THE USER FOR PLOT
C          SPECIFICATIONS AND CONTROLS THE PRODUCTION OF
C          THE CURRENT FRAME.
C
      IMPLICIT INTEGER (A-Z)
C
      COMMON /COUNT/ MODE,NCURV,NSETS,NAPARM,NFEE,MODP,NPT,FLAG,
     *               IVFLAG,ACFLAG,MACP
      COMMON /IDENTS/ SETID(3),VAR(2,3),IVID(3),DVID(3),ACPARM(5),
     *                NAMES(10,3)
      COMMON /NDEX/ INDEX(14,100),DSLIST(100,3),DSNUM,SETCNT
      COMMON /CURVS/  X(20,5),Y(20,5),NP(5),AVAL(5),CPAR
      COMMON /TEXT/IDCURV(6,5),IDSETS(3,10),IDCFIG(24,5),
     *              IDPARM(3,10,5),NPARMS(5),PARVAL(10,5)
      COMMON /BETA/ A(10,20),B(10,20),C(10,20),NBP(10),
     *              SETNAM(3),CONFIG(24),DVNAM(3),NMN,MAXNAA,NPARM,
     *              NAMPAR(3,10),PVAL(10),FEEVAL,TMACH,TALPH
C
      DIMENSION FELIST(10)
C
      REAL ACPARM,AVAL,CPAR,X,Y,A,B,C,PARVAL,PVAL,FEEVAL,TMACH,TALPH
C
      DATA YES  /1HY/,
     *     MOCK /2HMA/,
     *     ALFA /2HAL/,
     *     CRT  /2HCR/,
     *     FEE  /2HPH/,
     *     ZERO/0/
C
      IF (IVID(1).EQ.FEE) GO TO 300
C
C          I.V. IS ALPHA, MACH NUMBER, OR CRT
C          PROMPT FOR DATASET NAME
C
   50 NSETS=NSETS+1
      WRITE(5,5001)
      READ (5,5002) SETID
C
      CALL CREATO
      IF (FLAG.EQ.ZERO) GO TO 100
      FLAG=0
      WRITE (5,5003) SETID
      GO TO 50
C
C       PROMPT FOR DEP. VAR.
C
  100 WRITE(5,5005)
      READ (5,5002) DVID
C
C
      CALL FINDER
      IF (FLAG.EQ.ZERO) GO TO 250
      FLAG=0
      WRITE(5,5018) DVID,SETID
      GO TO 100
  250 CONTINUE
```

```
C
       IF (IVID(1).EQ.VAR(1,1)) GO TO 260
       IF (IVID(1).EQ.VAR(2,1)) GO TO 270
C
C           I.V. NOT FOUND  -  RETURN
C
      FLAG=1
      RETURN
C
  260 IVFLAG=1
      ACFLAG=2
      MODE=1
      GO TO 280
C
  270 IVFLAG=2
      ACFLAG=1
      MODE=2
C
  280 CONTINUE
      WRITE(5,5006) (VAR(ACFLAG,I),I=1,3)
      READ(5,5011) NAPARM
      WRITE(5,5016) NAPARM,(VAR(ACFLAG,I),I=1,3)
      READ(5,5008) (ACPARM(I),I=1,NAPARM)
C
C           PROCESS DATA ACCORDING TO MODE
C
      CALL ABC
C
C           ANYMORE ON THIS FRAME?
C
      WRITE(5,5009)
      READ (5,5004) ANSWER
      IF (ANSWER.NE.YES) GO TO 290
C
C            SAME DATASET?
C
      WRITE(5,5015)
      READ(5,5004) ANSWER
      IF (ANSWER.EQ.YES) GO TO 100
      GO TO 50
C
  290 RETURN
C
C           PHI IS I.V. - PROMPT FOR NUMBER OF DATASETS
C
  300 MODE=3
      WRITE(5,5010)
      READ (5,5011) NFEE
C
C           READ DATASET NAMES
C
      WRITE(5,5012) NFEE
      DO 350  N=1,NFEE
  350 READ (5,5002) (NAMES(N,I),I=1,3)
```

```
C
      DO 380  N=1,NFEE
         DO 360  I=1,3
         SETID(I)=NAMES(N,I)
  360    CONTINUE
C
      CALL CREATO
      FELIST(N)=DSNUM
  380 CONTINUE
C
C           READ DEP. VAR. NAME
C
      WRITE(5,5005)
      READ (5,5002) DVID
C
C           READ INVARIANT PARAMETER NAME AND VALUE
C
  400 MODP=0
      WRITE(5,5013) ((VAR(I,J),J=1,3),I=1,2)
      READ (5,5002) CONPAR
      IF (CONPAR.EQ.VAR(1,1)) MODP=1
      IF (CONPAR.EQ.VAR(2,1)) MODP=2
C
C           READ ACTIVE PARAMETRIC VALUES
C
  450 WRITE(5,5014)
      IF (MODP.EQ.1) GO TO 500
      READ (5,5008) TALPH
      CPAR=TALPH
      WRITE(5,5006) (VAR(1,J),J=1,3)
      READ (5,5011) NAPARM
      WRITE(5,5016) NAPARM,(VAR(1,J),J=1,3)
      GO TO 550
  500 READ (5,5008) TMACH
      CPAR=TMACH
      WRITE(5,5006) (VAR(2,J),J=1,3)
      READ (5,5011) NAPARM
      WRITE(5,5016) NAPARM,(VAR(2,J),J=1,3)
  550 READ (5,5008) (ACPARM(I),I=1,NAPARM)
C
      DO 650  N=1,NFEE
      NSETS=NSETS+1
C
      DSNUM=FELIST(N)
C
      CALL FINDER
C
      NPT=N
      CALL ABC
  650 CONTINUE
C
      NCURV=NAPARM
      CALL SORTOR
      RETURN
```

```
C
C*****************FORMATS***********************
C
 5001 FORMAT(/28H$ENTER DATASET IDENTIFIER - )
 5002 FORMAT(3A2)
 5003 FORMAT(/1X,29HDATAMANIA CAN'T FIND DATASET ,3A2/)
 5004 FORMAT(A1)
 5005 FORMAT(/33H$ENTER DEPENDENT VARIABLE NAME - )
 5006 FORMAT(/27H$ENTER NUMBER OF VALUES OF ,3A2,17H TO BE PLOTTED - )
 5008 FORMAT(5F10.0)
 5009 FORMAT(/30H$ANY MORE DATA ON THIS FRAME? )
 5010 FORMAT(/44H$ENTER NUMBER OF PHI VALUES TO BE PLOTTED - )
 5011 FORMAT(I1)
 5012 FORMAT(/1X,6HENTER ,I1,14H DATASET NAMES/
     *        1X,32HEACH FOLLOWED BY CARRIAGE RETURN)
 5013 FORMAT(/38H$ENTER PARAMETER TO BE HELD CONSTANT (,3A2,4H OR ,
     *       3A2,4H) - )
 5014 FORMAT(/37H$ENTER VALUE OF CONSTANT PARAMETER - )
 5015 FORMAT(/19H$USE SAME DATASET? )
 5016 FORMAT(/6H$ENTER,I2,11H VALUES OF ,3A2,
     *       26H (SEPARATE WITH COMMAS) - )
 5018 FORMAT(/1X,3A2,28H IS NOT A DEPENDENT VARIABLE
     *       /1X,11HIN DATASET ,3A2,13H.  TRY AGAIN./)
C
C***********************************************
C
      END
```

```
      SUBROUTINE NSTRUC
C
C          THIS SUBROUTINE WRITES INSTRUCTIONS FOR THE USE OF PROGRAM
C          DATAMANIA
C
      WRITE(5,5001)
      WRITE(5,5002)
      WRITE(5,5003)
 5001 FORMAT(///1X,48H     DATAMANIA IS DESIGNED TO READ DATAMAN TAPES
     *  /1X,55HAND PRODUCE PLOTS OF COEFFICIENT DATA AS A FUNCTION OF
     *  /1X,52HEITHER ANGLE OF ATTACK (ALPHA), MACH NUMBER, OR ROLL
     *  /1X,55HANGLE (PHI).  IN ORDER TO DO THIS, YOU NEED TO KNOW THE
     *  /1X,54HNAMES OF THE DATASETS WHICH CONTAIN THE INFORMATION TO
     *  /1X,11HBE PLOTTED.)
 5002 FORMAT(/1X,51H     AT THE BEGINNING OF EACH FRAME, DATAMANIA WILL
     *  /1X,55HPROMPT FOR THE INDEPENDENT VARIABLE NAME. YOUR RESPONSE
     *  /1X,49HOF ALPHA, MACH, OR PHI WILL DETERMINE THE MODE OF
     *  /1X,55HPROGRAM OPERATION.  FOLLOWING THIS YOU WILL BE PROMPTED
     *  /1X,45HFOR INFORMATION NEEDED TO COMPLETE THE FRAME.)
 5003 FORMAT(/1X,49H     AS EACH FRAME IS COMPLETED YOU WILL BE ASKED
     *  /1X,52HFIRST IF YOU WANT A TABULATION OF THE CURRENT FRAME,
     *  /1X,50HAND THEN, IF YOU WANT TO CREATE A PLOT FILE OF THE
     *  /1X,6HFRAME.////)
C
      RETURN
      END
```

```
      SUBROUTINE PLOTOR
C
C          THIS SUBROUTINE CREATES AN SGP PLOT FILE
C          FROM THE DATA PRODUCED BY SUBROUTINE ABC
C
      IMPLICIT INTEGER (A-Z)
C
      COMMON /CURVS/ X(20,5),Y(20,5),NP(5),AVAL(5),CPAR
      COMMON /TEXT/ IDCURV(6,5),IDSETS(3,10),IDCFIG(24,5),
     *              IDPARM(3,10,5),NPARMS(5),PARVAL(10,5)
      COMMON /COUNT/ MODE,NCURV,NSETS,NAPARM,NFEE,MODP,NPT,FLAG,
     *               IVFLAG,ACFLAG,MACP
      COMMON /IDENTS/ SETID(3),VAL(2,3),IVID(3),DVID(3),ACPARM(5),
     *                NAMES(10,3)
      COMMON /BSCALE/ XMAX,XMIN,YMAX,YMIN
C
      DIMENSION CNTRL(2,5),XLABEL(4),YLABEL(4),SYMB(5)
      DIMENSION NAME(3),SHAP1(12),SHAP2(12),PARM(3),CURV1(3),CURV2(3)
      DIMENSION ZLABEL(4),PLABEL(4)
      DIMENSION WLABEL(4),SHAPE(24)
C
      REAL X,Y,PARVAL,AVAL,CPAR,XMAX,XMIN,YMAX,YMIN
```

```
C
      DATA CNTRL /350,0,180,340,90,260,24,213,10,170/
      DATA NDP,IPNT,JPNT,STAR /20,2,1,2H* /
      DATA YLABEL /2H *,2H  ,2H  ,2H  /
      DATA PLABEL /2H  ,2HPH,2HI ,2H* /
      DATA SYMB /2H0 ,2H1 ,2H2 ,2H3 ,2H4 /
      DATA YES /1HY/
C
C          SET SCALING OPTIONS
C
      IOPT=2
      WRITE(5,5001)
      READ (5,5002) ANSWER
      IF (ANSWER.EQ.YES) GO TO 99
C
      IOPT=1
      WRITE(5,5003)
      READ(5,5002) ANSWER
      IF (ANSWER.EQ.YES) GO TO 99
C
C          READ USER DEFINED SCALING VALUES
C
      WRITE(5,5004)
      READ(5,5005) XMIN,XMAX
      WRITE(5,5006)
      READ(5,5005) YMIN,YMAX
C
C          SET LABELS
C
   99 CONTINUE
      IF (MODE.GT.2) GO TO 4
      DO 1  I=1,3
      XLABEL(I)=VAR(IVFLAG,I)
      ZLABEL(I)=VAR(ACFLAG,I)
    1 CONTINUE
      XLABEL(4)=STAR
      ZLABEL(4)=STAR
      GO TO 7
C
C
    4 CONTINUE
      DO 5  I=1,4
      XLABEL(I)=PLABEL(I)
    5 CONTINUE
C
      DO 6  I=1,NCURV
      NP(I)=NFEE
    6 CONTINUE
C
    7 CONTINUE
C
      CALL PLOTI(X,Y,IOPT,NP,NDP,NCURV,IPNT,JPNT,XLABEL,YLABEL)
C
      IF (MODE.EQ.3) GO TO 90
C
      XINC=CNTRL(2,NSETS)
      XLOC=CNTRL(1,NSETS)
```

```
C
      DO 50  N=1,NSETS
C
         DO 10  I=1,3
   10    NAME(I)=IDSETS(I,N)
         DO 20  I=1,12
         SHAP1(I)=IDCFIG(I,N)
   20    SHAP2(I)=IDCFIG(I+12,N)
C
      CALL TXT(XLOC,1100,NAME,6,0)
      CALL TXT(XLOC,1085,SHAP1,24,0)
      CALL TXT(XLOC,1070,SHAP2,24,0)
C
      YLOC=1055
      NUMLOC=XLOC+50
         DO 40  M=1,NPARMS(N)
C
            DO 30  I=1,3
   30       PARM(I)=IDPARM(I,M,N)
C
         CALL TXT(XLOC,YLOC,PARM,6,0)
         CALL NUMB(NUMLOC,YLOC,PARVAL(M,N),6H(F6.2),0)
C
         YLOC=YLOC-15
   40    CONTINUE
C
      XLOC=XLOC+XINC
C
   50 CONTINUE
C
      MAX=0
      DO 60  I=1,NCURV
      MAX=MAX0(MAX,NPARMS(I))
   60 CONTINUE
C
      XLOC=200
      YLOC=1040-MAX*15
C
      CALL TXT(XLOC,YLOC,28HSYMBOL   DV          DATASET,28,0)
      CALL TXT(300,YLOC,ZLABEL,6,0)
      YLOC=YLOC-8
      CALL TXT(XLOC,YLOC,28H------   --   ----   -------,28,0)
C
      DO 80  N=1,NCURV
C
         DO 70  I=1,3
         CURV1(I)=IDCURV(I,N)
         CURV2(I)=IDCURV(I+3,N)
   70    CONTINUE
C
      YLOC=YLOC-15
      CALL TXT(215,YLOC,SYMB(N),2,0)
      CALL TXT(265,YLOC,CURV1,6,0)
      CALL NUMB(300,YLOC,AVAL(N),6H(F6.2),0)
      CALL TXT(360,YLOC,CURV2,6,0)
   80 CONTINUE
```

```
C
      GO TO 450
C
C           IND.VAR. IS PHI
C
   90 CONTINUE
C
C
      DO 92  I=1,3
      ZLABEL(I)=VAR(MACP,I)
      WLABEL(I)=VAR(MODP,I)
   92 CONTINUE
C
C
C
      DO 100  I=1,3
      CURV1(I)=IDCURV(I,1)
      CURV2(I)=IDCURV(I+3,1)
  100 CONTINUE
C
      CALL TXT (350,1100,CURV1,6,0)
      CALL TXT (400,1100,5H VS. ,5,0)
      CALL TXT (440,1100,PLABEL,6,0)
C
      DO 150  I=1,24
      SHAPE(I)=IDCFIG(I,1)
  150 CONTINUE
C
      CALL TXT (200,1070,16 CONFIGURATION - ,16,0)
      CALL TXT (320,1070, IAPE,48,0)
C
      CALL TXT (125,1030,6HSYMBOL,6,0)
      CALL TXT (175,1030,ZLABEL,6,0)
      CALL TXT (300,1030,17HPARAMETRIC VALUES,17,0)
      CALL TXT (600,1030,8HDATASETS,8,0)
C
      YLOC=1010
C
      DO 200  N=1,NCURV
      CALL TXT (140,YLOC,SYMB(N),2,0)
      CALL NUMB(180,YLOC,AVAL(N),6H(F5.2),0)
      YLOC=YLOC-15
  200 CONTINUE
C
      YLOC=1010
C
      DO 300  N=1,NPARMS(1)
         DO 250  I=1,3
         PARM(I)=IDPARM(I,N,1)
  250    CONTINUE
C
      CALL TXT (310,YLOC,PARM,6,0)
      CALL NUMB(360,YLOC,PARVAL(N,1),6H(F6.2),0)
      YLOC=YLOC-15
  300 CONTINUE
```

```
C
      YLOC=1010
C
      DO 400  N=1,NSETS
C
         DO 350  I=1,3
         NAME(I)=IDSETS(I,N)
  350    CONTINUE
C
      CALL TXT (610,YLOC,NAME,6,0)
      YLOC=YLOC-15
  400 CONTINUE
C
      CALL TXT (400,800,WLABEL,8,0)
      CALL NUMB(445,800,CPAR,6H(F5.2),0)
C
  450 CALL ENDFRM
C
      RETURN
C
C****************FORMATS*********************
C
 5001 FORMAT(/22H$USE DEFAULT SCALING? )
 5002 FORMAT(A1)
 5003 FORMAT(/30H$USE SAME SCALE AS LAST PLOT? )
 5004 FORMAT(/36H ENTER MINIMUM AND MAXIMUM VALUES OF
     *       /45H$INDEPENDENT VARIABLE (SEPARATED BY COMMA) - )
 5005 FORMAT(2F10.0)
 5006 FORMAT(/36H ENTER MINIMUM AND MAXIMUM VALUES OF
     *       /43H$DEPENDENT VARIABLE (SEPARATED BY COMMA) - )
C
C********************************************
C
      END
```

```
      SUBROUTINE SORTOR
C
      COMMON /CURVS/ X(20,5),Y(20,5),NP(5),AVAL(5),CPAR
      COMMON /COUNT/ MODE,NCURV,NSETS,NAPARM,NFEE,MODP,NPT,SAMSET
C
      NPM=NFEE-1
C
      DO 300  I=1,NPM
      JJ=I+1
         DO 200  J=JJ,NFEE
         IF (X(I,1).LT.X(J,1)) GO TO 200
         TEMP=X(J,1)
         X(J,1)=X(I,1)
         X(I,1)=TEMP
            DO 100  K=1,NCURV
            TEMP=Y(J,K)
            Y(J,K)=Y(I,K)
            Y(I,K)=TEMP
  100       CONTINUE
  200    CONTINUE
  300 CONTINUE
C
      DO 400  J=2,NCURV
      DO 400  I=1,NFEE
      X(I,J)=X(I,1)
  400 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE SUMMRY
C
C           THIS ROUTINE PRODUCES INFORMATION ABOUT THE DATASET
C           CURRENTLY BEING SEARCHED
C
      IMPLICIT INTEGER (A-Z)
C
      COMMON /FILE3/ NREC3,NEXREC
      COMMON /NDEX/ INDEX(14,100),DSLIST(100,3),DSNUM,SETCNT
C
      DIMENSION NAME(3),DATE(6),BODY(24),PARMS(10,3),DV(3)
C
      REAL PARVAL(10),MOX(20),MIN,MAX
C
      NREC3=INDEX(1,DSNUM)
      READ (3'NREC3) NAME,DATE,BODY,NMN,MAXNAA,NDV,NPAR,
     *               ((PARMS(I,J),J=1,3),I=1,NPAR),(PARVAL(I),I=1,NPAR)
      WRITE(5,5001) NAME,DATE,BODY
C
      DO 10  I=1,NPAR
   10 WRITE(5,5002) (PARMS(I,J),J=1,3),PARVAL(I)
```

```
C
      WRITE(5,5003)
C
      NREC3=INDEX(2,DSNUM)+1
C
      DO 30  I=1,NMN
      READ (3'NREC3) (MOX(L),L=1,MAXNAA)
      MIN=MOX(1)
      MAX=MOX(1)
C
          DO 20  J=2,MAXNAA
          MIN=AMIN1(MIN,MOX(J))
          MAX=AMAX1(MAX,MOX(J))
   20     CONTINUE
C
      WRITE(5,5004) I,MIN,MAX
   30 CONTINUE
C
      WRITE(5,5005)
C
      DO 40  I=1,NDV
      NREC3=INDEX(I+4,DSNUM)
      READ (3'NREC3) DV
      WRITE(5,5006) DV
   40 CONTINUE
C
      WRITE(5,5007)
C
      RETURN
C
C*************************FORMATS************************
C
 5001 FORMAT(//13X,11HSUMMARY OF ,3A2/1X,45(1H*)/1X,1H*,
     *         32X,8HCREATED ,6A2/1X,1H*,5X,16HCONFIGURATION - ,
     *         24A2/1X,1H*/1X,1H*,14X,10HPARAMETERS/1X,1H*,
     *         13X,12(1H-))
 5002 FORMAT(1X,1H*,10X,3A2,3H = ,F6.2)
 5003 FORMAT(1X,1H*/1X,1H*,13X,12HMACH NUMBERS/1X,1H*,13X,
     *         12(1H-)/1X,1H*,6X,5HINDEX,10X,5HRANGE)
 5004 FORMAT(1X,1H*,6X,I2,6X,5HFROM ,F6.3,4H TO ,F6.3)
 5005 FORMAT(1X,1H*/1X,1H*,10X,19HDEPENDENT VARIABLES/1X,1H*,9X,21(1H-))
 5006 FORMAT(1X,1H*,17X,3A2)
 5007 FORMAT(1X,1H*/1X,1H*/1X,45(1H*))
C
C********************************************************
C
      END
```

```
      SUBROUTINE TABULA
C
      IMPLICIT INTEGER (A-Z)
C
      COMMON /CURVS/ X(20,5),Y(20,5),NP(5),AVAL(5),CPAR
      COMMON /TEXT/  IDCURV(6,5),IDSETS(3,10),IDCFIG(24,5),
     *               IDPARM(3,10,5),NPARMS(5),PARVAL(10,5)
      COMMON /COUNT/ MODE,NCURV,NSETS,NAPARM,NFEE,MODP,NPT,FLAG,
     *               IVFLAG,ACFLAG,MACP
      COMMON /IDENTS/ SETID(3),VAR(2,3),IVID(3),DVID(3),ACPARM(5),
     *                NAMES(10,3)
C
      REAL X,Y,AVAL,CPAR,PARVAL
C
      DIMENSION FEE(3)
C
      DATA FEE /2H P,2HHI,2H  /
C
      WRITE(5,5001)
C
      IF (MODE.EQ.3) GO TO 450
C
C
C
C
      DO 300  N=1,NSETS
      WRITE(5,5003) (IDSETS(I,N),I=1,3),(IDCFIG(I,N),I=1,24)
      WRITE(5,5004)
         DO 250  M=1,NPARMS(N)
         WRITE(5,5005) (IDPARMS(I,M,N),I=1,3),PARVAL(M,N)
  250    CONTINUE
  300 CONTINUE
C
      WRITE(5,5008)
C
      DO 400  N=1,NCURV
      WRITE(5,5006) (VAR(IVFLAG,J),J=1,3),(IDCURV(I,N),I=1,3),
     *              (VAR(ACFLAG,J),J=1,3),AVAL(N),(IDCURV(I,N),I=4,6)
         DO 350  M=1,NP(N)
         WRITE(5,5007) X(M,N),Y(M,N)
  350    CONTINUE
  400 CONTINUE
C
      WRITE(5,5008)
C
      RETURN
C
C
C
  450 CONTINUE
      WRITE(5,5011)
      DO 500  N=1,NSETS
      WRITE(5,5002) (IDSETS(I,N),I=1,3)
  500 CONTINUE
C
      WRITE(5,5012) (IDCFIG(I,1),I=1,24)
```

```
C
      MACP=2
      IF (MODP.EQ.2) MACP=1
C
C
C
C
      WRITE(5,5004)
      DO 750  M=1,NPARMS(1)
      WRITE(5,5005) (IDPARMS(I,M,1),I=1,3),PARVAL(M,1)
  750 CONTINUE
C
      WRITE(5,5008)
      WRITE(5,5009) (VAR(MODP,J),J=1,3),CPAR
      DO 850  N=1,NCURV
      WRITE(5,5010) FEE,(IDCURV(I,N),I=1,3),(VAR(MACP,J),J=1,3),AVAL(N)
         DO 800  M=1,NFEE
         WRITE(5,5007) X(M,N),Y(M,N)
  800    CONTINUE
  850 CONTINUE
C
      WRITE(5,5008)
C
      RETURN
C
C******************FORMATS***********************
C
 5001 FORMAT(////1X,16(1H$),28H TABULATION OF CURRENT FRAME )
 5002 FORMAT(1X,1H$,16X,3A2)
 5003 FORMAT(/1X,1H$/1X,1H$/1X,1H$,4X,10HDATASET - ,3A2/1X,1H$,
     *       4X,16HCONFIGURATION - ,24A2)
 5004 FORMAT(1X,1H$,17X,17HPARAMETRIC VALUES/
     *       1X,1H$,17X,17H-----------------)
 5005 FORMAT(1X,1H$,17X,3A2,4X,F6.2)
 5006 FORMAT(/1X,1H$/1X,1H$/1X,1H$,6X,3A2,8X,3A2,2X,4HFOR ,3A2,
     *       1H=,F6.2,6H FROM ,3A2)
 5007 FORMAT(1X,1H$,2X,F10.4,3X,F10.4)
 5008 FORMAT(1X,1H$/1X,43(1H$))
 5009 FORMAT(/20X,3A2,1H=,F5.2/)
 5010 FORMAT(1X,1H$/1X,1H$/1X,1H$,6X,3A2,8X,3A2,2X,4HFOR ,3A2,
     *       1H=,F6.2)
 5011 FORMAT(1X,1H$/1X,1H$,15H USING DATASETS )
 5012 FORMAT(2H $/19H $ CONFIGURATION - ,24A2/2H $)
C
C**********************************************
C
      END
>
```

REFERENCES

1. Burt, J. R., Jr.: An Experimental Investigation of the Aerodynamic Characteristics of Several Nose-Mounted Canard Configurations at Transonic Mach Numbers. U.S. Army Missile Command, Redstone Arsenal, Alabama, Technical Report RD-75-2, August 1974.

2. Henderson, J. H.: An Investigation of Jet Plume Effects on the Stability Characteristics of a Body of Revolution in Conjunction with Fins of Various Geometry and Longitudinal Positions at Transonic Speeds (Sting Mounted Model with Normal Jet Plume Simulation). U.S. Army Missile Command, Redstone Arsenal, Alabama, Technical Report RD-75-37, June 1975.

3. Burt, J. R., Jr.: An Experimental Investigation of the Aerodynamic Characteristics of Several Nose-Mounted Canard Configurations at Supersonic Mach Numbers. U.S. Army Missile Command, Redstone Arsenal, Alabama, Technical Report RD-75-17, January 1975.

4. Burt, J. R., Jr.: An Experimental Investigation of the Aerodynamic Characteristics of Nose Mounted Canard Configurations at Supersonic Mach Numbers (1.5 through 4.5). U.S. Army Missile Command, Redstone Arsenal, Alabama, Technical Report RD-77-5, October 1976.

5. Henderson, J. H.: Investigation of Jet Plume Effects on the Longitudinal Stability Characteristics of a Body of Revolution with Various Fin Configurations at Mach Numbers from 0.2 to 2.3 (Normal Jet Plume Simulator). U.S. Army Missile Command, Redstone Arsenal, Alabama, Technical Report RD-76-22, February 1976.

6. Henderson, J. H., Dahlke, C. W., and Batiuk, G.: An Experimental Investigation using a Normal Jet Plume Simulator to Determine Jet Plume Effects on a Long Slender Rocket Configuration at Mach Numbers from 0.2 to 1.5. U.S. Army Missile Command, Redstone Arsenal, Alabama, Technical Report TD-77-2, February 1977.

7. Schwind, R. G.: High Angle Canard Missile Test in the Ames 11-Foot Transonic Wind Tunnel. NASA CR 2993, June 1978.

8. Hemsch, M. J. and Nielsen, J. N.: Test Report for Canard Missile Tests in Ames 6- by 6-Foot Supersonic Wind Tunnel. NEAR TR 72, August 1974.

9. Smith, C. A. and Nielsen, J. N.: Prediction of Aerodynamic Characteristics of Cruciform Missiles to High Angles of Attack Utilizing a Distributed Wake. NEAR TR 208, January 1980.

10. Mullen, J., Jr. and Schwartz, A. M.: PLOTA/PLOTI User's Guide. NEAR TR 144, August 1977.

11. Schwartz, A. M. and Black, C. A.: Simple Graphics Package Host Routines, Machine Portable Graphics. NEAR TR 145, August 1977.

TABLE 1. MISSILE TESTS INCLUDED IN DATA BASE

TEST NUMBER	TITLE	REFERENCE
CALSPAN T17-093	Canard Control	1
CALSPAN T17-123	Plume Effects	2
ARC 66-036	Canard Control	3 and 8
AEDC V41A-C1A	Canard Control	4
AEDC SF172/TF360	Plume Effects	5
AEDC TF-416	Plume Effects	6
ARC T183-11	High-Angle-of-Attack	7

TABLE 2. FORMAT AND NOMENCLATURE FOR DATAMAN TAPES

(a) Format Statements

The DATAMAN tapes are written using these FORTRAN statements.

```
    WRITE(KTAPE,100) FDSET, IDENT, DATE(1), DATE(2), (CONFIG(I),
                     I = 1,8), ITYPE, MODE, NMN, MAXNAA, NDV,
                     NPAR, NDM

100 FORMAT(12A6, 7I6)

    WRITE(KTAPE,101) (IDPVAR(I), I=1,2), (NDPVAR(I), I=1,2),
                     (NAMPAR(I), I=1, NPAR), (FILEREF(I),
                     I=1,3), (REF(I,2), I=1,7), (NAMDM(I),
                     I=1,20)

101 FORMAT(22A6)

    WRITE(KTAPE,102) ((IRUNNO(I,J), I = 1,NMN), J = 1,2),
                     (NAAX(M), M = 1,NMN)

102 FORMAT(22I6)

    WRITE(KTAPE,103) ANGMIN, ANGMAX, (XMLIM(I), I = 1,2),
                     (APLIM(I), I=1,2), ((DLIMIT(I,J), I=1,
                     NDV), J = 1,2), (PARVAL(I), I=1,NPAR),
                     (REF(I,1), I = 1,7), ((XMACH(M,I), I = 1,
                     MAXNAA), M = 1,NMN), ((ALPHA(I,M), I = 1,
                     MAXNAA), M = 1, NMN), (RNL(M), M = 1,NMN),
                     ((DM(M,I), M = 1, NMN), I = 1,20)

    DO 10 K = 1, NDV

 10 WRITE(KTAPE,103) ((D(I,M,K), I = 1, MAXNAA), M = 1, NMN),
                     (GD(M,K), M = 1, NMN)

103 FORMAT(8E15.8)
```

TABLE 2. Continued

(b) Description of DATAMAN Variable Names

Variable	Type	Dimension	Description
FDSET	A	none	Contains the characters *FDSET
IDENT	A	none	Six character label used to identify the dataset
DATE	A	2	Dataset creation date
CONFIG	A	8	Configuration of model tested
ITYPE	I	none	Data type (1 = raw data) and 2 = nominalized data). Has value 1 for all nine tapes. The dependent variable data are actually presented in coefficient form.
MODE	I	none	Data mode (1 = pitch angle varies) and 2 = yaw angle varies). Has value 1 for all nine tapes.
NMN	I	none	Number of Mach numbers tested (≤20)
MAXNAA	I	none	Maximum number of alphas per Mach number (≤50)
NDV	I	none	Number of dependent variables (≤10)
NPAR	I	none	Number of parameters (≤10). The parametric values are constant for each dataset.
NDM	I	none	Number of univariate dependent variables (≤20). Has value 0 for all nine tapes.
IDPVAR	A	2	Names of independent variables. Contains the characters MACH and ALPHA for all nine tapes.
NDPVAR	A	10	Names of dependent variables
NAMPAR	A	10	Names of parameters

TABLE 2. Concluded

Variable	Type	Dimension	Description
FILREF	A	3	Document references. Blank for all nine tapes.
REF	F,A	(7,2)	Reference data values and units.
NAMDM	A	20	Names of univariate dependent variables. Blank for all nine tapes.
IRUNNO	I	(2,10)	Run and rerun numbers associated with each Mach number.
NAAX	I	20	Number of alphas associated with each Mach number.
ANGMIN	F	none	Lower limit used in gradient computation. Set to zero on all nine tapes.
ANGMAX	F	none	Upper limit used in gradient computation. Set to zero on all nine tapes.
XMLIM	F	2	Lower and upper limits for Mach number.
APLIM	F	2	Lower and upper limits for alpha.
DLIMIT	F	(10,2)	Lower and upper limits for dependent variables.
PARVAL	F	10	Parametric values.
XMACH	F	(20,50)	Mach numbers.
ALPHA	F	(50,20)	Angles of attack.
RNL	F	20	Reynolds numbers.
DM	F	(20,20)	Univariate dependent variable data. Set to zero on all nine tapes.
D	F	(20,50, 10)	Dependent variable data.
GD	F	(20,10)	Gradient dependent variable data. Set to zero on all nine tapes.

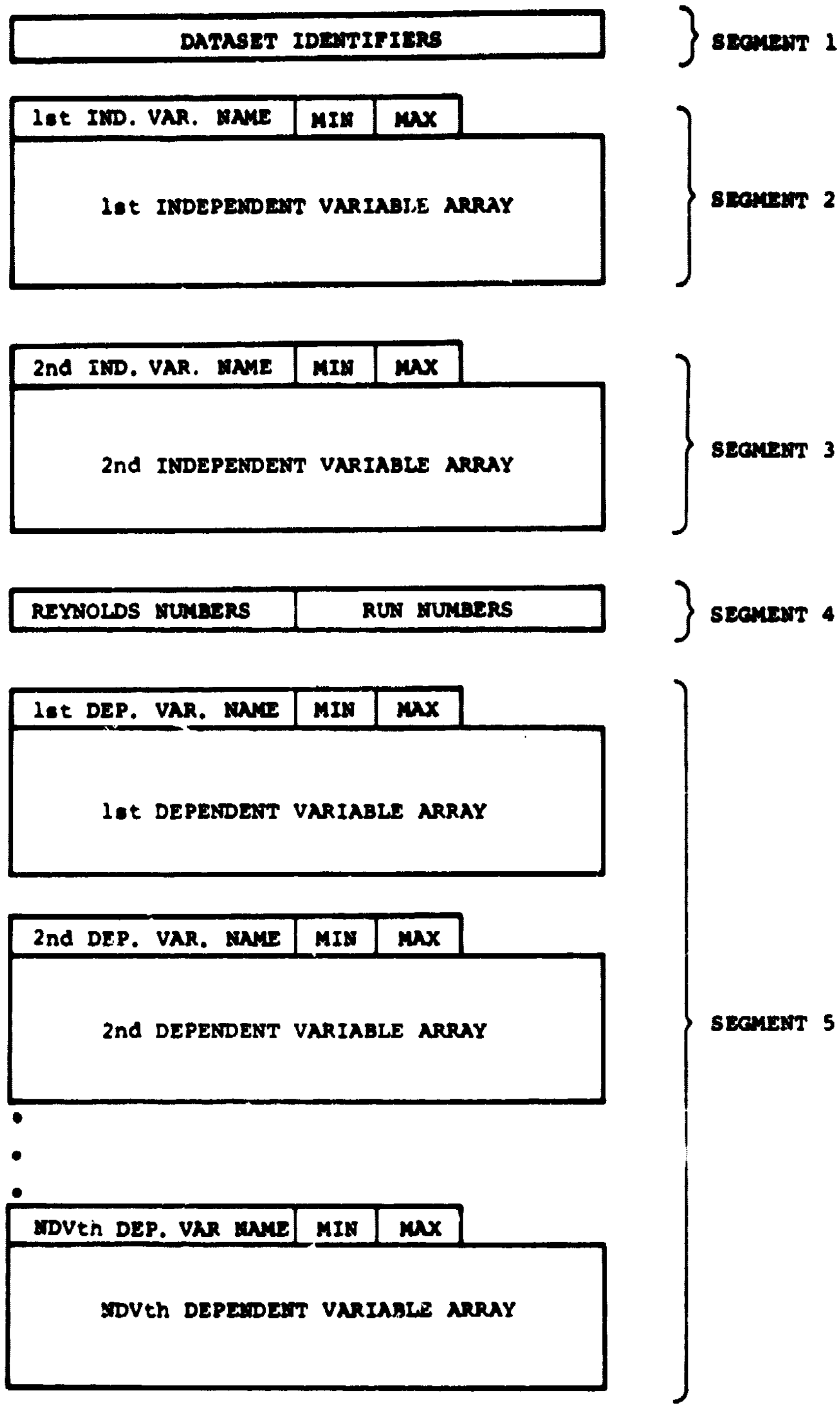
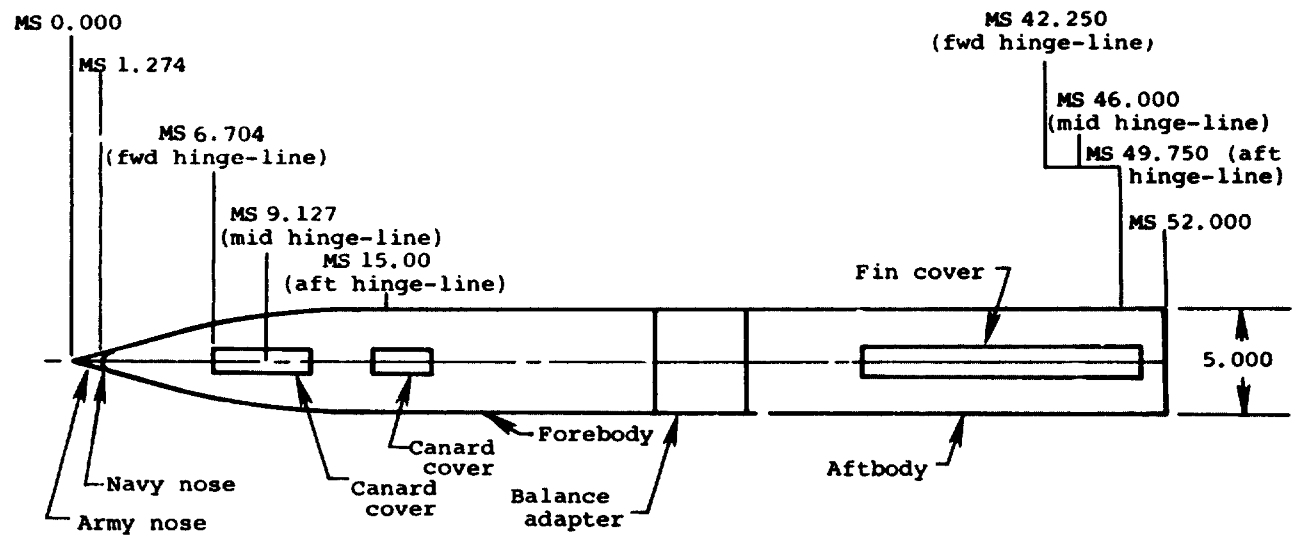


Figure 1. - Structure of DATAMANIA random access file.

```
RE2048 08 OCT 74    CONFIGURATION  6 (BW1C1T1)          AMES 66-036  2  7  9  9   } Segment 1
DETA   D1     D3     D2     D4     D1-3   D2-4   PHI-C  PHI-T
  0.00   0.00   0.00  -3.00  -3.00   0.00  -3.00   0.00 -45.00

MACH       1.500   1.995                                                          } Segment 2
   1.500   1.502   1.500   1.500   1.500   1.500   1.500
   1.995   1.995   1.995   1.995   1.995   1.995   1.995

ALPHA   -3.179  12.520                                                            } Segment 3
 7  -2.032   0.257   1.209   3.315   6.333   9.443  12.520
 7  -3.179  -0.074   0.980   3.066   6.084   9.133  12.210

   2.40    2.40 |  78   75    0    0                                              } Segment 4

CN        -0.444   2.018                                                          } Segment 5
  -0.444  -0.009   0.138   0.421   0.878   1.427   2.018
  -0.373   0.007   0.134   0.405   0.822   1.339   1.978

CA         0.359   0.460
   0.419   0.412   0.413   0.417   0.432   0.452   0.460
   0.365   0.360   0.359   0.363   0.376   0.386   0.382

CM        -2.326   0.257
   0.140  -0.240  -0.382  -0.679  -1.119  -1.742  -2.326
   0.257   0.066  -0.013  -0.155  -0.427  -0.765  -1.008

CY        -0.038   0.007
  -0.038  -0.037  -0.032  -0.028  -0.014  -0.003   0.007
  -0.024  -0.027  -0.023  -0.019  -0.018  -0.012  -0.015

CYN       -0.004   0.090
   0.090   0.075   0.064   0.055   0.029   0.011  -0.004
   0.037   0.044   0.040   0.037   0.033   0.025   0.027

CRM       -0.015   0.000
  -0.005  -0.004  -0.005  -0.007  -0.010  -0.012  -0.015
   0.000  -0.004  -0.005  -0.006  -0.007  -0.007  -0.008

CRMB      -0.009   0.006
   0.006   0.004   0.006   0.003   0.000  -0.001  -0.009
   0.005   0.004   0.004   0.002   0.000   0.003  -0.003

CYB       -0.010   0.010
  -0.010  -0.007  -0.003  -0.009  -0.003   0.004  -0.002
  -0.001   0.002   0.002   0.002   0.008   0.010   0.009

CNC       -0.074   0.124
  -0.074  -0.032  -0.017   0.009   0.049   0.089   0.124
  -0.047  -0.017  -0.009   0.012   0.039   0.062   0.085
```

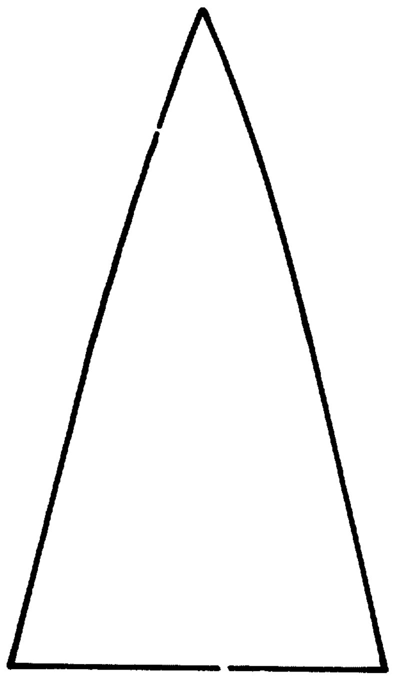
Figure 2. - Example of DATAMANIA random access dataset.

All dimensions in inches

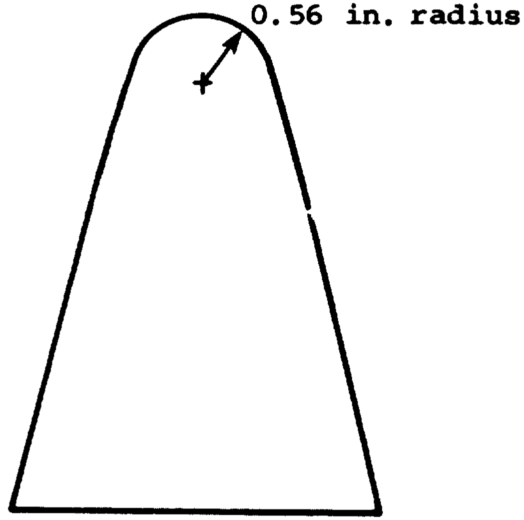


(a) Sketch of model body showing hinge-line positions for canard and tail fins.

Figure 3.- Dimensions of Army Generalized Missile Model.



(b) Army nose.

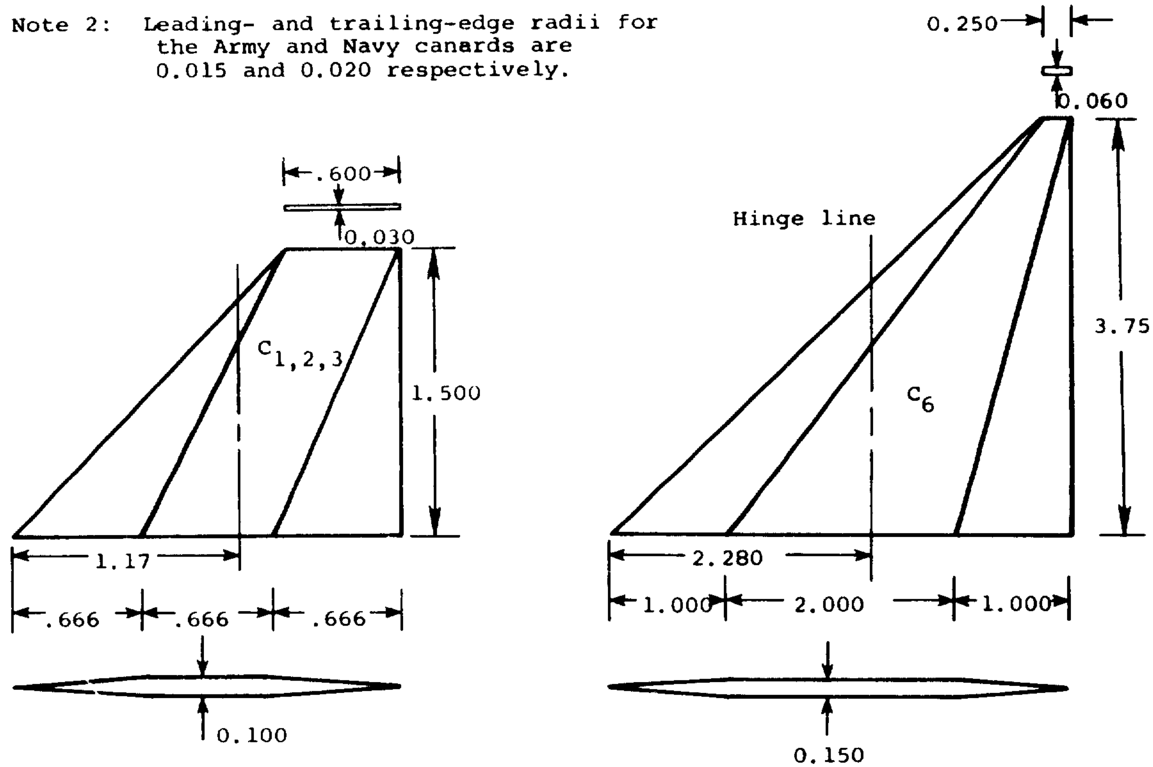


(c) Navy nose.

Figure 3.- Continued.

Note 1: All dimensions are in inches.

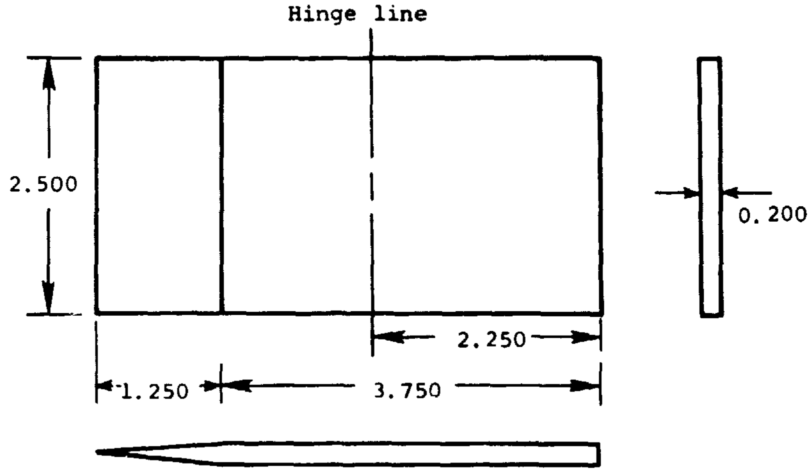
Note 2: Leading- and trailing-edge radii for the Army and Navy canards are 0.015 and 0.020 respectively.



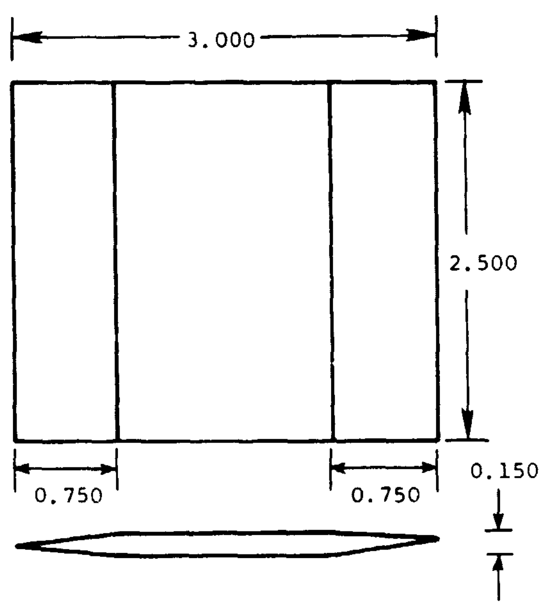
(d) Army canards.

(e) Navy canards.

Figure 3 .- Continued.

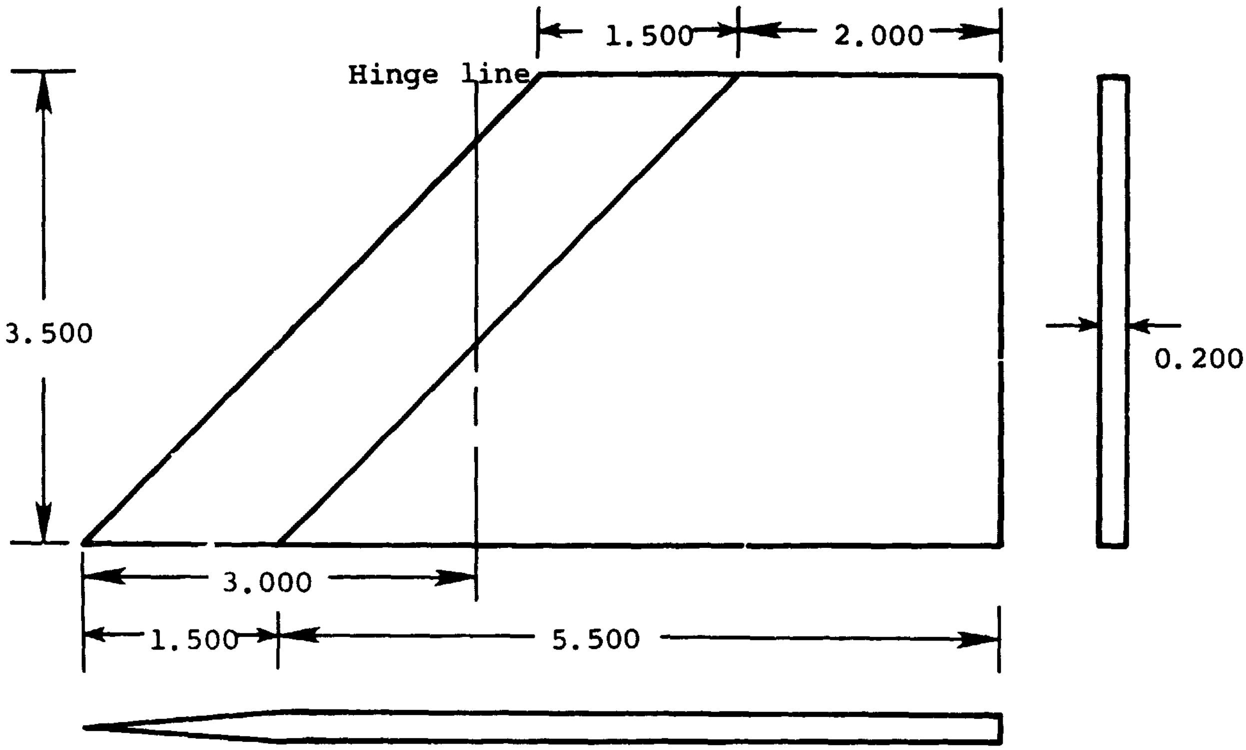


(f) Army F1 tail fins.



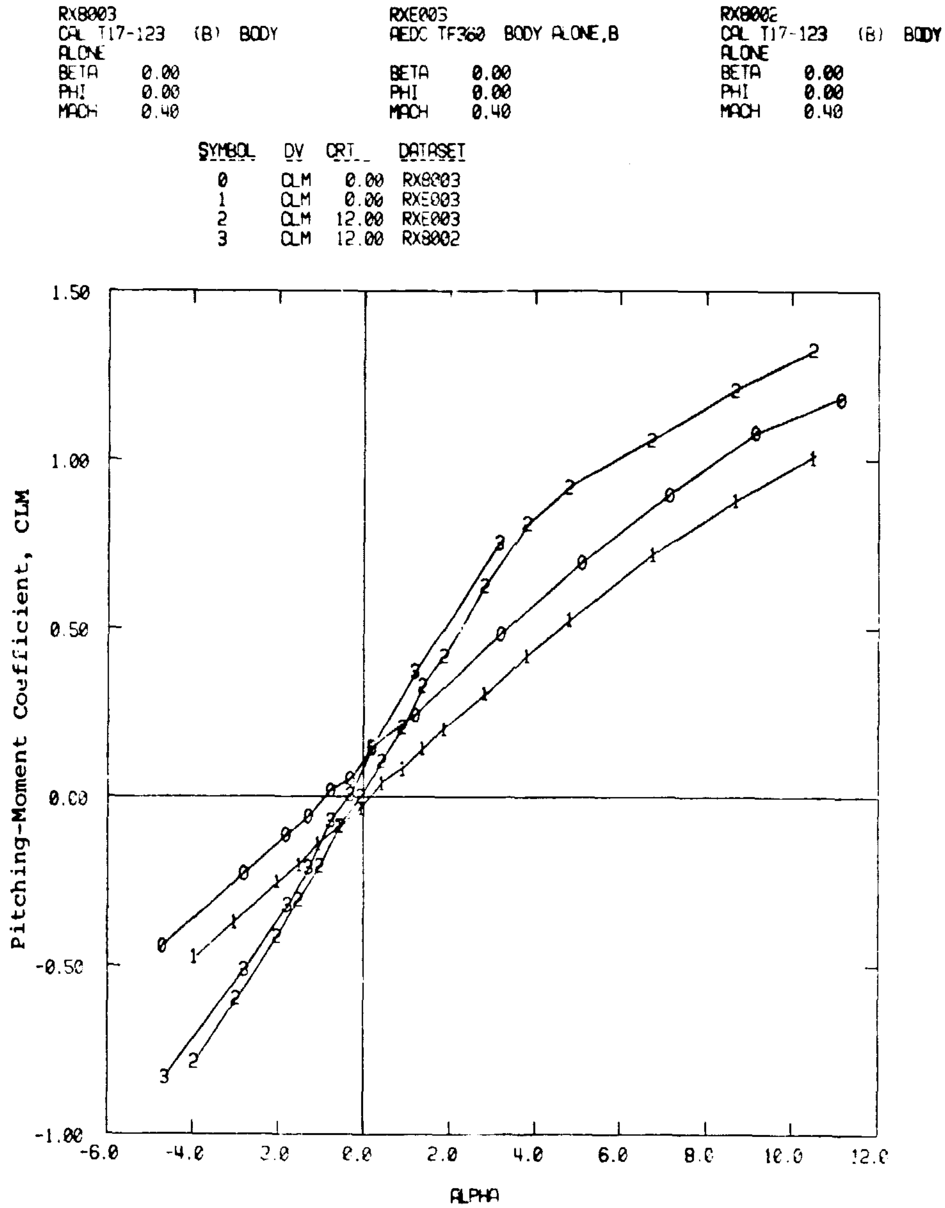
(g) Army F2 tail fins.

Figure 3 .- Continued.



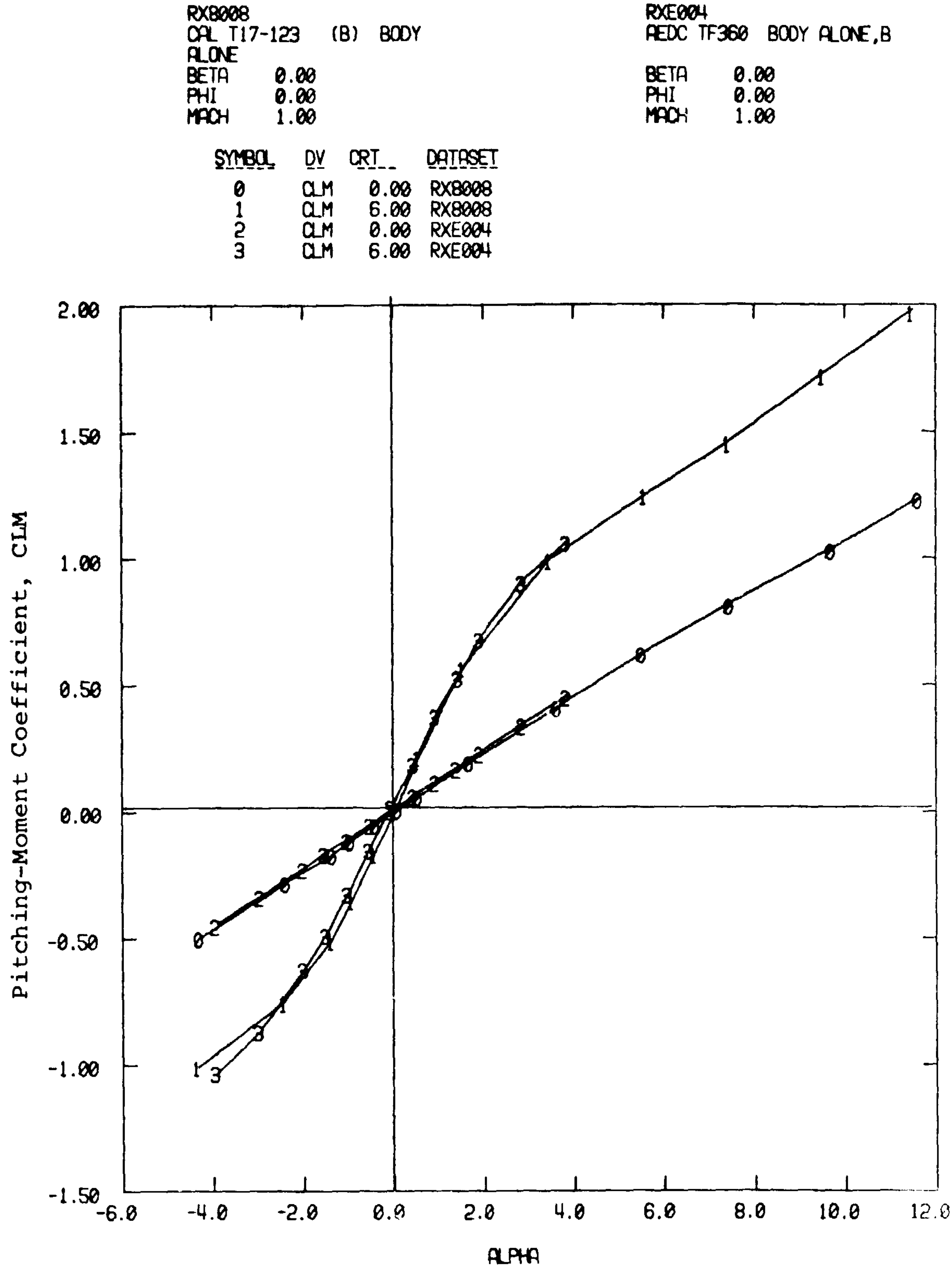
(h) Navy tail fins.

Figure 3 .- Concluded.



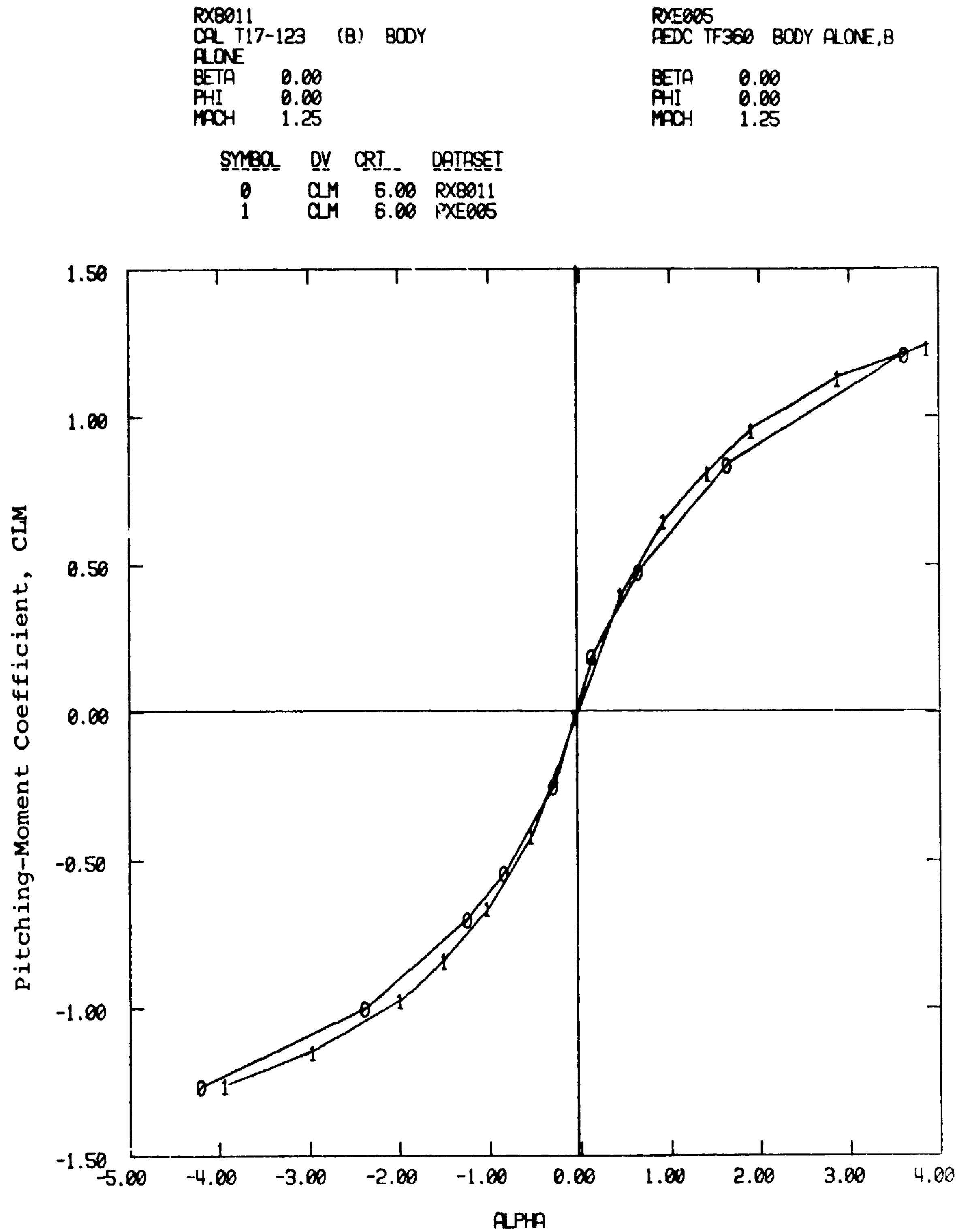
(a) $M_\infty = 0.4$; CRT = 0, 12.

Figure 4.- Data comparisons for plume interaction tests in Calspan and AEDC wind tunnels; body-alone.



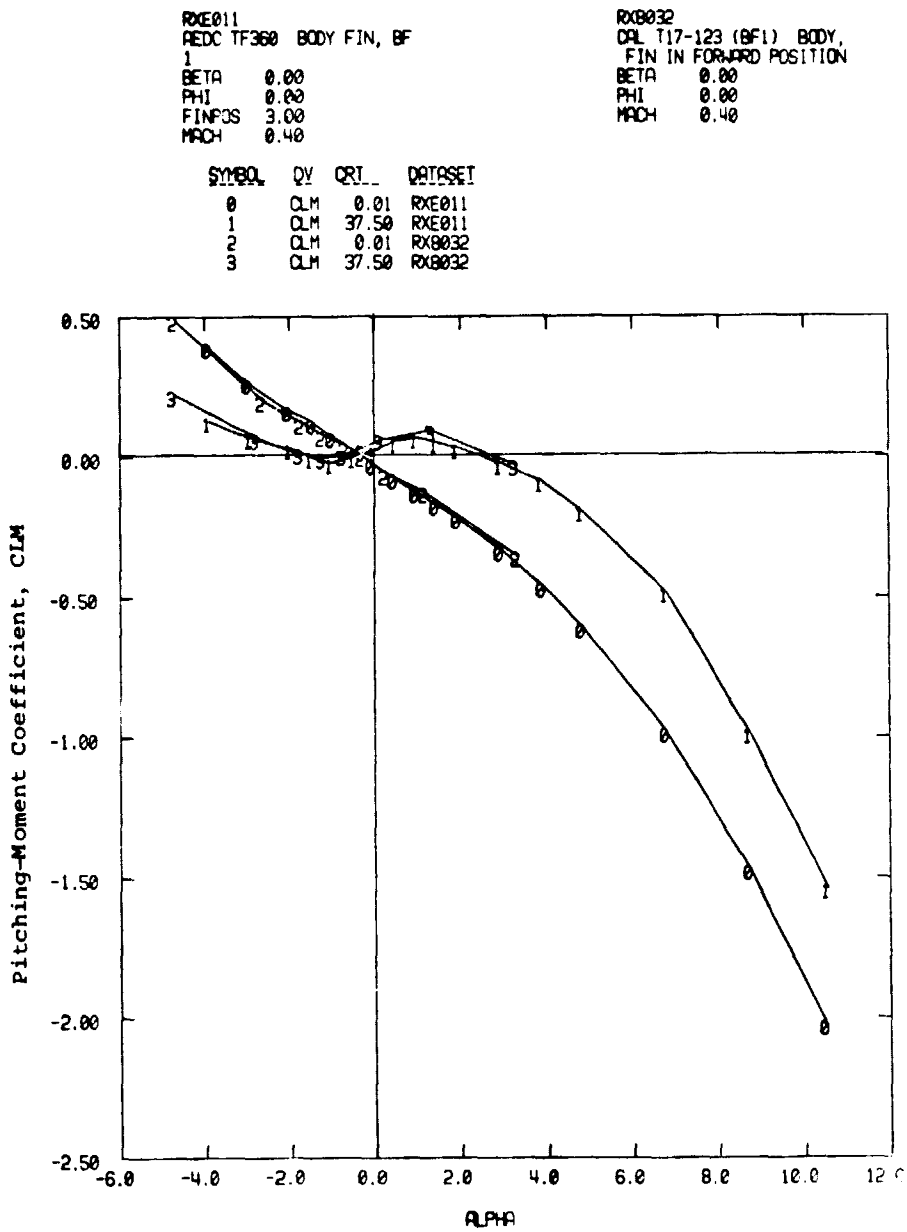
(b) M_∞ = 1.0; CRT = 0,6.

Figure 4.- Continued.



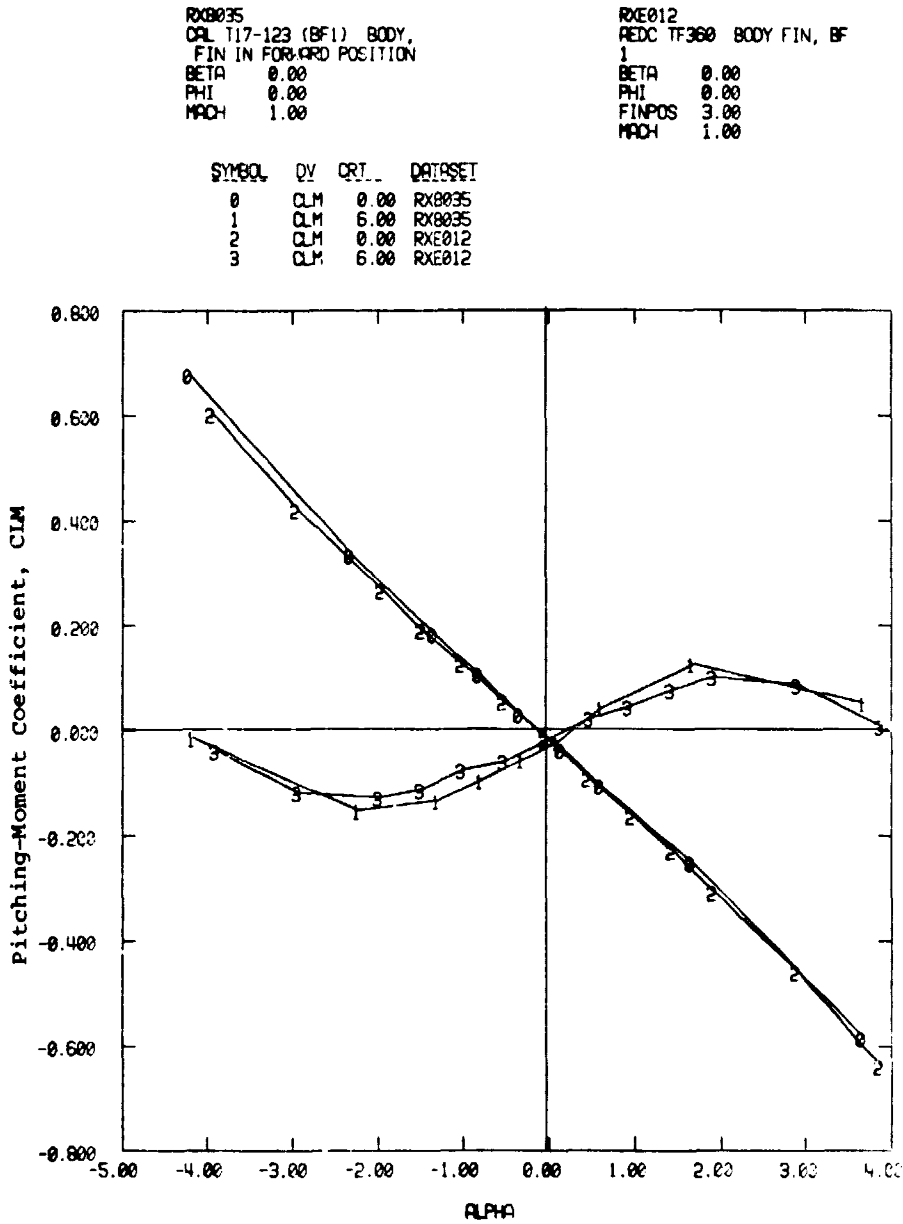
(c) $M_\infty = 1.25$; CRT = 6.

Figure 4.- Concluded.



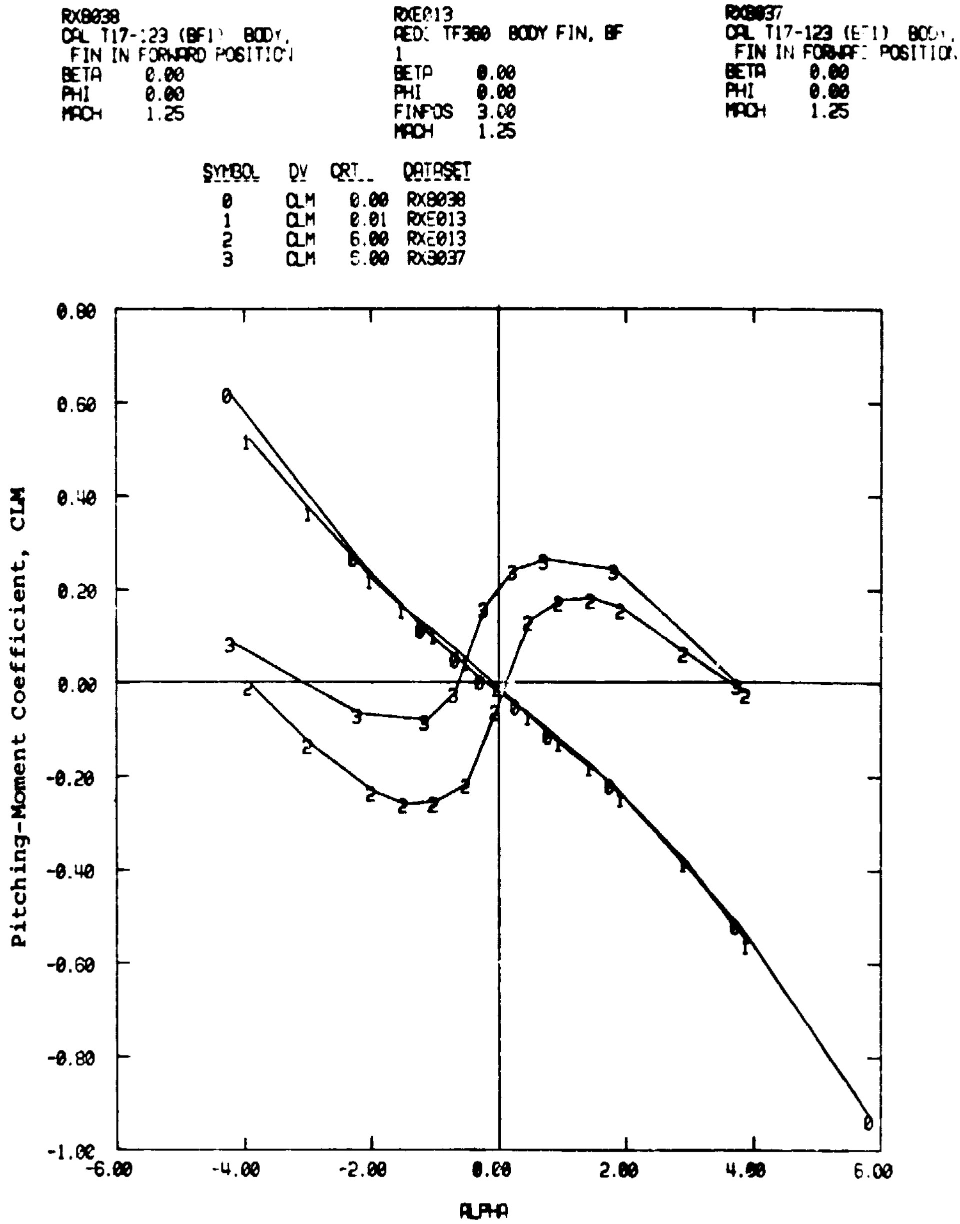
(a) $M_\infty = 0.4$ CRT = 0.01, 37.5.

Figure 5.- Data comparisons for plume interaction tests in Calspan and AEDC wind tunnels; body with Army F1 fins.



(b) M_∞ = 1.0, CRT = 0, 6.

Figure 5.- Continued.



(c) M_∞ = 1.25; CRT = 0, 6.

Figure 5.- Concluded.

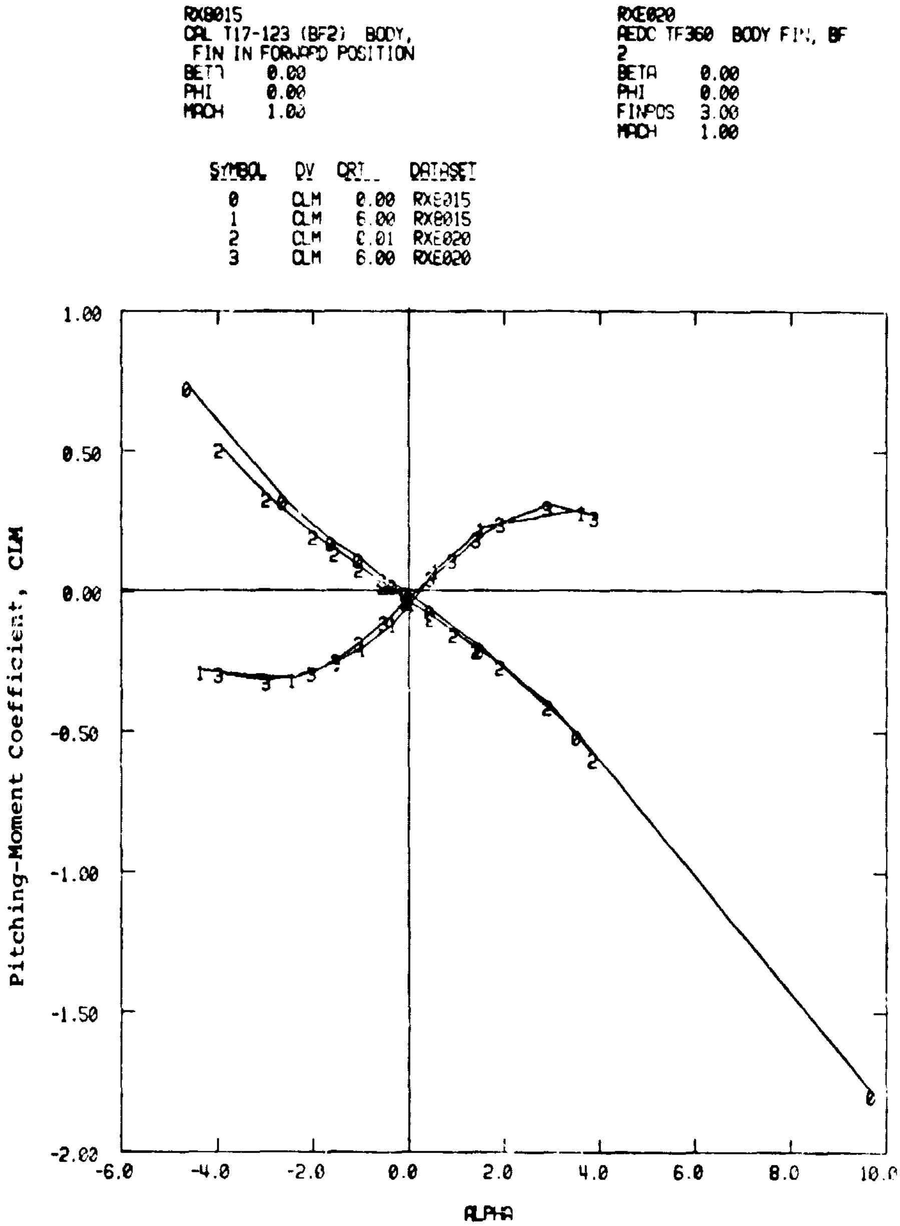


Figure 6.- Data comparisons for plume interaction tests in Calspan and AEDC wind tunnels; body with Army F2 fins.

```
AUE008
CALSPAN CANARD CONTROL,
 B1N1C2T2
BETA     0.00
PHICND   0.00
PHITAL   0.00
CNDPOS   3.00
TALPOS   1.00
DCND1    0.00
DCND2    0.00
DCND3    0.00
DCND4    0.00
```

```
RE2036
CONFIGURATION  6 (BN1C1T
1)          AMES 66-03
BETA     0.00
D1       0.00
D3       0.00
D2       0.00
D4       0.00
D1-3     0.00
D2-4     0.00
PHI-C    0.00
PHI-T    0.00
```

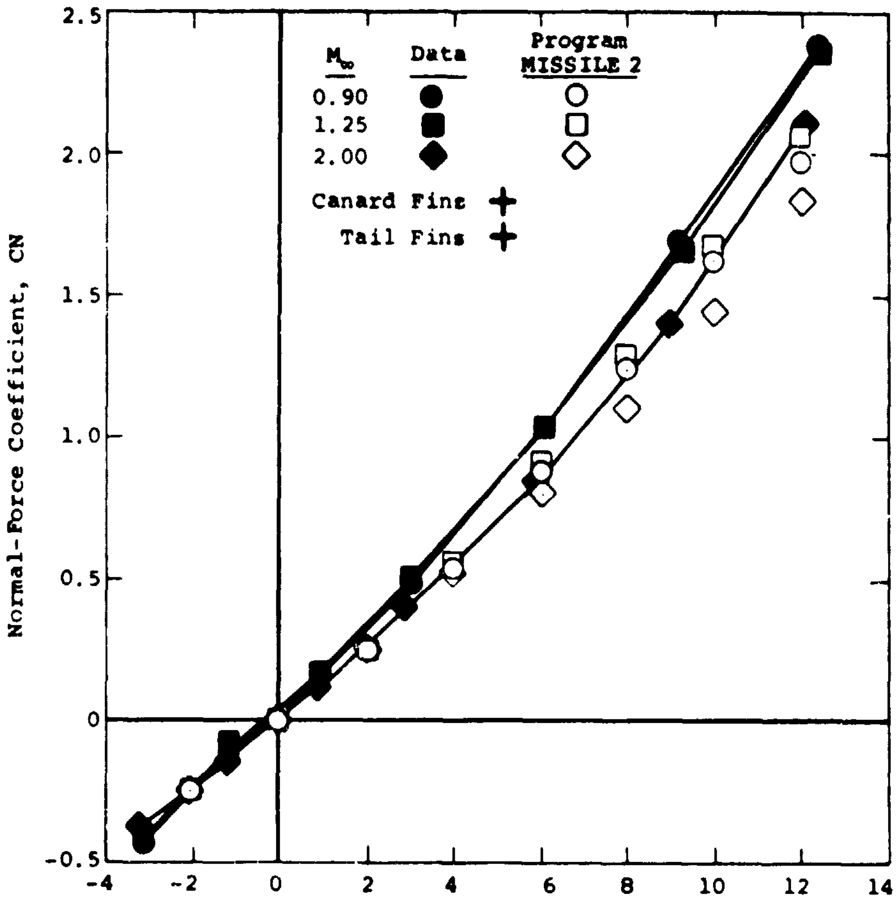
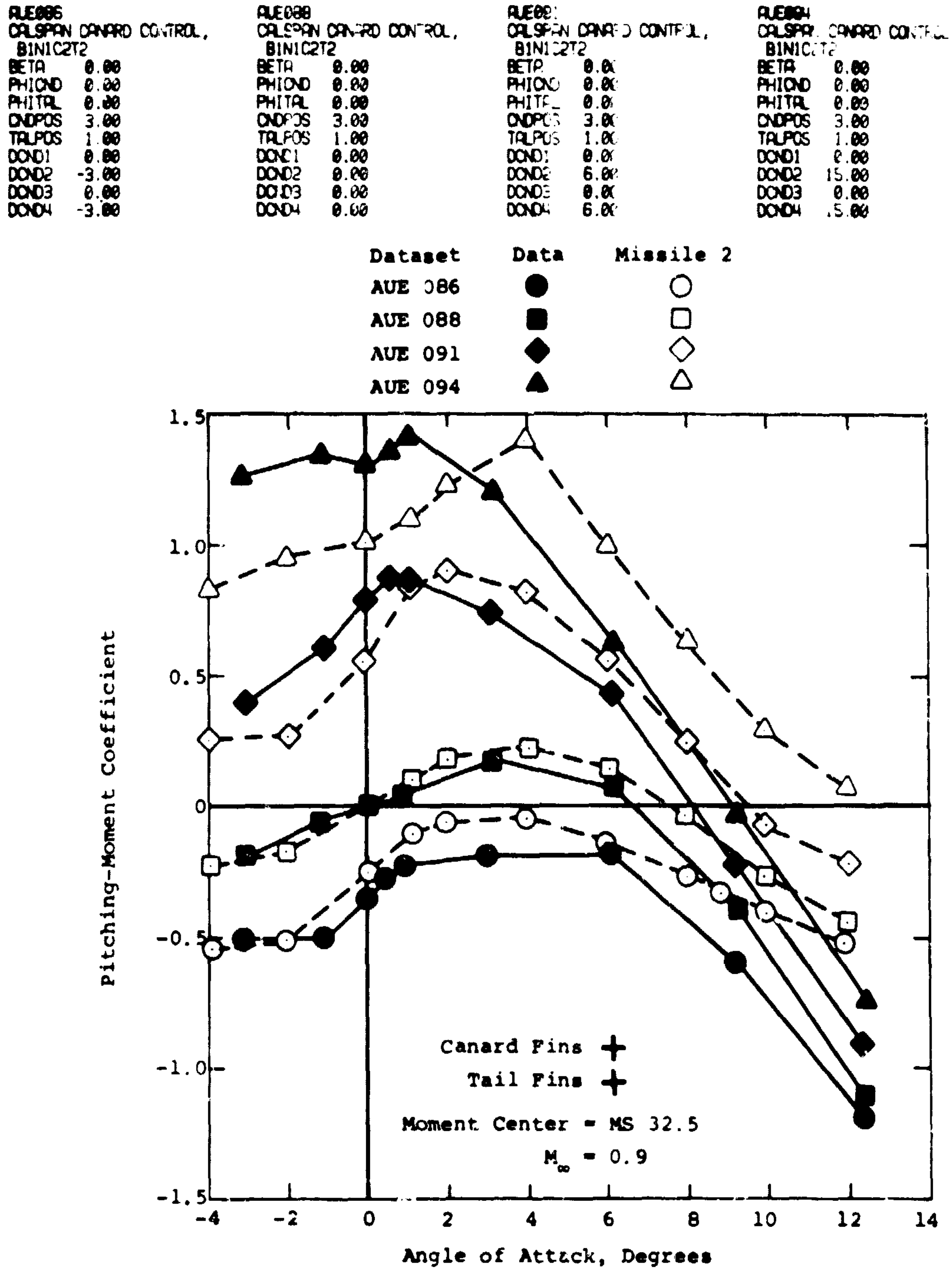
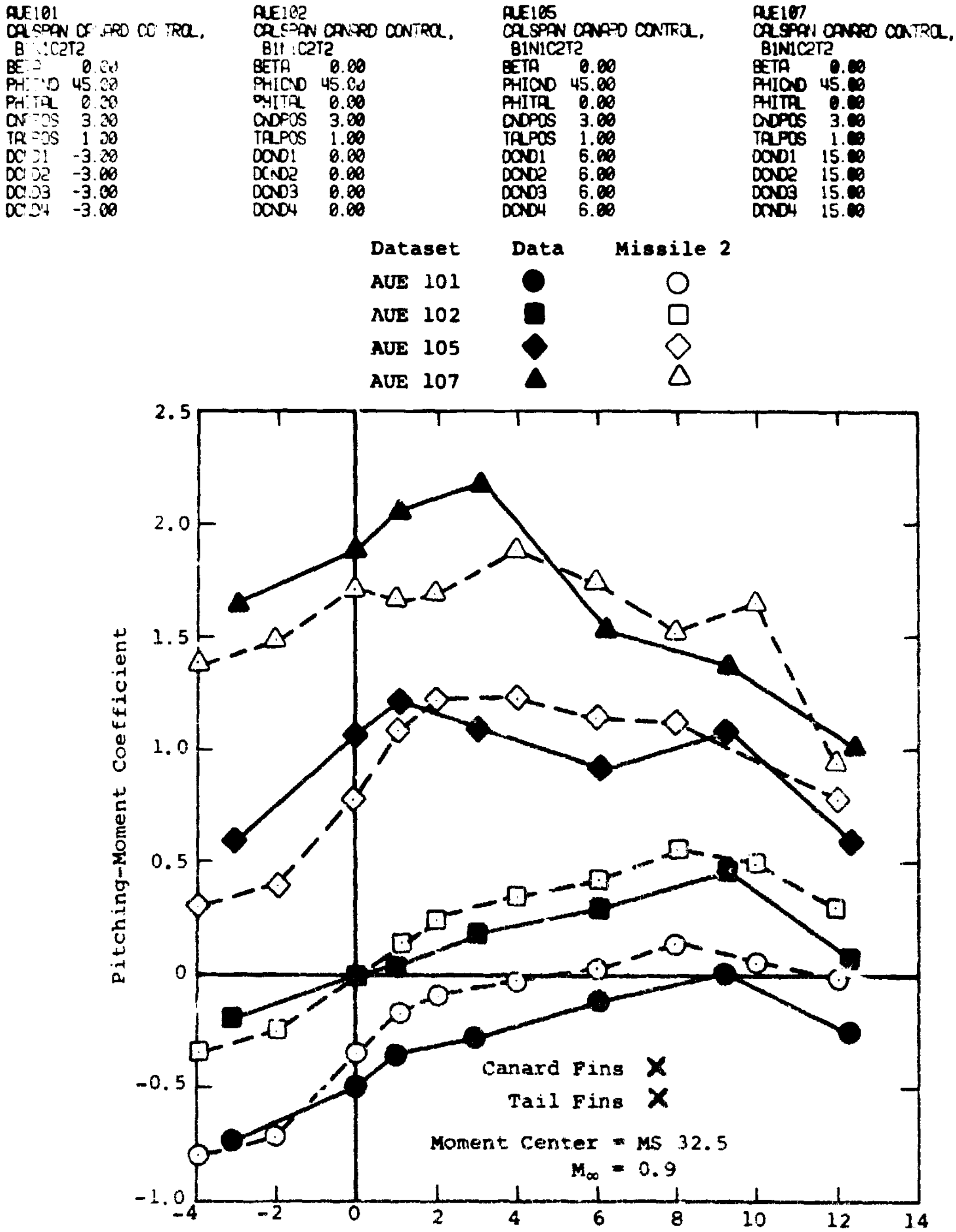


Figure 7.- Comparison of data and predictions from Program MISSILE 2 for Burt's Canard-Configuration tests; normal-force coefficient, $\phi = 0$, inline tail fins.



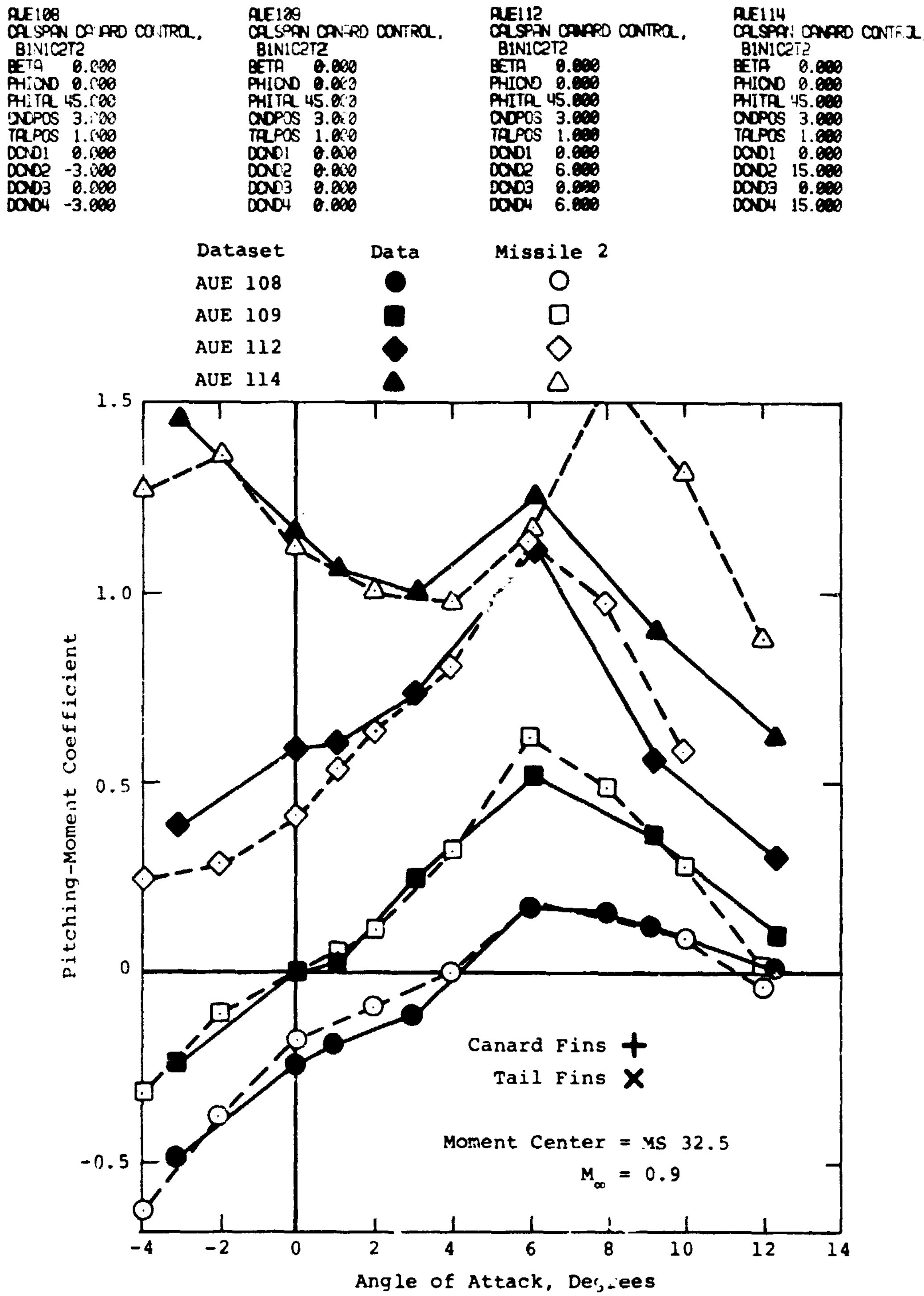
(a) ϕ = 0; inline tail fins.

Figure 8.- Comparison of pitching-moment coefficient data and predictions from Program MISSILE 2 for Burt's Canard-Configuration tests; M_∞ = 0.9.



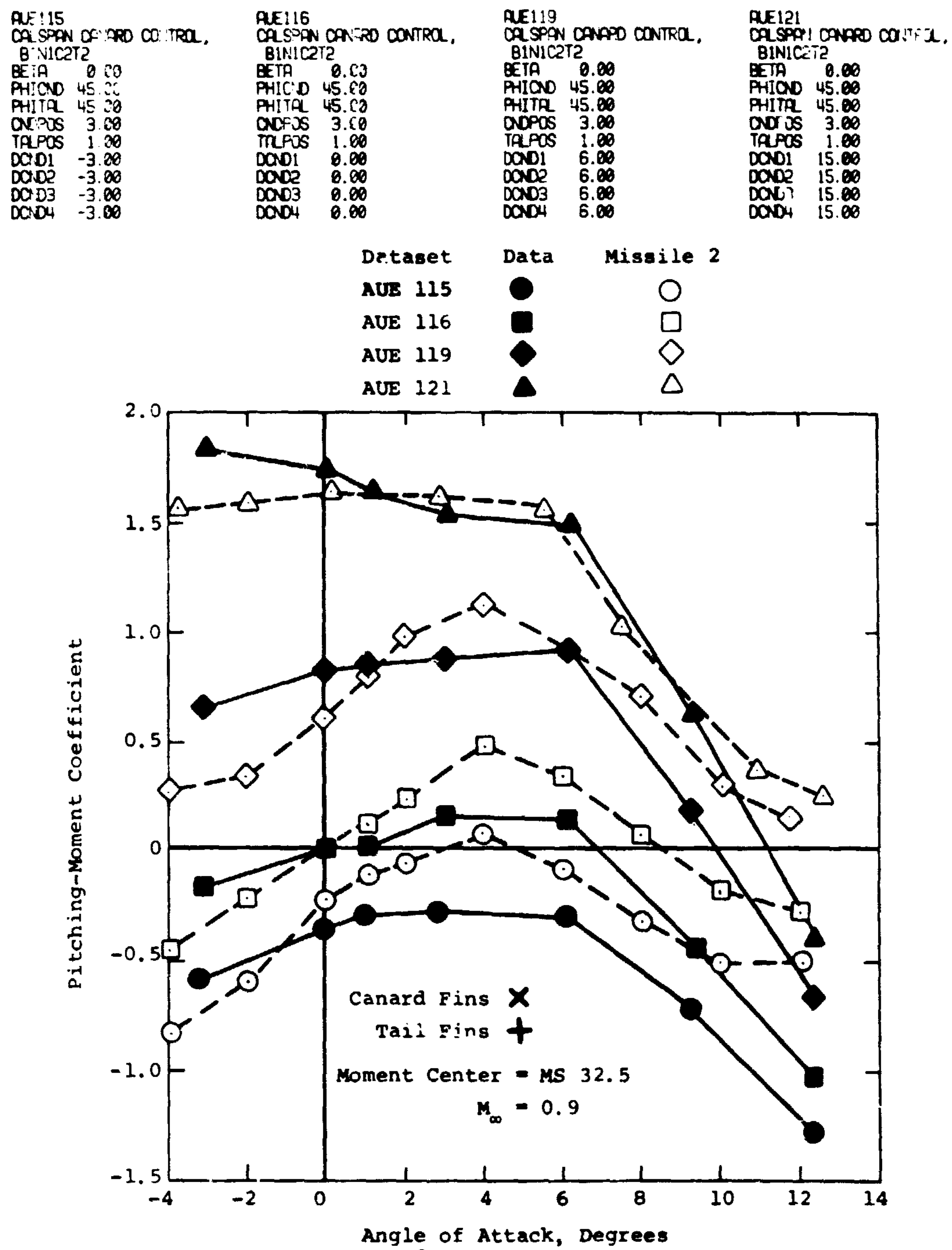
(b) $\phi = 45^{\circ}$; inline tail fins.

Figure 8.- Continued.



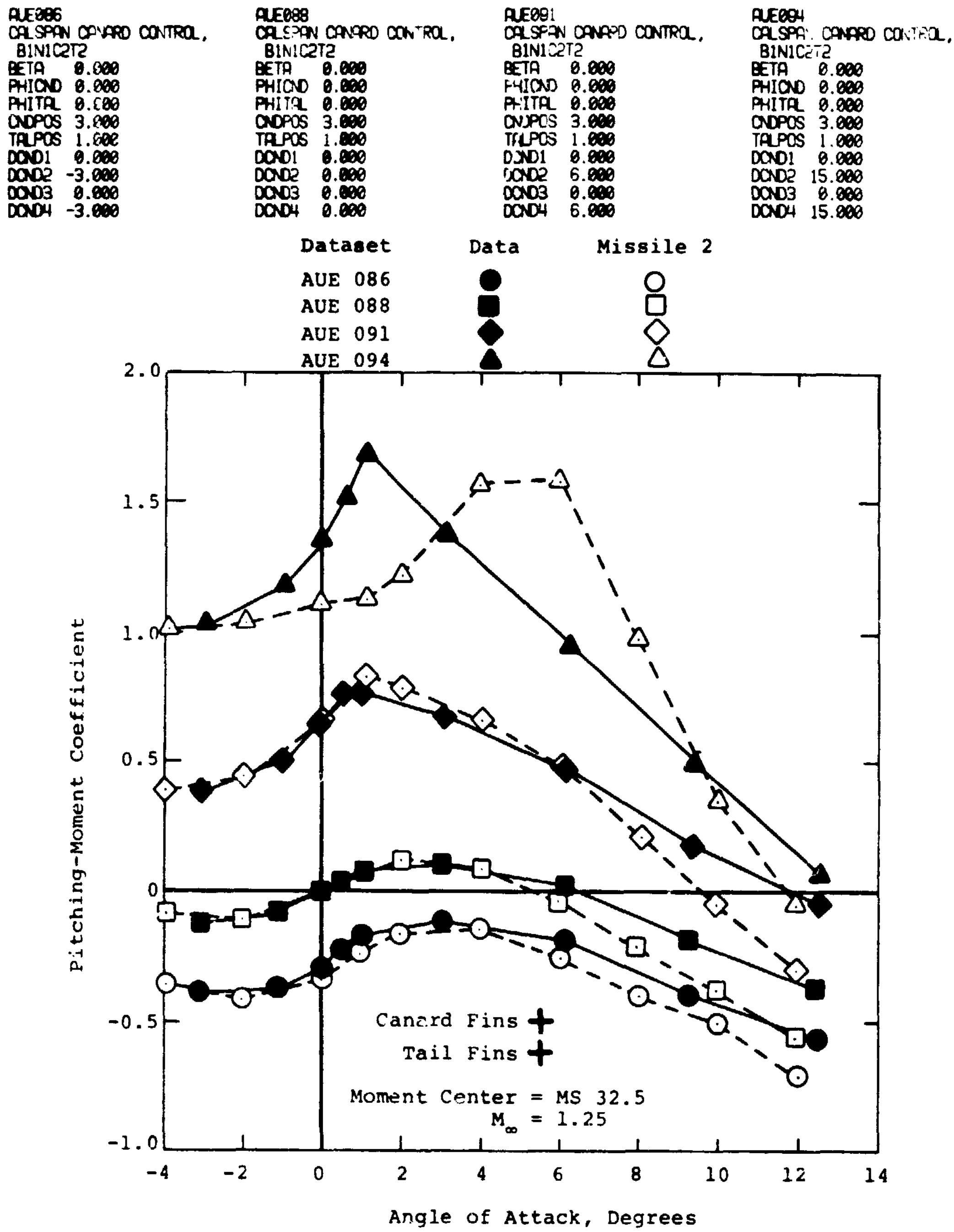
(c) ϕ = 0; interdigitated tail fins.

Figure 8.- Continued.



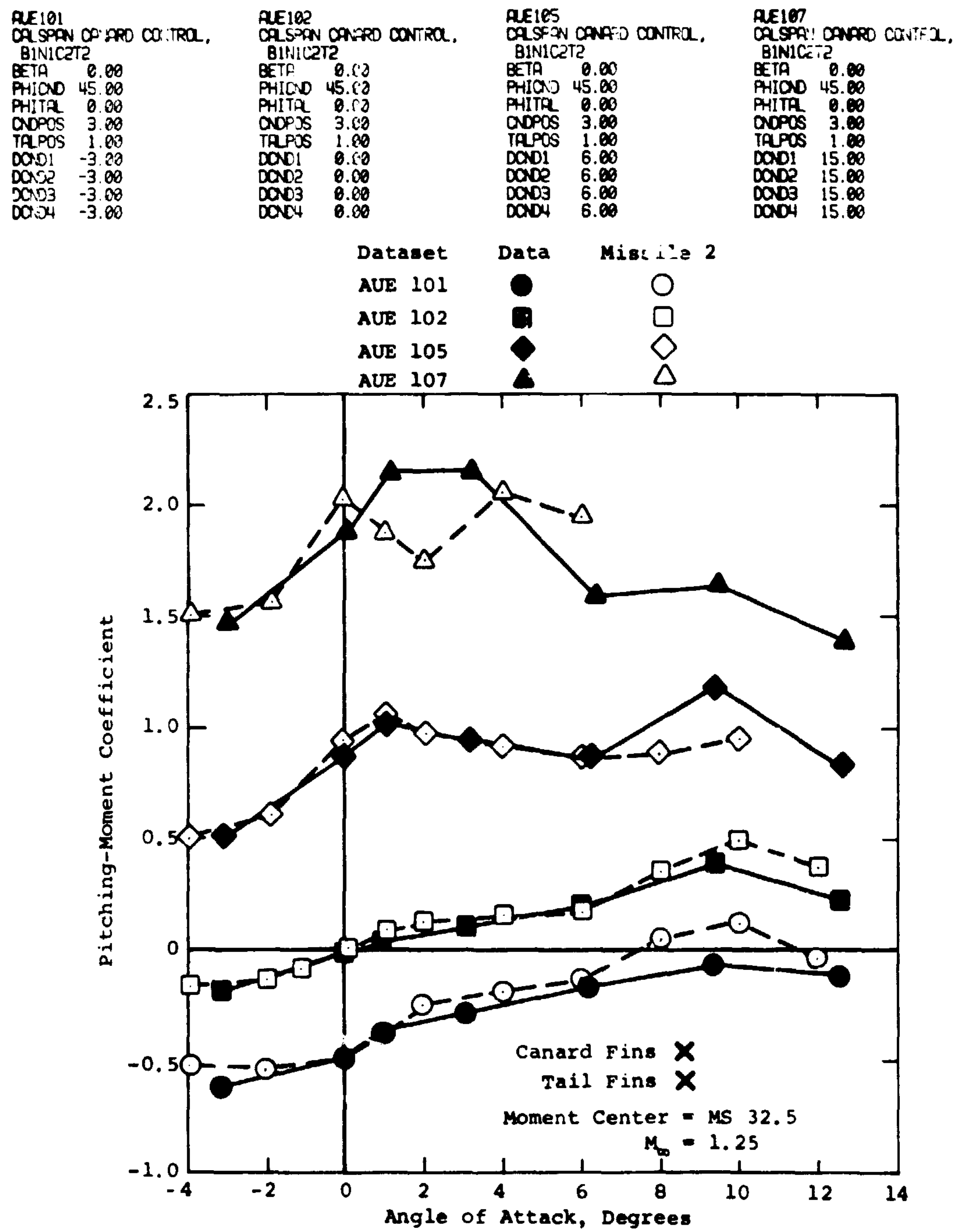
(d) $\phi = 45^\circ$; interdigitated tail fins.

Figure 8.- Concluded.



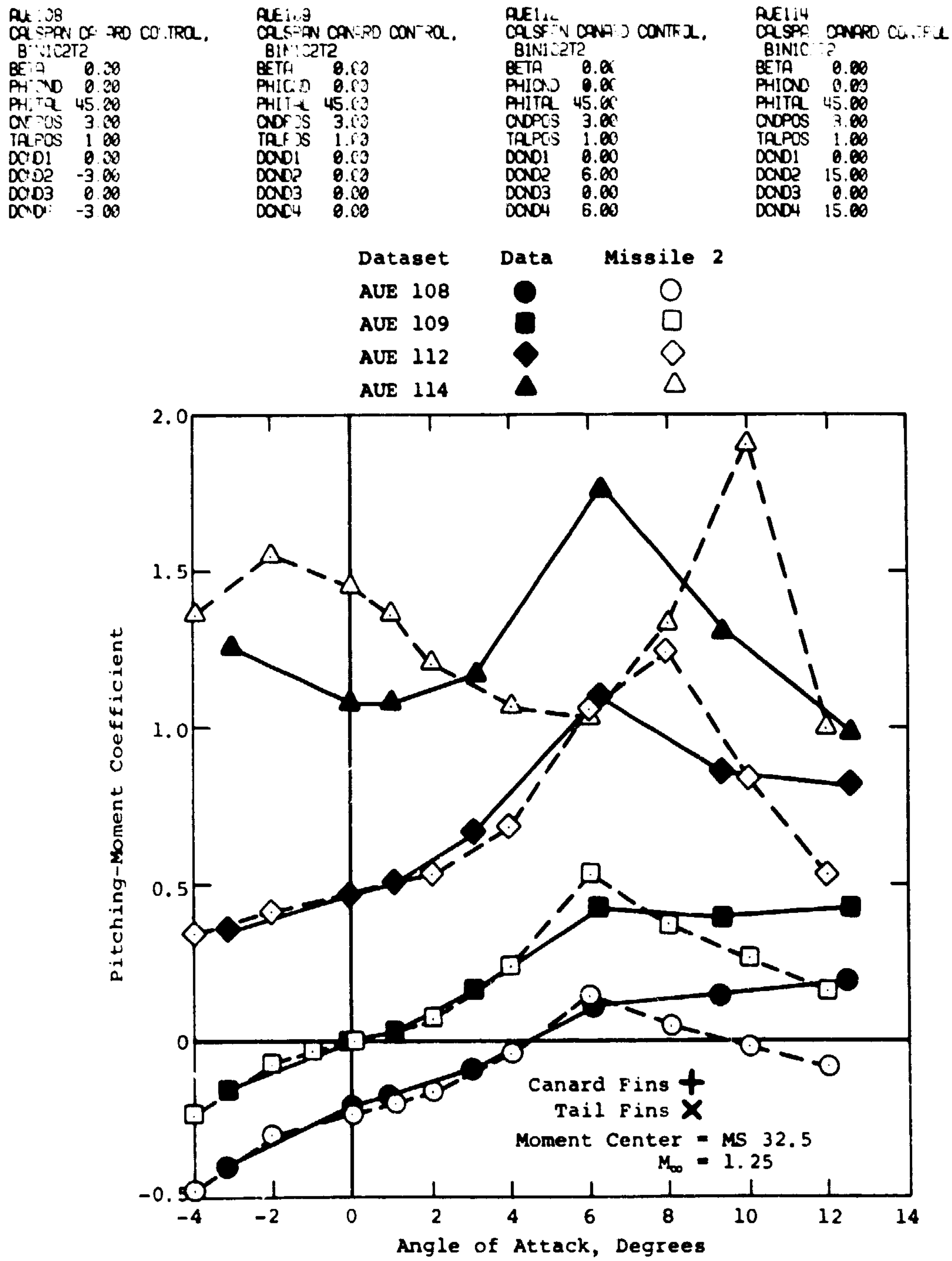
(a) ϕ = 0; inline tail fins.

Figure 9.- Comparison of pitching-moment coefficient data and predictions from Program MISSILE 2 for Burt's Canard-Configuration tests; M_∞ = 1.25.



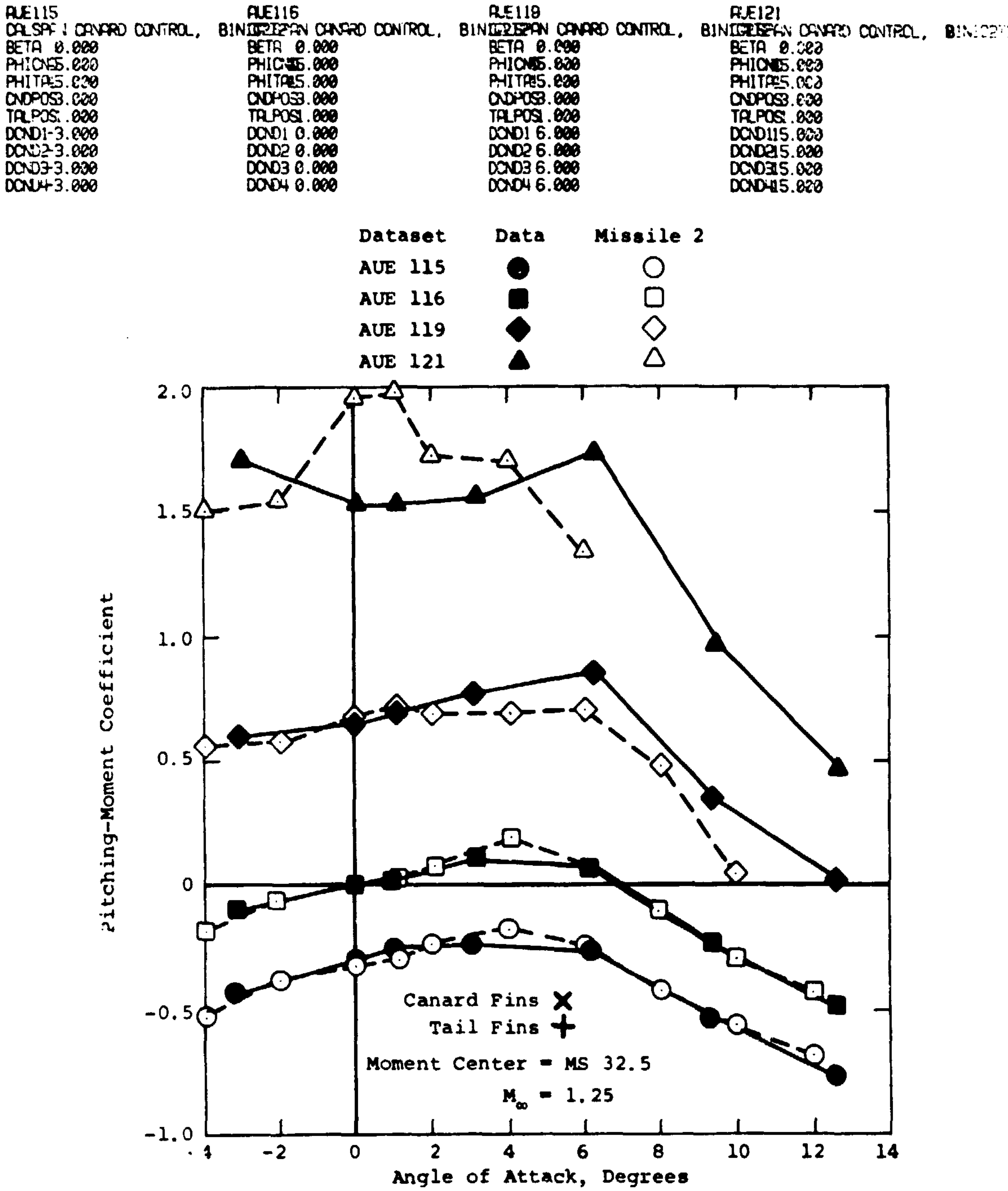
(b) $\phi = 45^{\circ}$; inline tail fins.

Figure 9.- Continued.



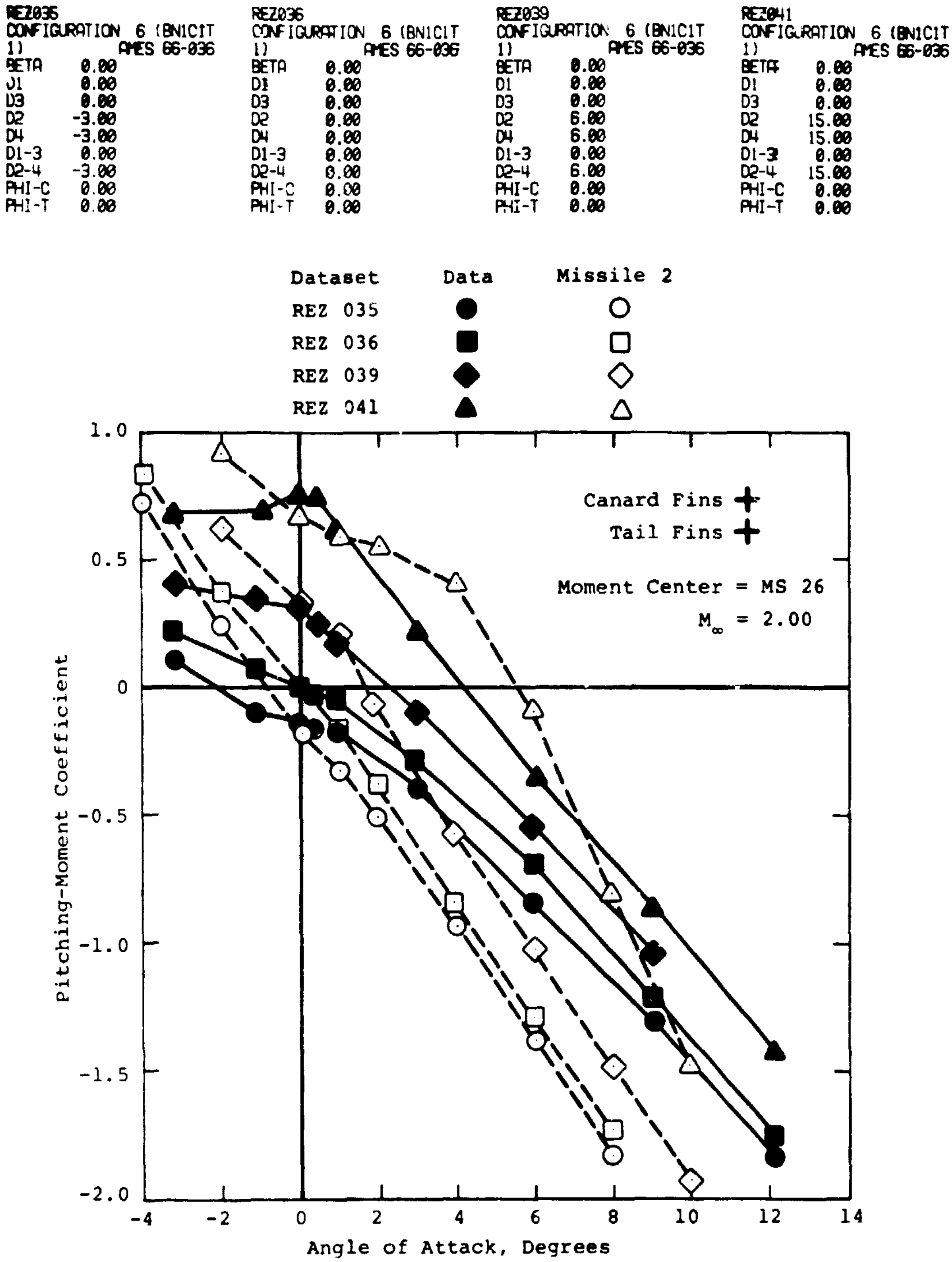
(c) ϕ = 0; interdigitated tail fins.

Figure 9.- Continued.



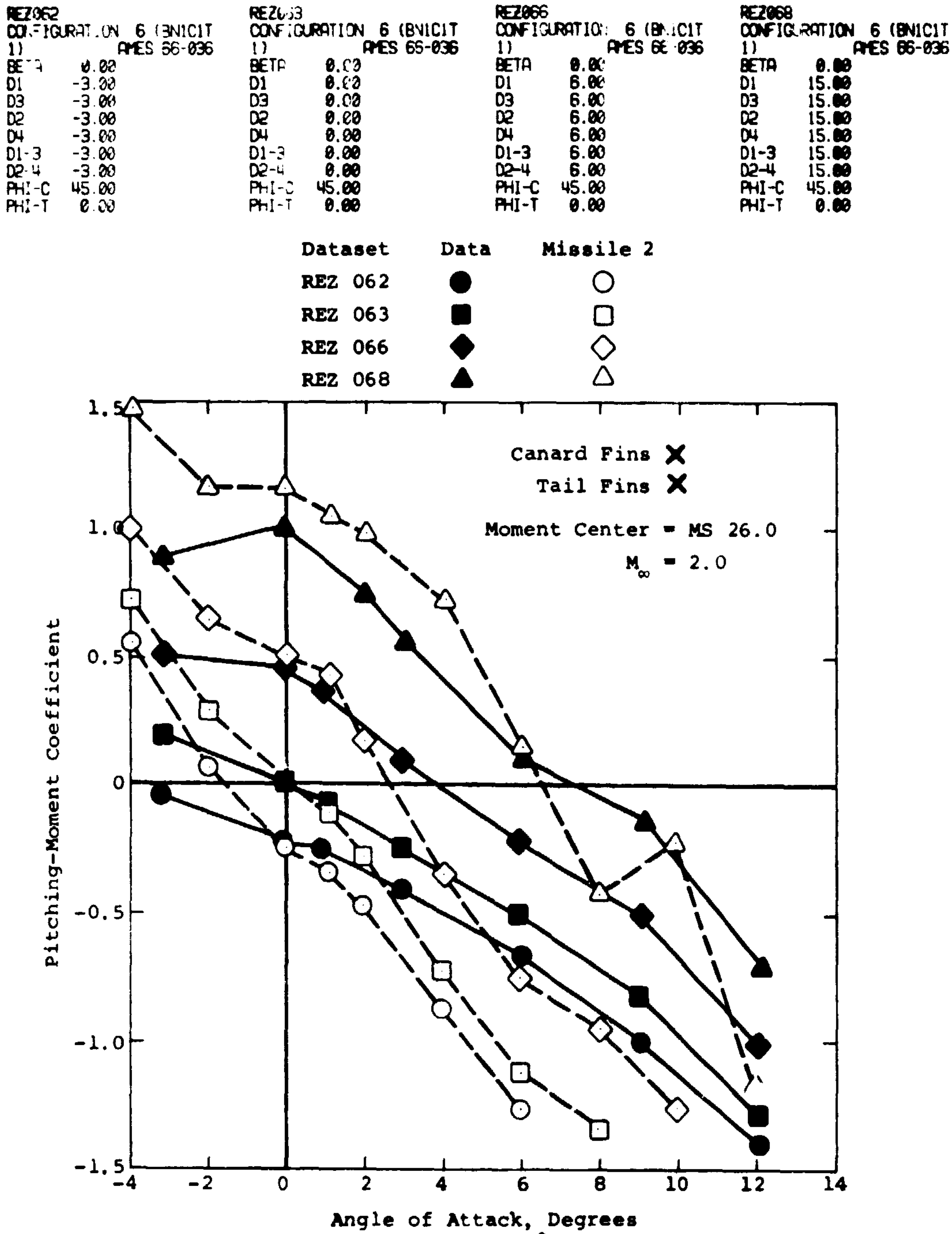
(d) $\phi = 45^\circ$; interdigitated tail fins.

Figure 9.- Concluded.



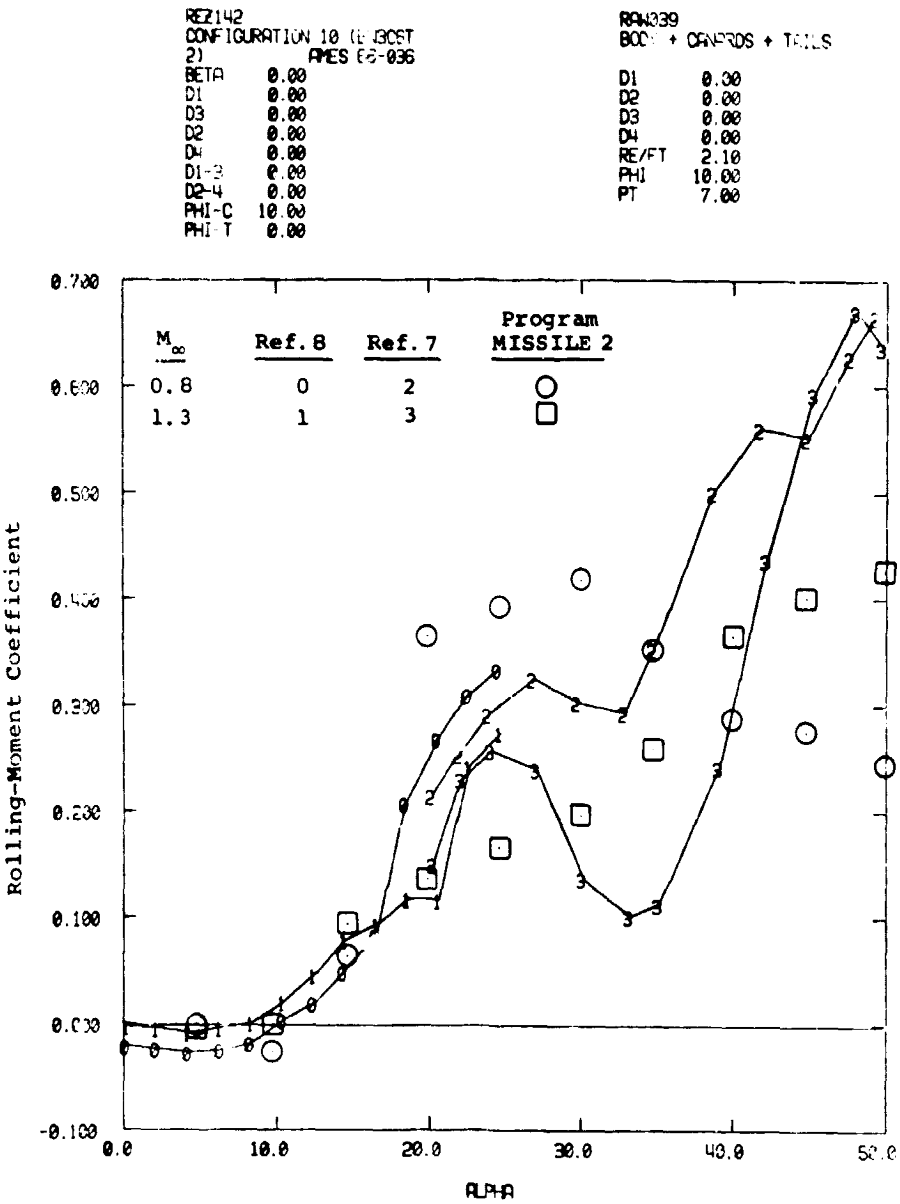
(a) $\phi = 0$.

Figure 10.- Comparison of pitching-moment coefficient data and predictions from Program MISSILE 2 for Burt's Canard-Configuration tests; inline tails; $M_\infty = 2.0$.



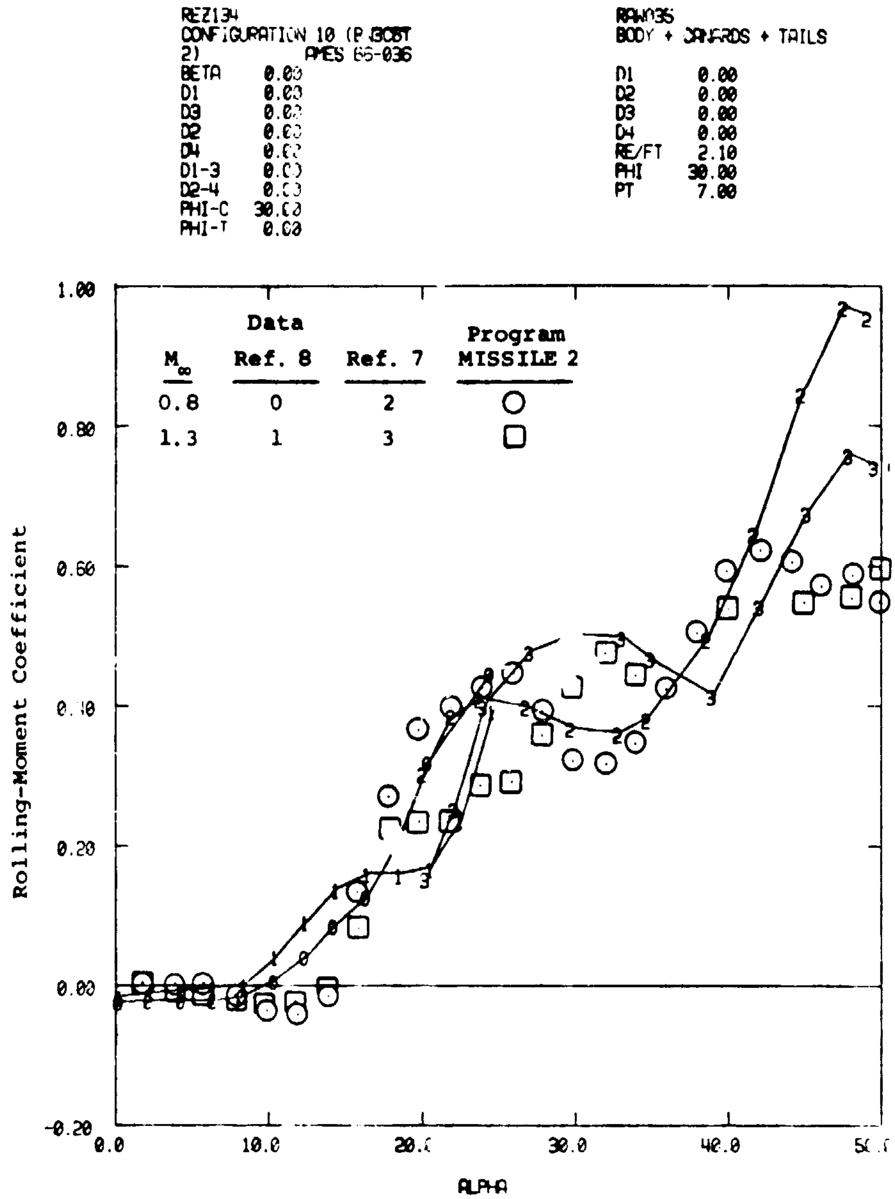
(b) $\phi = 45^\circ$.

Figure 10.- Concluded.



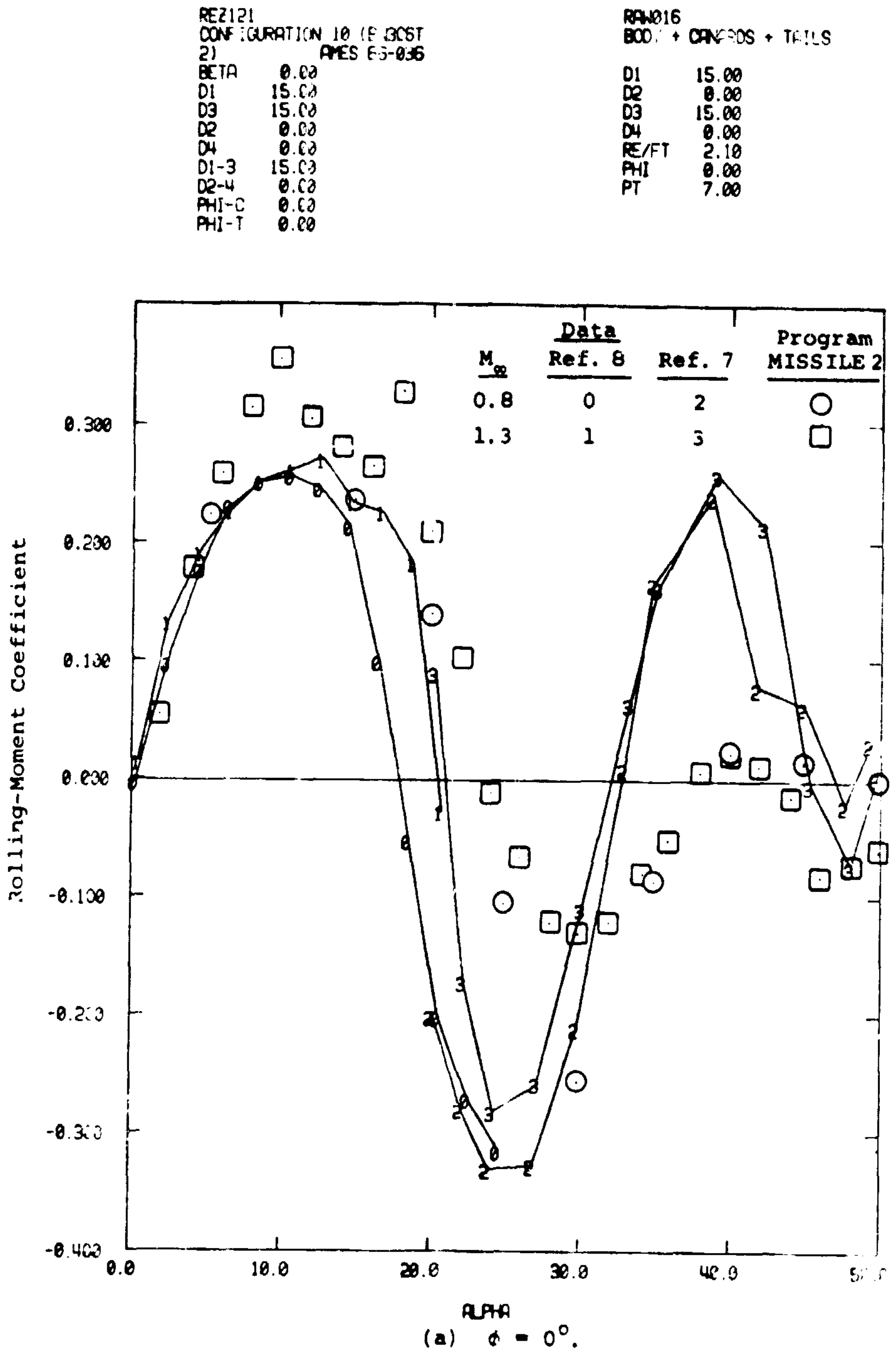
(a) $\phi = 10^\circ$

Figure 11.- Comparison of rolling-moment coefficient data and predictions of Program MISSILE 2 for High-Alpha tests at NASA/Ames. No fin deflections.



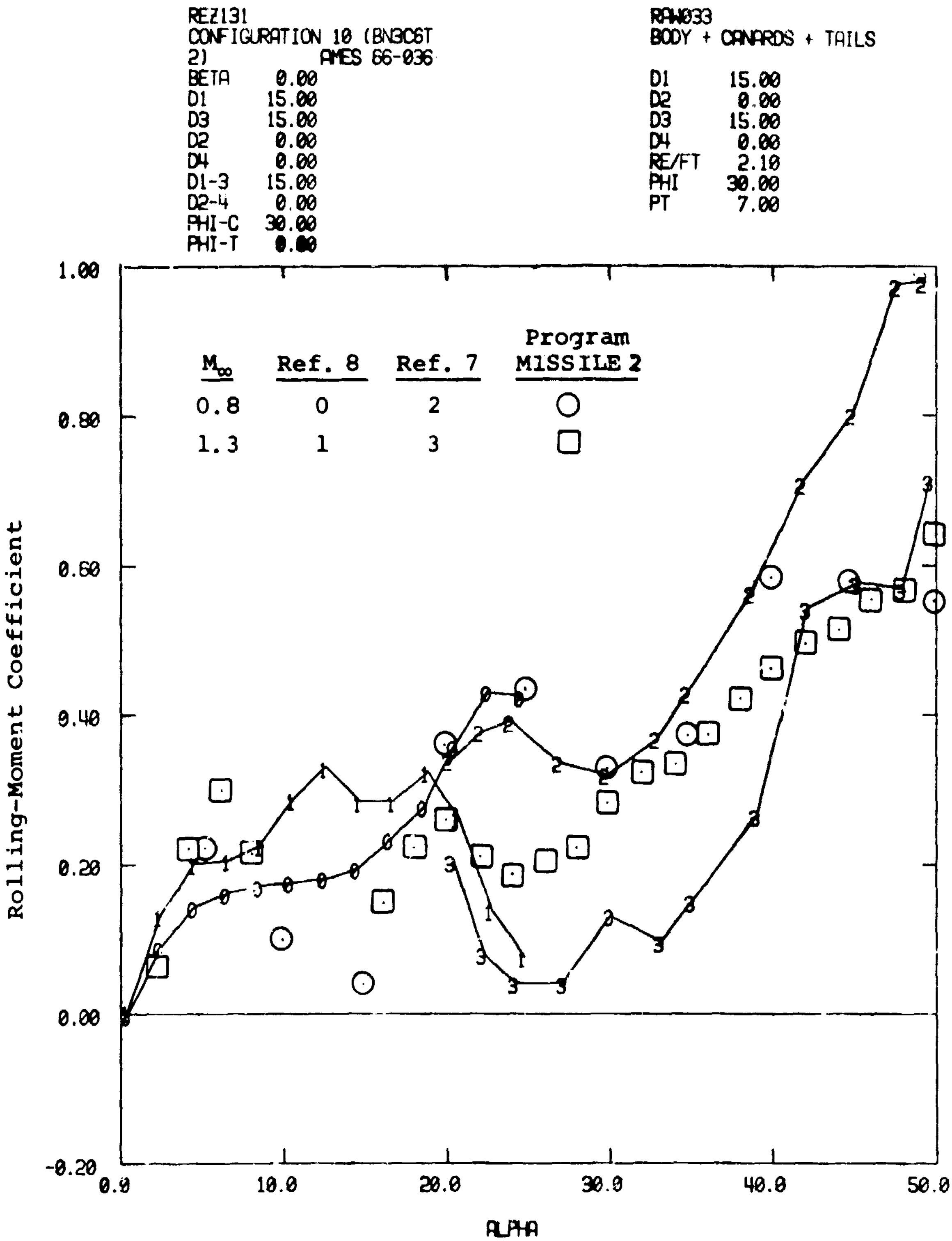
(b) $\phi = 30^\circ$.

Figure 11.- Concluded.



(a) $\phi = 0°$.

Figure 12.- Comparison of rolling-moment coefficient data and predictions of Program MISSILE 2 for High Alpha tests at NASA/Ames. Yaw control fins deflected 15°.



(b) $\phi = 30°$.

Figure 12.- Concluded.

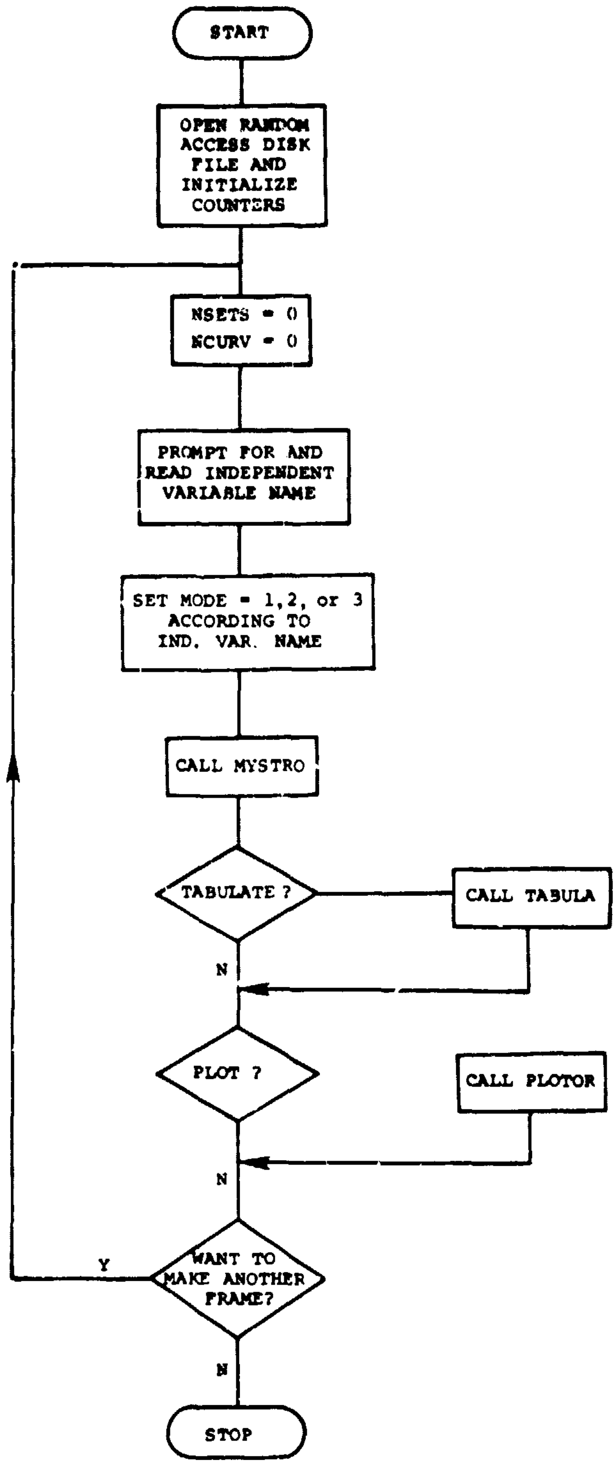


Figure 13.- Flow chart for driver program, MAIN.

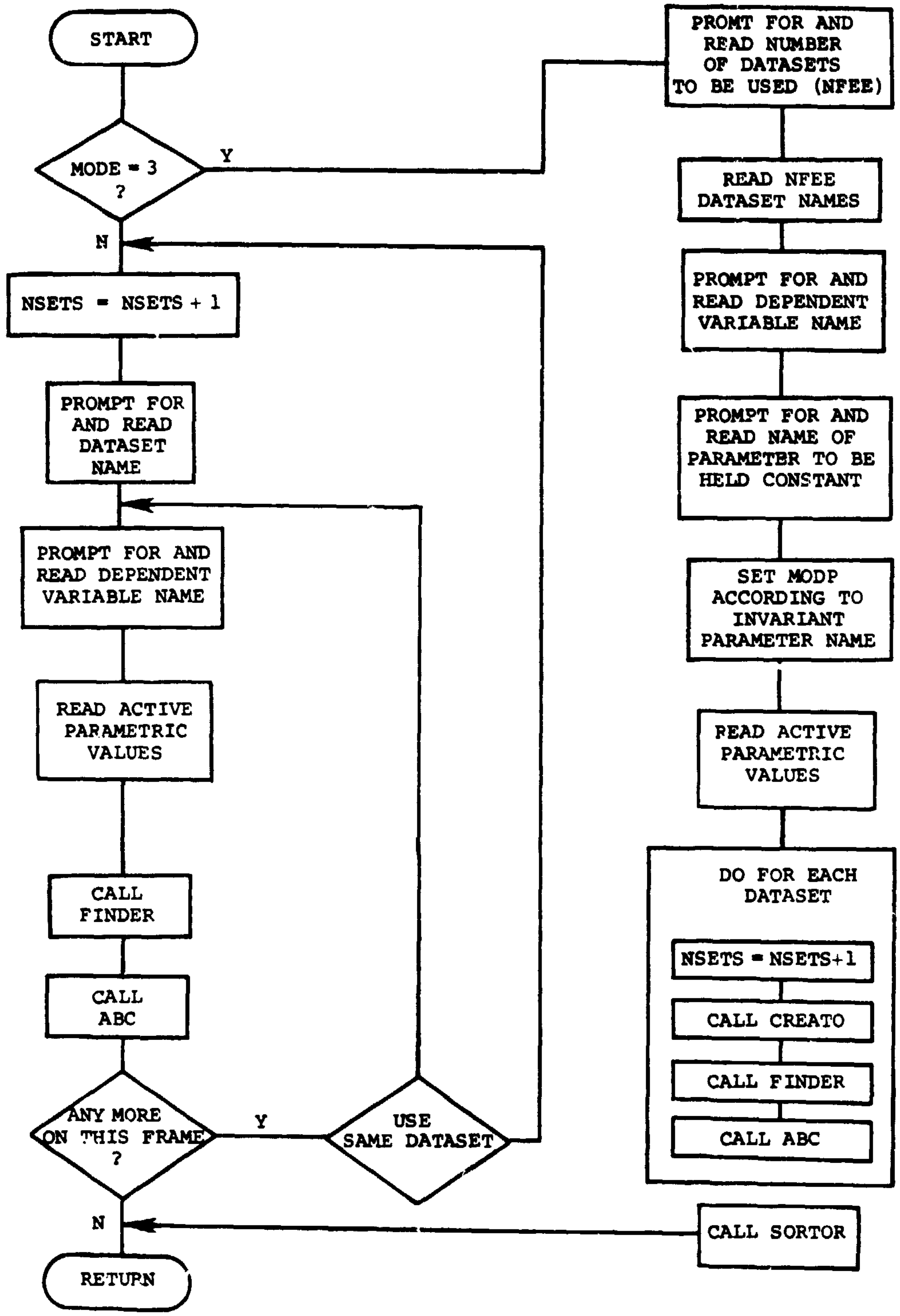


Figure 14.- Flow chart for subroutine MYSTRO.

C-2

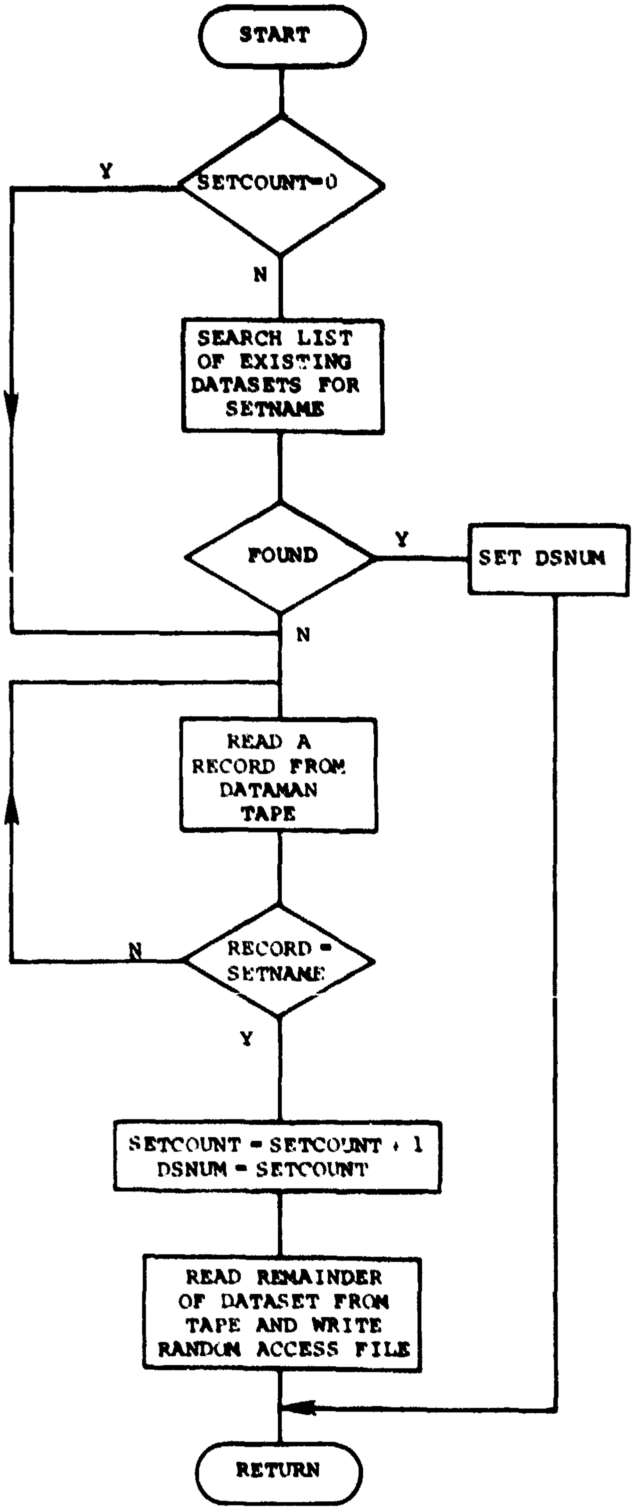


Figure 15.- Flow chart for subroutine CREATO.

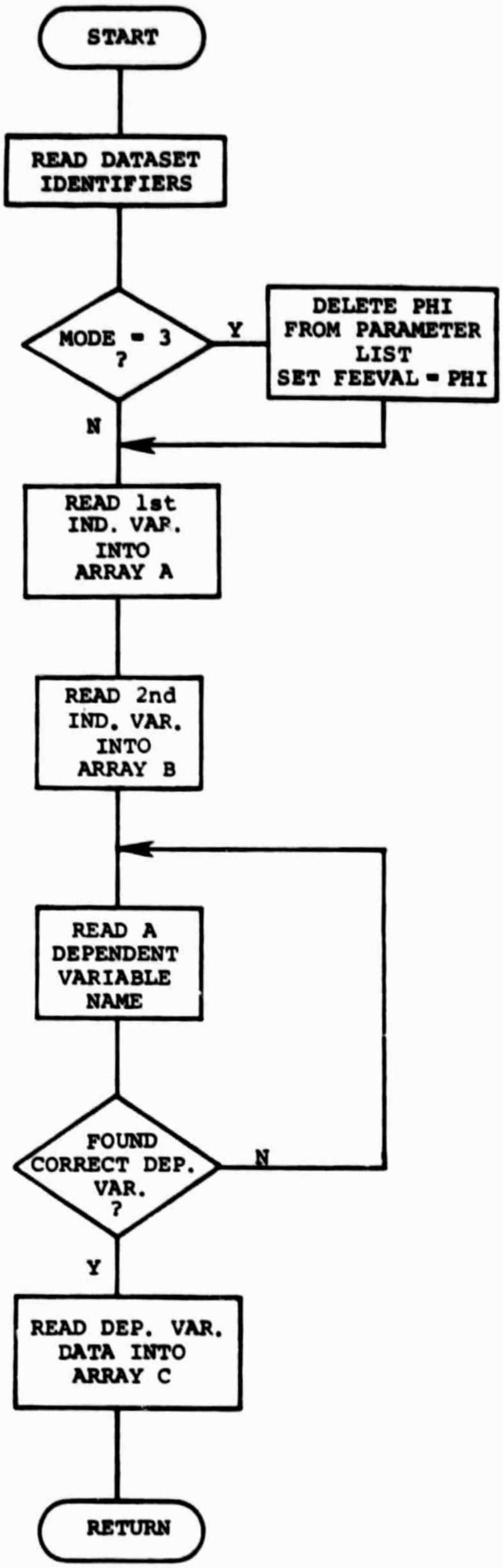


Figure 16.- Flow chart for subroutine FINDER.

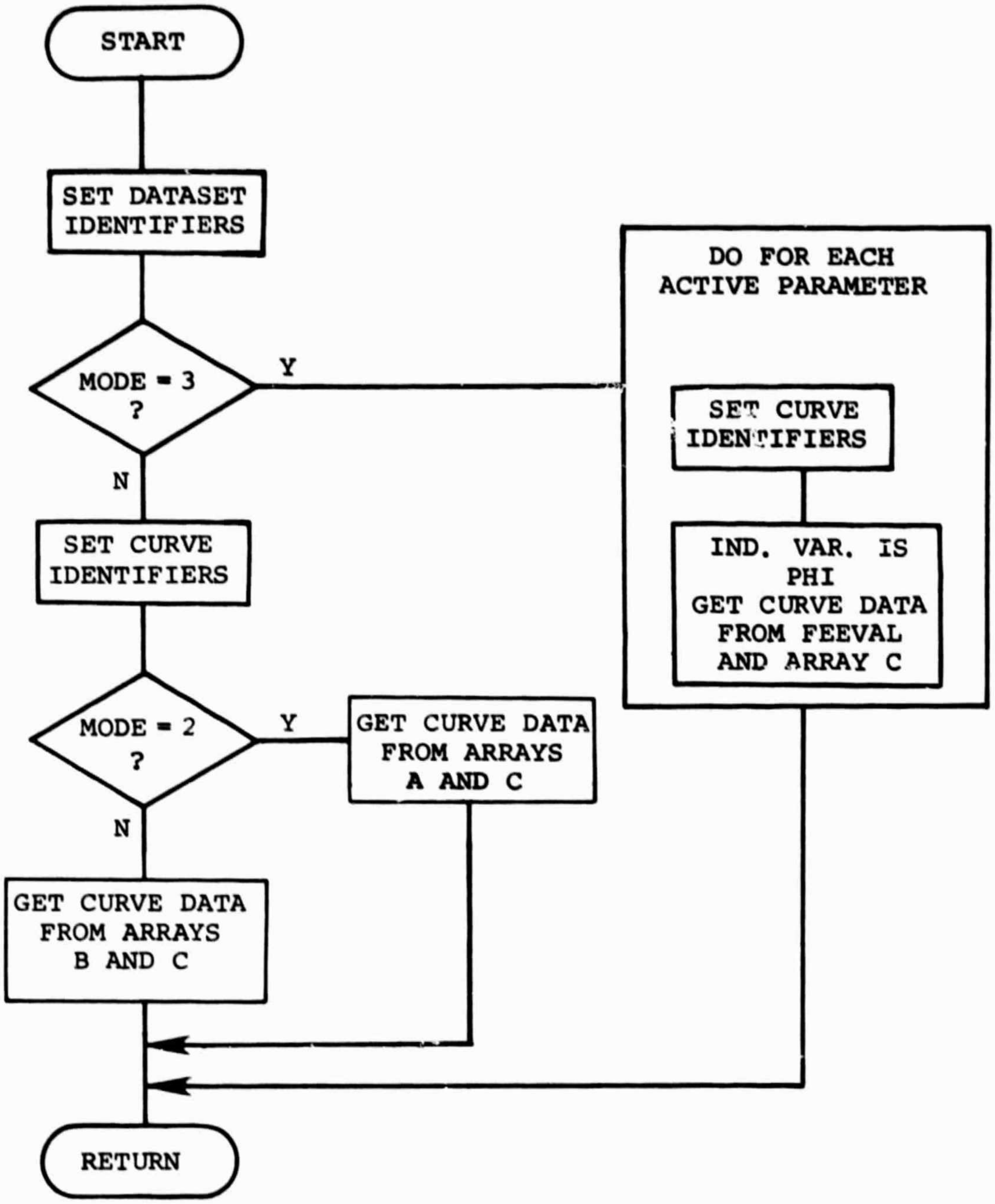


Figure 17.- Flow chart for subroutine ABC.