NASA CONTRACTOR REPORT

NASA CR-112185

NASA CR-112185

DEVELOPMENT AND APPLICATIONS OF SUPERSONIC UNSTEADY CONSISTENT AERODYNAMICS FOR INTERFERING PARALLEL WINGS

PROGRAMMER'S MANUAL

by Andrew A. Paine

Prepared by

Bell Aerospace Company DIVISION OF textron
POST OFFICE BOX ONE
BUFFALO, NEW YORK 14240

for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • AUGUST 1972

NASA CONTRACTOR REPORT

NASA CR-112185

NASA CR-112185

DEVELOPMENT AND APPLICATIONS OF SUPERSONIC UNSTEADY CONSISTENT AERODYNAMICS FOR INTERFERING PARALLEL WINGS:

PROGRAMMER'S MANUAL

By ANDREW A. PAINE

Prepared by

BELL AEROSPACE COMPANY
Post Office Box One
Buffalo, New York 14240

for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION · WASHINGTON D.C.

TABLE OF CONTENTS

SECTION 1

INTRODUCTION

This manual describes the computer program written in support of the problem to determine Aerodynamic Influence Coefficients (AIC's) on parallel interfering wings. (See Ref. 1.)

The information presented here is geared to the programmer. It is sufficient to fully describe the program logic and the required peripheral storage. Figure 1 gives an overview of the entire program, and is the basis for the control program. All element generated information is stored externally to reduce core storage. A separate section is devoted to the development of these external files so that I/O time may be optimized through efficient buffer description. Individual subroutine write-ups are presented along with the complete Fortran source listing. Program alterations (data storage) are also discussed.

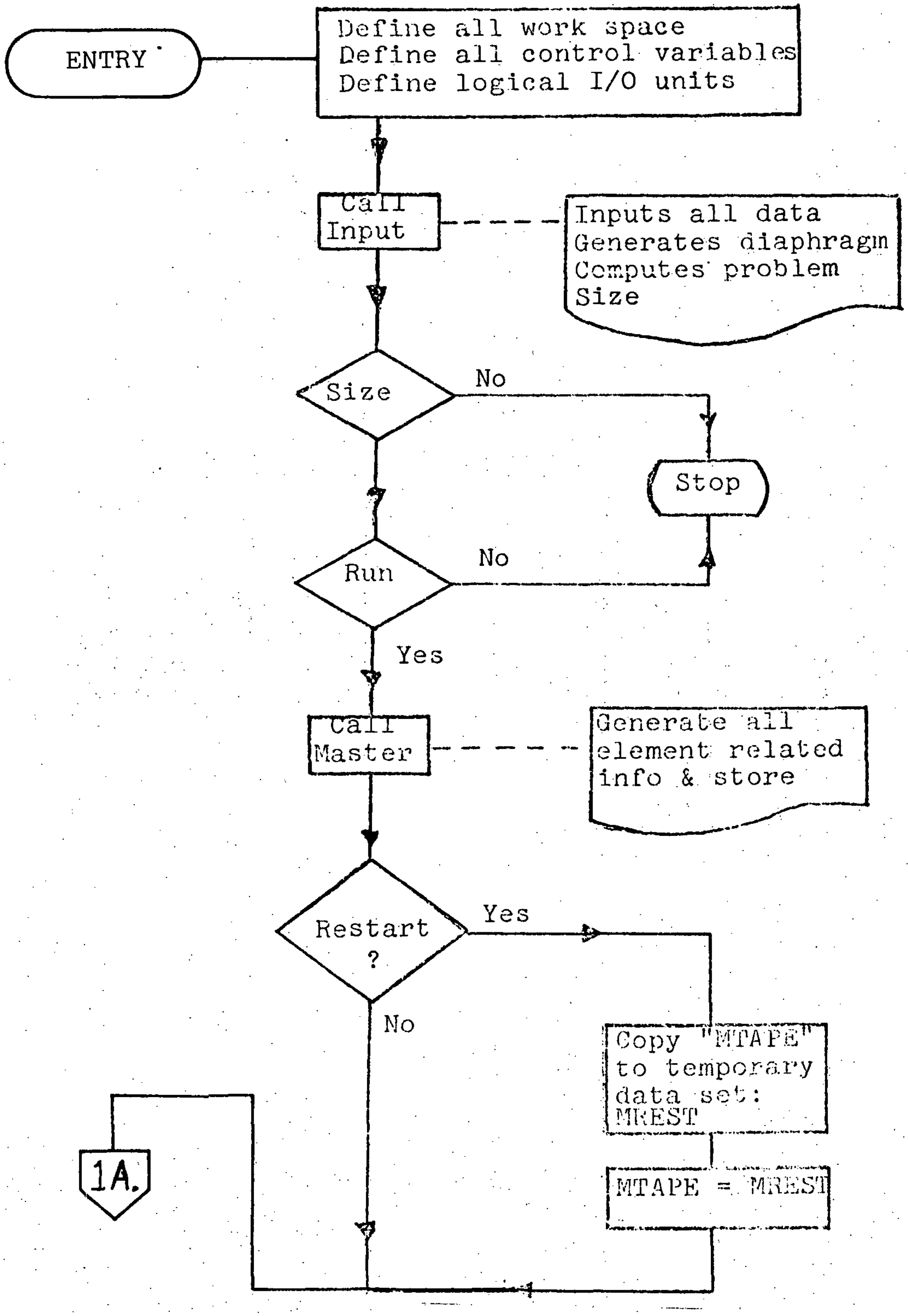


Figure 1.1 Computer Program Overview

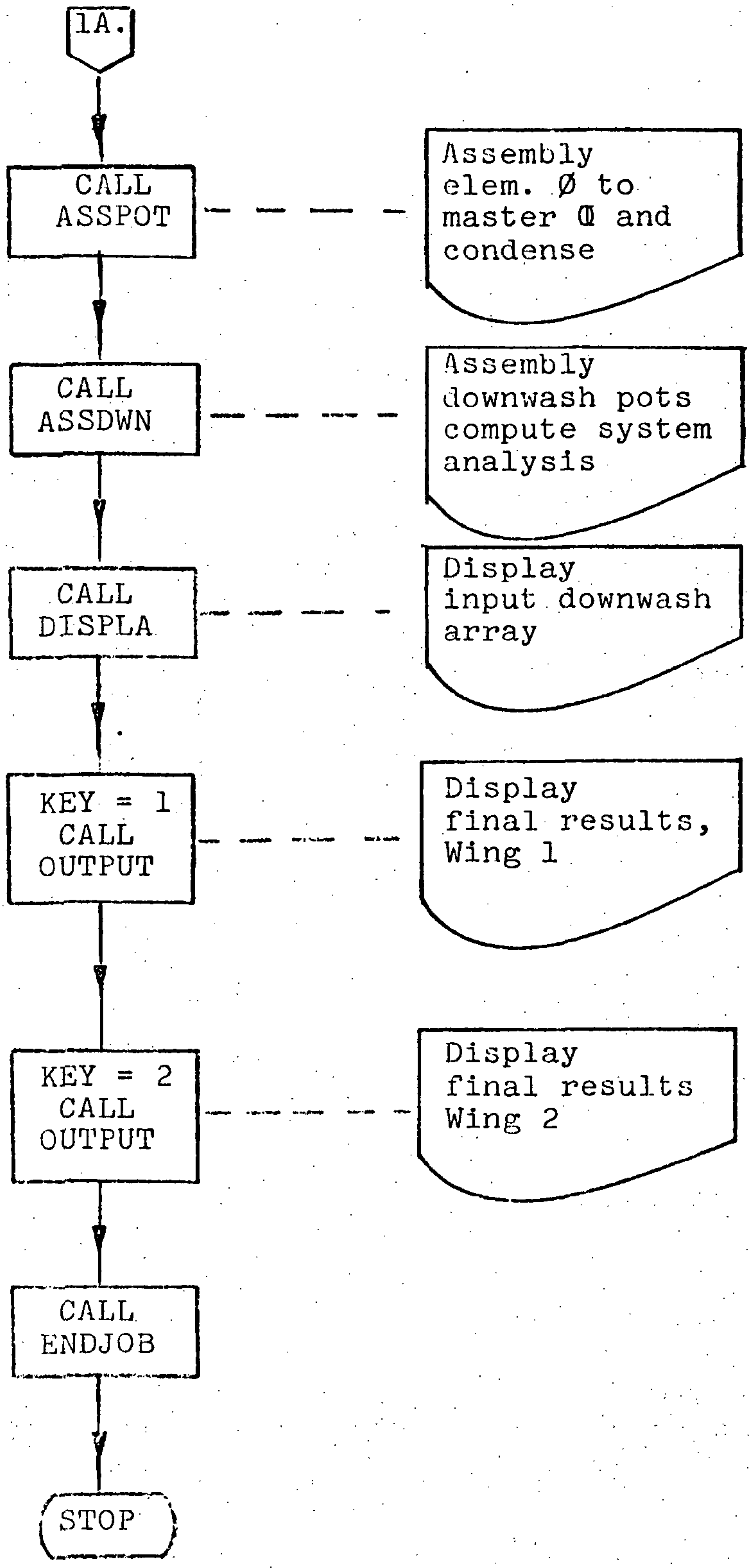

1A.
CALL ASSPOT
Assembly elem. Ø to master Φ and condense
CALL ASSDWN
Assembly downwash pots compute system analysis
CALL DISPLA
Display input downwash array
KEY = 1 CALL OUTPUT
Display final results, Wing 1
KEY = 2 CALL OUTPUT
Display final results Wing 2
CALL ENDJOB
STOP

Figure 1.1 (Continued)

SECTION 2
EXTERNAL FILE STRUCTURE

This program utilizes eight (8) files during the course of operation. The delivery version of the AIC / INT program comes with the following variable names and real unit designations, any or all of which the programmer may alter:

Unit Name	Unit ID	Type	Usage
NTAPE	1	(S,P)	Velocity Potentials
LTAPE	2	(S,P)	Pressure Arrays
MTAPE	3	(S,P)	Downwash Potentials
KTAPE	4	(S)	Scratch
SYSIN	5	(S)	Standard Card Input (80 Col Card)
SYSOUT, OUT	6	(S)	Standard Line Printer (132 Char/Line)
MREST	8	(S)	Restart Only, Store MTAPE
TYSIN, TYSOUT	9	(S)	Store Card Images from SYSIN; also Store Title Cards

where S = Scratch, P = Permanent.

If the restart feature is desired then units so designated (P) must be permanent files.

(NOTE: As units 5 and 6 are standard input and output, they will not be considered in the following discussion.)

Unit TYSIN=TYSOUT has a logical record length (LRECL) of 20 words (@ 4-literal/word).

Unit NTAPE uses LRECL = 14 words only.

Unit LTAPE, KTAPE uses LRECL = variable

Unit MTAPE uses both of above.

All complete, complex, matrices are stored on tape by Subroutine TINOUT. These arrays are stored column-wise. Each is preceded by a header, whose first word is "-10" and a trailer, whose first word is "-20". Element type information (LRECL = 14) has only the end of information defined by a trailer, whose first word is "-20". Figure 21 gives an overview of the information stored on specific units.

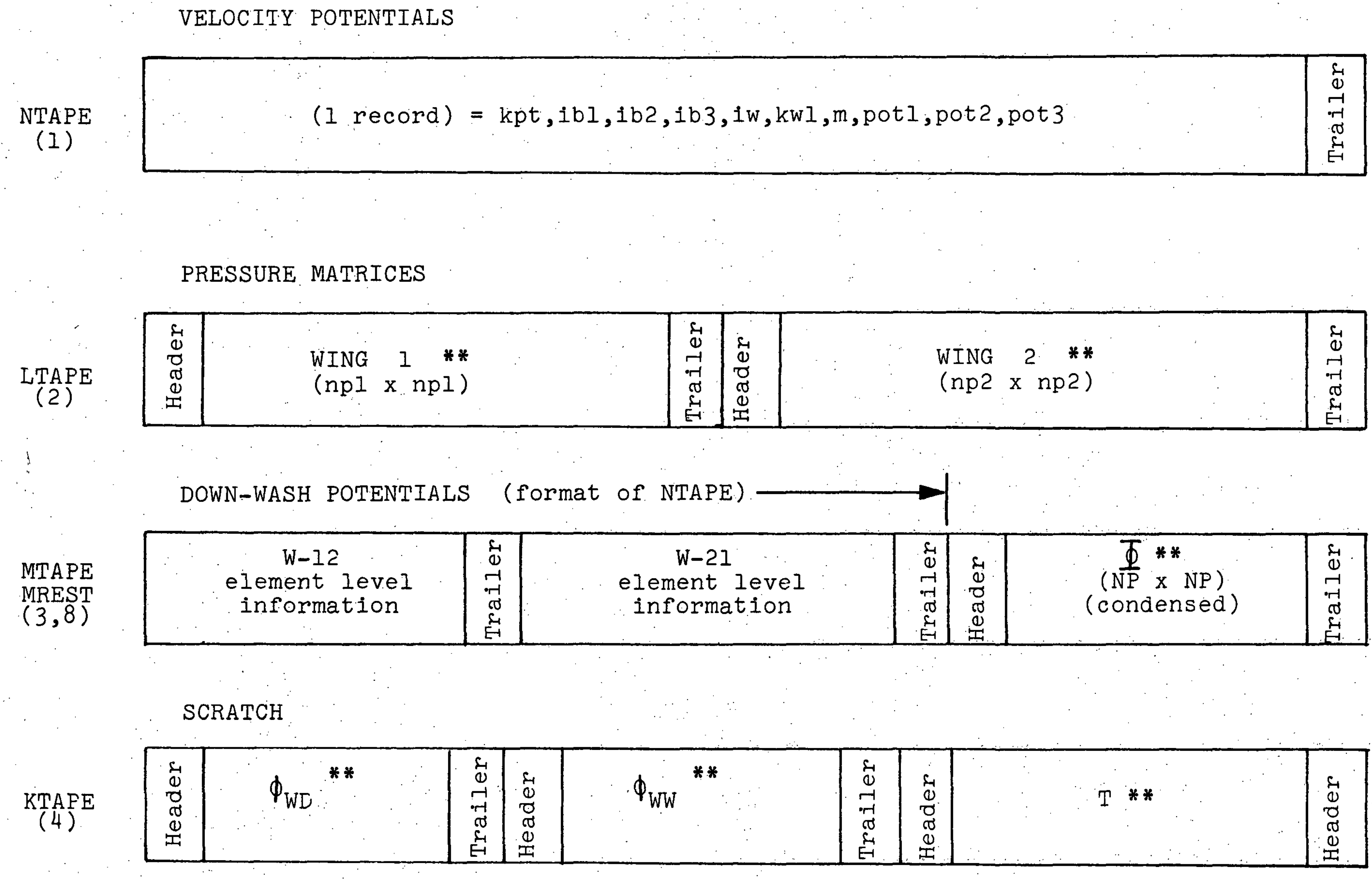


Figure 2.1 External File Structure

Those areas marked ** indicate that an array was written by Subroutine TINOUT.

SECTION 3

SUBROUTINE WRITE-UPS

The AIC / INT program encompasses 25 subprograms, each with a unique sequence number. Columns 73, 74, 75 contain the "DECK" number and columns 76 thru 80 contain the card sequence number for that subprogram. The first card is always ---00010 with successive increments of 10.

Included with each description is a statement declaring the size of the program. This number is intended as a guide only, as it reflects the storage requirement of an IBM/360/65, FORTRAN IV G level 19 compiler.

In this manual the subroutine writeups are presented in alphabetical order. These are cross-referenced by deck number on the following page.

SUBROUTINE WRITE-UPS

1. Subroutine Name: ASSDWN

2. Purpose:

Assemble downwash contribution to the potentials, condense and compute the final velocity potential distribution across the wing surface.

3. Equations and Procedures:

Downwash influence coefficients at nodes of wing 1 due to the presence of wing 2 are written as

$$\left[W_{12}\right] = \left[W_{12_w} \;\vdots\; W_{12_d}\right]$$

where the subscripts **w** and **d** denote the source strength situated on the wing and diaphragm respectively. Similarly, for wing 2 due to wing 1,

$$\left[W_{21}\right] = \left[W_{21_w} \;\vdots\; W_{21_d}\right]$$

The solution for the source strengths on the upper surface of wing 1 (σ_{1u}) and the lower surface of wing 2 (σ_{2L}) is computed from

$$\begin{Bmatrix} \sigma_{1u} \\ \\ \sigma_{2L} \end{Bmatrix} = \begin{bmatrix} W - W_D \cdot \tilde{\Gamma} \end{bmatrix}^{-1} \left[\begin{Bmatrix} \dfrac{DZ_1}{Dt} \\ -\dfrac{DZ_2}{Dt} \end{Bmatrix} + \begin{bmatrix} W_D \cdot \tilde{\Gamma} \end{bmatrix} \begin{Bmatrix} \dfrac{DZ_1}{Dt} \\ -\dfrac{DZ_2}{Dt} \end{Bmatrix} \right]$$

$$[W] = \left[\begin{array}{c:c} I_{11} & W_{12_w} \\ \hdashline -W_{21_w} & I_{22} \end{array}\right]$$

$$[W_D] = \left[\begin{array}{c:c} 0 & -W_{12_d} \\ \hdashline W_{21_d} & 0 \end{array}\right]$$

$$[\tilde{\Gamma}] = \frac{1}{2}\begin{bmatrix} -I_{11} & 0 \\ 0 & I_{22} \end{bmatrix} \begin{bmatrix} \Gamma \end{bmatrix} \begin{bmatrix} I_{11} & 0 \\ 0 & -I_{22} \end{bmatrix}$$

The array Γ is computed in subroutine ASSPOT and DZ/Dt is the input downwash computed in subroutine master.

The total source strengths across the wing are given by:

$$2\,\Delta\sigma_1 = (\sigma_{1u} - \sigma_{1L})$$

$$2\,\Delta\sigma_2 = (\sigma_{2u} - \sigma_{2L})$$

$$\sigma_{1L} = -\frac{DZ_1}{Dt}$$

$$\sigma_{2u} = \frac{DZ_2}{Dt}$$

4. Input Arguments:

INDWN	=	input downwash array
NP1	=	total no. nodes (wing only), wing 1
NP2	=	total no. nodes (wing only), wing 2
ND1	=	total no. nodes (diaphragm only) wing 1
ND2	=	total no. nodes (diaphragm only) wing 2
NP	=	total no. nodes = NP1 + NP2

4. Input Arguments: (cont'd)

ND = total no. nodes = ND1 + ND2
NP2ND2 = total no. nodes wing 2 = NP2 + ND2
NP1ND1 = total no. nodes wing 2 = NP1 + ND1
NMOD = no. mode shapes
MTAPE = unit defining downwash potentials
KTAPE = unit defining velocity potentials

5. Output Arguments:

All input, unmodified

6. Error Returns: None

7. Calling Sequence:

```
CALL ASSDWN (W12, W21, A, WT, WORK, T, INDWN, NP, NP1, NP2,
            ND1, ND2, ND, MTAPE, KTAPE, NMOD, NP2ND2, NP1ND1,
            C)
```

8. Input Tapes:

MTAPE = down wash potentials
KTAPE = condensed velocity potentials

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required:

5474 bytes (1369 words)

12. Subroutine User: MAIN

13. Subroutine Required:

CMATPR, TINOUT, STORE, CMPRD, CMINV

14. Remarks:

W12, W21, A, WT, WORK, T. C
are all complex arrays, used as work space in subroutine.

1. Subroutine Name: ASSPOT

2. Purpose:

Assemble from element information the condensed velocity potential array.

3. Equations and Procedures:

$$\left[T \right] = \left[\phi_{DD} \right]^{-1} \left[\phi_{DW} \right]$$

$$\left[\Phi_{condensed} \right] = \left[\phi_{WW} \right] - \left[\phi_{WD} \right] \left[T \right]$$

4. Input Arguments:

PHI	=	work space
PHIXX	=	work space
PHI22	=	work space
NP1	=	Nbr grid points, wing 1
NP2	=	Nbr grid points, wing 2
ND1	=	Nbr grid points diaphragm 1
ND2	=	Nbr grid points diaphragm 2
NTAPE	=	tape logical unit nbr
KTAPE	=	" " " "
MTAPE	=	" " " "

5. Error Returns: None

6. Calling Sequence:

```
Call ASSPOT (PHI,PHIXX,PHI22,T,NP1,NP2,ND1,ND2,NTAPE,KTAPE,MTAPE)
```

7. Input Tapes: UNIT = NTAPE

8. Output Tapes: UNIT = KTAPE
UNIT = MTAPE

9. Storage Required:

2750 bytes (688 words)

10. Subroutine User:

MAIN

11. Subroutine Required:

ASSY, CORED, TINPUT, CMINV, CMATPR, CRASH, CMPRD

12. Remarks:

None

1. Subroutine Name: ASSY

2. Purpose:

Assemble a large matrix from individual element information stored on tape.

3. Equations and Procedures:

Records on input are exactly 14 words long.

Word (1)	=	row position
Word (2)	=	col position for word (9)
Word (3)	=	col position for word (11)
Word (4)	=	col position for word (13)

4. Input Arguments:

PHI	=	Array to be assembled, previously cleared
N	=	Row dimension of PHI
NP1	=	Nbr grid points, wing 1
NP2	=	Nbr grid points, wing 2
ND1	=	Nbr diaphragm points, wing 1
NTAPE	=	Unit (tape)
NSET	=	Assembly branch (1, 2, 3, or 4)

5. Output Arguments:

PHI = Assembled array. Because of program logic and/or problem influence, array may be partially empty.

6. Error Returns:

End-Of-File condition stops program with a value "7".

7. Calling Sequence:

Call ASSY (PHI,N,NP1,NP2,ND1,NTAPE,NSET)

8. Input Tapes: UNIT = NTAPE

9. Storage Required: 2822 bytes (705 words)

10. Subroutine User: ASSPOT

11. Subroutine Required: None

NAME COMMON: /BASIC/

This common statement appears in all subroutines. It contains all the basic control data required for operation and control of the total program.

The total length of this common is 100 words. Words 1 through 15 apply only to Wing 1; words 16 through 30 apply only to Wing 2. The array is defined as follows:

Word	Variable	Meaning
1,16	NEL	Total # elements on planform
2,17	CAPPA	Frequency (pure imaginary - stored as real)
3,18	MLINE	# grid points on mach line
4,19	NMOD	# mode shapes
5,20	NP	# grid points (wing only)
6,21	NELW	# elements (wing only)
7,22	NGP	Total # grid points on planform
8,23	ND	# grid points diaphragm
9,24	NFLD	# elements in diaphragm
10,25	SYMC	Symmetry factor
11,26	NDIV	# divisions diaphragm
12,27	REF	Reference length
13,28	NLND	# leading edge grid points
14,29	THETA	Wing angle
15,30	-	N/A
30-45	-	N/A
46	PI	PI = 3.1415 etc
47	NP	NP1 + NP2
48	-	NP1 + ND1
49	-	NP2 + ND2
50	ND	ND1 + ND2
51	-	NP * NP
52	-	ND * ND
53	NPND	NP * ND
54	KTAPE	Tape, Unit = 4
55	LTAPE	Tape, Unit = 2
56	MTAPE	Tape, Unit = 3

Word	Variable	Meaning
57	NTAPE	Tape, Unit = 1
58	SYSOUT	Line printer 6
59	SYSIN	Card reader 5
60	TYSIN	Temporary card reader 9
61	KPRINT	MAIN
62	KPRINT	INPUT, DATA, DIAPH3
63	KPRINT	MASTER, ORIENT
64	KPRINT	TRINTX
65	KBRINT	PSI
66	KPRINT	DWASHY
67	KPRINT	ASSPOT, ASSY
68	KPRINT	ASSDWN
69	KPRINT	CMINV
70	KPRINT	TAPES (22)
71	KPRINT	STORE, TINOUT
72	KPRINT	CMPRD
73	-	N/A
74	-	N/A
75	IWING	Master, receiving element is on
76	KWING	Master, influencing element is on
77	L	Master, receiving element
78	M	Master, influencing
79	IN	Master, node point #
80	SWITCH	Master, branch control
81	ISW	Master, branch control
82	-	N/A
83	-	Punch option (yes≠0)
84	IPUN	Punch Unit = 7
85	EPS	eps.
86	RESTART	NO = 0, YES = -1

Word	Variable	Meaning
87	MACH	Mach number
88	VEETA	$\beta = \sqrt{MACH^2-1}$
89	NTITLE	# title cards
90	ISOLAT	Extrapolation; Yes =1, No = 0
91	NEWPTS	# new grid points for extrapolation
92	MU	Mach angle
93	D	Stagger distance
94	-	N/A
95	-	N/A
96	-	N/A
97	DEPS	Convergence criteria, PSI (sum)
98	-	N/A
99	NERR	Errors generated
100	RUN	Yes = 1, No = 0

1. Subroutine Name: CMATPR

2. Purpose:

   To display a complex array of order N,M in a formal fashion.

3. Equations and Procedures:

   Total number of columns to be printed is broken into integral groups of 4 and displayed as rows 1 to N, columns I to J, when $(J - I + 1) \leq 4$

4. Input Arguments:

   A = Complex array, order N*M exactly
   N = Number of rows in "A"
   M = Number of columns in "A"
   NAME = Literal data, 4 characters long

5. Output Arguments: None

6. Error Returns: None

7. Calling Sequence: Call CMATPR (A,N,M,4HNAME)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: 1196 bytes (299 words)

12. Subroutine User: DISPLA, OUTPUT, ASSDWN, ASSPOT

13. Subroutine Required: None

14. Remarks:

    The literal data stored in NAME is a control for header information to be displayed from this subroutine. If NAME = 4HNONE, no header information is displayed. Otherwise, NAME ,N,M are displayed at top of the array.

1. Subroutine Name: CMINV

2. Purpose: Invert a matrix

3. Equations and Procedures:

   The standard Gauss-Jordan method is used. The determinant is also calculated. A determinant of zero indicates that the matrix is singular.

4. Input Arguments:

   A = input matrix; complex
   N = order of "A"

5. Output Arguments:

   A = inverse of input matrix;complex

6. Error Returns: None

7. Calling Sequence:

   CALL CMINV (A, N)

8. Storage Required: 3484 bytes (871 words)

9. Subroutine User: ASSDWN, ASSPOT

10. Subroutine Required: None

11. Remarks:

    a. Input array must be stored as a general matrix

    b. A singular array terminates program with a message.

1. **Subroutine Name:** CMPRD

2. **Purpose:**

Special purpose complex matrix multiplication.

$R(N,L) = A(N,M) * B(M,L)$

3. **Equations and Procedures:**

$$r(i,j) = \sum_{k=1}^{m} a(i,k) * b(k,j)$$

All arrays are treated as singularly subscripted, stored column wise and fully packed. This is a row into column product.

4. **Input Arguments:**

A = pre-multiplier array, order (N,M): complex

B = post-multiplier array, order (M,L): complex

N = row dimension of "A"

M = column dimension of "A"

L = column dimension of "B"

5. **Output Arguments:**

R = resultant array, order (N,L): complex

6. **Error Returns:** None

7. **Calling Sequence:**

CALL CMATPR (A,B,R,N,M,L)

8. **Storage Required:** 728 bytes (182 words)

9. **Subroutine User:**

OUTPUT
ASSDWN
ASSPOT

10. **Subroutine Required:** None

11. **Remarks:** All three arrays must occupy separate locations

1. Subroutine Name: CORED

2. Purpose:

   Purpose is to force a CORE DUMP of main storage as an aid to locating the reason for calling this routine.

3. Equations and Procedures: Self-explanatory

4. Input Arguments: None

5. Output Arguments: None

6. Error Returns: None

7. Calling Sequence: CALL CORED

8. Storage Required: 308 bytes (77 words)

9. Subroutine User: CMINV, ASSPOT, DIAPH3

10. Subroutine Required: None

11. Remarks:

    NOTE This subroutine must be modified to fit the machine on which it is used. The minimum requirement of this subroutine is to CALL EXIT.

1. Subroutine Name: CRASH

2. Purpose:

   To display a full page message, "ERROR HALT", indicating that job has terminated abnormally.

3. Equations and Procedures: Self-explanatory

4. Calling Sequence: CALL CRASH

5. Storage Required: 884 bytes (221 words)

6. Subroutine User: CMINV, ASSPOT, DIAPH3, DATA, INPUT

7. Subroutine Required: None

8. Remarks: None

1. Subroutine Name: DATA

2. Purpose:

   Read, sort and store all input data. Perform tests where possible and track input errors. Display input data.

3. Equations and Procedures: Self-explanatory

4. Input Arguments:

   IW = kode (1 or 2) to identify call number

5. Output Arguments:

   XYZ = grid point coordinate data
   IBLN = element Boolean
   XMOD = mode shape data
   LND = leading edge grid points
   TITLE = work space for storing title cards

6. Error Returns: KONTRL(99) > 0 means stop

7. Calling Sequence:

   Call DATA (XYZ,IBLN,XMOD,LND,TITLE,IW)

8. Input Tapes: TYSIN (temporary card images)

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: 3462 bytes (866 words)

12. Subroutine User: INPUT

13. Subroutine Required: CRASH, EXIT

14. Remarks: None

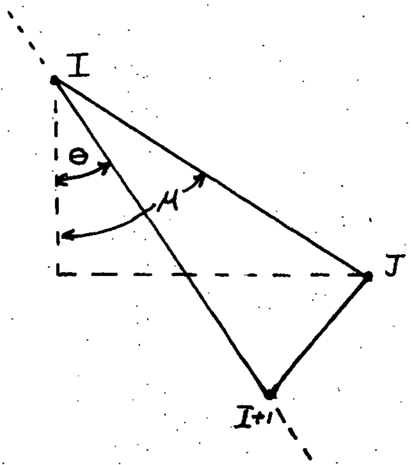
1. Subroutine Name: DIAPH3

2. Purpose:

Generate grid point coordinates and element Boolean for the diaphragm region.

3. Equations and Procedures:

Each new grid point is generated from a pair of corresponding points along the leading edge. Since the angle θ and u are known, the length IJ can be computed. It is then a simple matter to compute the X,Y coordinates of J.



4. Input Arguments:

XY = X,Y coordinates array
IBLN = existing Boolean (element)
LND = leading edge grid point numbers
IAR = work space
NLND = number of LND
NEL = number of elements on wing
NGP = number of grid points on
NDIV = number of divisions along mach line
MACH = mach number
MU = mach angle
OUT = SYSOUT value
PI = PI

5. Output Arguments:

LDNW = mach line grid points (excluding lead point)
NLDNW = number of LDNW
NEL = total number of elements
NGP = total number of grid points
NERR = error condition

6. Error Returns:

NERR > 0 implies negative grid points generated; either logic has failed or input data in error.

7. Calling Sequence:

```
Call DIAPH3 (XY,IBLN,LND,LDNW,IAR,NLND,NLDNW,NEL,NGP,
             NELD1,NDIV,MACH,MU,OUT,PI,NERR)
```

8. Storage Required:

3462 bytes (866 words)

9. Subroutine User:

INPUT

10. Subroutine Required:

CRASH, CORED

11. Remarks:

None

1. Subroutine Name: DISPLA

2. Purpose:

   Display a portion of fixed length complex array.

3. Equations and Procedures:

   That portion of the array to be displayed is transferred to a fixed length array and then the standard complex matrix printout is called.

4. Input Arguments:

   INDWN = complex array, (100, 6), to be displayed
   WORK = complex array, (NP,NMOD) work space
   NP1 = not used
   NP2 = not used
   NP = NP1 + NP2 = number of rows of INDWN to be displayed
   NMOD = nbr. of columns of INDWN to be displayed

5. Output Arguments: None

6. Error Returns: None

7. Calling Sequence:

   CALL DISPLA (INDWN,WORK,NP1,NP2,NP,NMOD)

8. Storage Required: 690 bytes (173 words)

9. Subroutine User: MAIN

10. Subroutine Required: CMATPR

11. Remarks: None

1. Subroutine Name: DWASHY

2. Purpose: To evaluate the following integral across a triangular element using Gaussian quadrature technique.

$$I = -\Omega_\eta \int \left[\frac{e^{-i\hat{k}(x_r - \xi_u)}}{(X_r - \xi_u)} \Psi(\xi_u , \eta) - \frac{e^{-i\hat{k}(x_r - \xi_L)}}{(X_r - \xi_L)} \Psi(\xi_L , \eta) \right] d\eta$$

3. Equations and Procedures:

The equations and procedures required to compute the above integral are the same as those used in Subroutine TRINTX. The only difference here is that this is a single integral.

4. Input Arguments:

Y1, Y2, Y3 = ordered set of Y-coord values of the triangle (smallest to largest)

Z = Z-coord of triangle (Z1+Z2+Z3)/3

A1, A2, A3 = slopes of sides of triangle

B1, B2, B3 = Y-intercept of slopes

Case = (1 or 2) relative location of the base of triangle

5. Output Arguments: None

6. Error Returns: None

7. Calling Sequence:

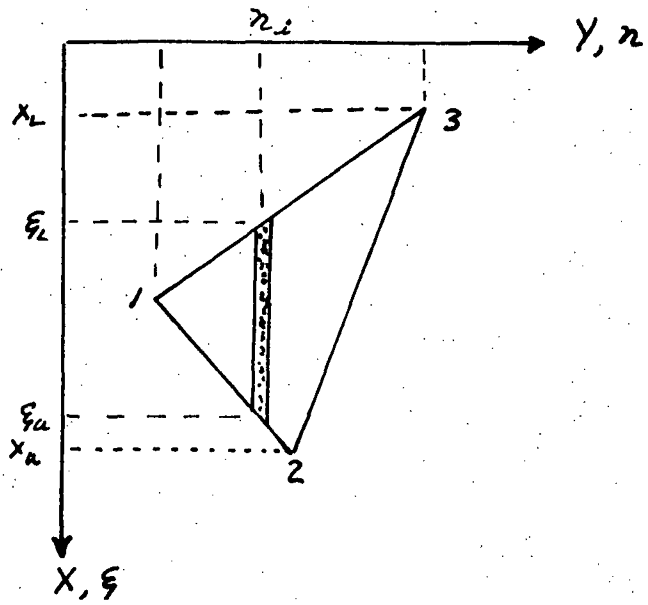
CALL DWASHY (Y1, Y2, Y3, A12, A13, A23, B12, B13, B23, CASE, Z)

8. Storage Required: 2970 bytes (743 words)

9. Subroutine User: MASTER

10. Subroutine Required: PSI

11. Remarks:



A typical element is shown here to indicate what points are chosen for integration.

1. Subroutine Name: ENDJOB

2. Purpose:

   Display a full page title stating that program has terminated normally.

3. Equations and Procedures: Self-explanatory

4. Input Arguments: None

5. Output Arguments: None

6. Error Returns: None

7. Calling Sequence: Call ENDJOB

8. Storage Required: 820 bytes (205 words)

9. Subroutine User: MAIN

10. Subroutine Required: None

11. Remarks:

    Page centering is accomplished by adjustment of the prefix N in line 1 of the format statement 1132, e.g. (... N(1H /) ...)

1. Subroutine Name: INPUT

2. Purpose:

   Read input stream and store on temporary data set. Display card images as read. Control reading and storing (Sub.DATA) of data and diaphragm generation (Sub.DIAPH3).

3. Equations and Procedures: Self-explanatory

4. Input Arguments: None

5. Output Arguments:

XYZ	=	grid point coordinate data
IBLN	=	element Boolean
XMOD	=	mode shape data
LND	=	leading edge grid point numbers
MLINE	=	mach line grid point numbers
TITLE	=	work space

6. Error Returns: None

7. Calling Sequence:

   Call INPUT (XYZ,IBLN,XMOD,LND,MLINE,TITLE)

8. Scratch Tapes: TYSIN = TYSOUT

9. Storage Required: 4020 bytes (1005 words)

10. Subroutine User: MAIN

11. Subroutine Required:

    CRASH
    DATA
    DIAPH3

12. Remarks: None

1. Subroutine Name: FAKTOR

2. Purpose: To compute the total number of elements having a common node point (wing only) and return this value as a weighing factor for each node point.

3. Equations and Procedures:

   Each element contains three node points stored in an sequential array. It thus requires only the comparison of a given node point number, say i, to all node points in the list. The number of appearances is counted and the reciprocal of this value is the weight factor.

4. Input Arguments:

   IBLN = Boolean array of node points vrs element no.

   NEL = Total number of elements (wing only)

   NGP = Total number of grid points (wing only)

5. Output Arguments:

   FAC = Weight factor per node point

6. Error Returns: None

7. Calling Sequence:

   CALL FAKTOR (FAC,IBLN,NFL,NGP)

8. Input Tapes: None

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required:

    552 bytes (138 words)

12. Subroutine User: MASTER

13. Subroutine Required: None

14. Remarks: None

1. Subroutine Name: MASTER

2. Purpose: This routine is the master logic control program for generation of element to element interaction data.

3. Equations and Procedures:

The logical flow of this routine is shown in the accompanying flow chart. In addition to this flow chart, pressure matrices and input downwash matrices are computed as follows:

Pressure matrices for the basic elements are:

$$[A] = [\tilde{A}] + i\kappa[\tilde{B}] \qquad (3 \times 3)$$

$$[\tilde{A}] = \frac{|2\Delta|}{6} \cdot \{e\} \cdot \Omega_x$$

$$[\tilde{B}] = \frac{|2\Delta|}{24} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

$$2\Delta = \det \begin{bmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\Omega_x = \frac{1}{2\Delta}\left[(y_2-y_3),\ (y_3-y_1),\ (y_1-y_2)\right]$$

$$\{e\} = \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}$$

Input downwash, at each node

$$\left(\frac{w}{U}\right) = \left[\Omega_x + i\kappa\, \Omega\right] \rho$$

4. Input Arguments:

XMOD	=	mode shape data
XYZ	=	coordinate data
MLINE	=	mach line node points
IBLN	=	element boolean array
LND	=	leading edge node points

5. Output Arguments:

INDWN	=	input downwash array

6. Error Returns: None

7. Calling Sequence:

```
CALL MASTER (INDWN, XMOD, XYZ, MLINE, IBLN, PM, PTUSED, LND)
```

8. Input Tapes: None

9. Output Tapes:

NTAPE	=	(Unit=1)	, velocity potentials
MTAPE	=	(Unit=3)	, downwash potentials
LTAPE	=	(Unit=2)	, pressure matrices

10. Scratch Tapes:

11. Storage Required:

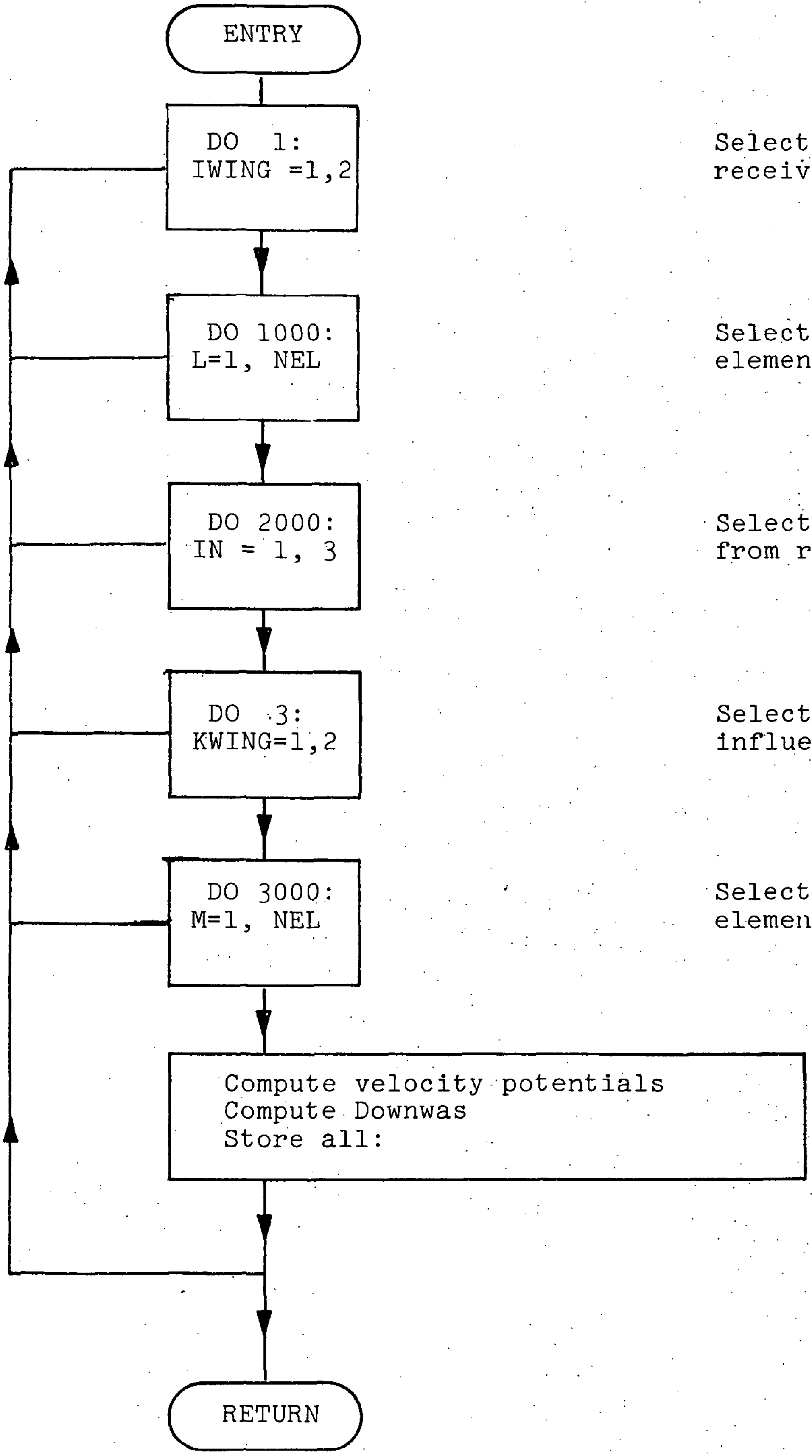
7428 bytes, (1857 words)

12. Subroutine User: MAIN

13. Subroutine Required:

FAKTOR , DWASHY , TRINTX
ORIENT , TRINOUT

14. Remarks: None



Select wing for
receiving elements

Select receiving
element from IWING

Select 1 node point
from receiving element

Select wing for
influencing element

Select influencing
element on KWING

1. Subroutine Name   ORIENT

2. Purpose:

   To determine the orientation of a 2-dimensional triangle in space and compute the slopes and intercepts of all 3 sides.

3. Equations and Procedures:

   a) arbitrarily choose an order for starting

   b) through a logical comparison determine the new order

   c) so ordered, the slopes and intercepts are computed by $y = mx + b$

4. Input Arguments:

   X = array of 3 -x coordinates

   Y = array of 3 -y coordinates

5. Output Arguments:

   X1, X2, X3 = ordered x-values, low to high

   Y1, Y2, Y3 = corresponding y values

   A12,A13,A23 = slopes of lines

   B12,B13,B23 = intercepts of lines

   CASE = normally 1 but set = 2 if base of triangle is parallel to y-axis

6. Error Returns: None

7. Calling Sequence:

```
CALL ORIENT ( X, Y, A12, A13, B12, B13, CASE,
              X1, X2, X3, A23, B23, Y1, Y2, Y3)
```

8. Subroutine User: Master

9. Subroutine Required: None

10, Remarks: None

1. Subroutine Name: OUTPUT

2. Purpose:

   Display the major results of the total program. Some final calculations are performed prior to display.

3. Equations and Procedures:

   [PRESSURES] = [PRESSURES INPUT] * [VELOCITY POTENTIALS]

   [CEN. FORCES] = [MODESHAPES] * [PRESSURES]

4. Input Arguments:

   PHI = velocity potentials (NP * NMOD)
   PM = work space (pressure matrix storage)
   XMOD = mode shapes (50 * 6 * 2)
   PRES = work space (final pressures)
   GENF = work space (generalized forces)
   NP = order of wing for display
   NMOD = Nbr. mode shapes used
   LTAPE = Unit definition for pressures
   HEY = (1 or 2) describes wing for display

5. Output Arguments: None

6. Error Returns: None

7. Calling Sequence:

   CALL OUTPUT (PHI,PM,XMOD,PRES,GENF,NP,NMOD,LTAPE,KEY)

8. Input Tapes:

   UNIT = LTAPE
   UNIT = TYSIN

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: 2090 bytes (525 words)

12. Subroutine User: MAIN

13. Subroutine Required: TINOUT, CMPRD, CMATPR

14. Remarks: COMMON/BASIC/KON(100)

1. Subroutine Name: PSI

2. Purpose: Evaluate the function described below.

3. Equations and Procedures:

$$\Psi(\mu,\nu) = J_0\left(\frac{\kappa\mu}{m}\right) \mathrm{Sin}^{-1}\left(\frac{\nu}{\mu}\right) + \sum^{N} \frac{(-1)^r}{r} J_{2r}\left(\frac{\kappa\mu}{m}\right) \mathrm{Sin}\left(2r\, \mathrm{Sin}^{-1}\left(\frac{\nu}{\mu}\right)\right)$$

where the J_n are Bessel functions of integer order. Sufficient terms are taken to insure error in the sum ≤ 0.0005.

4. Input Arguments:

XJ	=	χ-coordinate
YJ	=	Y-coordinate
Ƶ	=	Ƶ-coordinate

5. Output Arguments:

PSI = computed value $\Psi(u,v)$

6. Error Returns:

IER	Meaning
1	max terms in sum (N) was > 40
2	max terms in sum (N) was > 100
3	argument of Bessel function was negative
4	convergence criteria on sum was not met

All messages non-fatal. Final results should be suspect.

7. Calling Sequence:

Call PSI (XJ,YJ,Ƶ,SEI)

8. Storage Required:

3736 bytes (944 words)

1. Subroutine Name: SIZE

2. Purpose: To compute a) the starting locations of all variable length arrays used during analysis, b) test for maximum work space exceeding that alloted to program. If b is true, message is returned to calling program.

3. Equations and Procedures:

   Using input data concerning the current problem, all work space for each subroutine to be called is computed by successive additions of each array length to a grand total. Any array being shared by more than one subroutine is located near the top of the work space. When all space computed, grand totals are compared to the alloted maximums.

4. Input Arguments: KON = COMMON/BASIC/KONTRL,

   IS = contains maximum array sizes and maximum lengths of work space

   KEY = 1 or 2 for branching

5. Output Arguments: IS = contains starting locations of all work space arrays

6. Error Returns: If maximum size(s) exceeded, KON(100) = 0 is returned as an error flag.

7. Calling Sequence: CALL SIZE (KON, IS, KEY)

8. Storage Required: 1598 bytes (400 words)

9. Subroutine User: MAIN

10. Subroutine Required: None

1. Subroutine Name: STORE

2. Purpose:

   To insert a submatrix into a master array

3. Equations and Procedures:

   Program is capable of inserting elements as follows:
   (a = master array)

   a. $a(i,j) = b(k,l)$

   b. $a(i,j) = -b(k,l)$

   c. $a(i,j) = a(i,j) - b(k,l)$

   Each method is selected by a control variable "KODE" (see input argument list).

4. Input Arguments:

   WT = master array, complex
   W = submatrix, complex
   NP = row dimension of "WT"
   NN = row dimension of "W"
   M1 = starting subscript (rows) in "WT"
   L1 = starting subscript (cols) in "WT"
   M2 = stopping subscript (rows) in "WT"
   L2 = stopping subscript (cols) in "WT"
   KODE = select mode of storage

   1 = type a above
   2 = type b above
   3 = type c above

5. Output Arguments: None

6. Error Returns: None

7. Calling Sequence:

   Call STORE (WT, W, NP, NN, M1, L1, M2, L2, KODE)

8. Storage Required: 1390 bytes (347 words)

9. Subroutine User: ASSDWN

10. Subroutine Required: None

11. Remarks:

    It is assumed that before KODE = 3 is used that "WT" array contains valid data.

1. Subroutine Name: TAPES

2. Purpose:

   Special purpose subroutine to display 3 (threeee) tapes used in the AIC/INT program. These tapes store all element related information so generated, and are the basis of the RESTART phase of the program.

3. Equations and Procedures: Self-explanatory

4. Input Arguments:

   C = array of complex storage, 2500 words will handle the maximum case generated.

5. Output Arguments: None

6. Error Returns: None

7. Calling Sequence:

   CALL TAPES(C)

8. Input Tapes:

   UNIT =1
   UNIT =3
   UNIT =2

9. Output Tapes: None

10. Scratch Tapes: None

11. Storage Required: 984 bytes (246 words)

12. Subroutine User: MAIN

13. Subroutine Required: CMATPR, TINOUT

14. Remarks:

    This subroutine was designed specifically for "DEBUG" purposes.

    WARNING placement of call to this subroutine is critical, as all tapes are left "REWOUND". This is an abnormal situation at any point in program except the end.

1. Subroutine Name: TINOUT

2. Purpose:

Read/write a complex array of order (L,M) from/to tape.

3. Equations and Procedures:

Each array is placed on tape as follows:
a. Identification header is written first;
b. Array follows, stored by columns;
c. Identification trailer is written last;

HEADER:

Word	1	2	3	4	5	6	7	8 thru 14
	-10	L	M	N	A	M	E	0

TRAILER: Word 1 = -20,
Remainder record the same as header

4. Input Arguments:

X = array, complex, to be written to tape
L = row dimension of X
M = column dimension of X
NAME = 4 literal characters describing the name or array X
IO = control for action taken
KTAPE = unit where information will be read/written

5. Output Arguments:

X = array, complex, read from tape

6. Error Returns: None

7. Calling Sequence:

CALL TINOUT (X, L, M, NAME, IO, KTAPE)

8. Input Tapes: UNIT = KTAPE (IO=2)

9. Output Tapes: UNIT = KTAPE (IO=1)

10. Storage Required: 1722 bytes (431 words)

11. Subroutine User: MASTER, ASSPOT, ASSDWN, OUTPUT

12. Subroutine Required: CORED

13. Remarks:

Four error conditions may arise because of failure in other parts of the program (AIC/INT). These errors will cause termination of the program thru the call to CORED. All four errors involve failure to match input information (thru argument) with that read from the tape.

ERROR NBR	MEANING
1	First record not a header
2	Name of array requested does not match that found in header
3	Order of array requested disagrees with that found in header
4	Last record read was not a trailer

Any of these messages indicate the program logic has failed.

NAME COMMON: /TRINT/

The total length of this common block is 56 words. It appears in only four subroutines: MASTER, TRINTX, DWASHY, and PSI. The variables in the block (in order of appearance) are defined as follows:

Variable	Type	Dimension	Definition
FPOT	C	3	Velocity potentials
DWN	C	3	Downwash potentials, X-integration
DWNI	C	3	Downwash potentials, Y-integration
UU	R	5	Pivot points, Gaussian integration
CC	R	5	Weight Factors, Gaussian integration
XI	R	3	X-coord, influencing element
YI	R	3	Y-coord, influencing element
ZI	R	3	Z-coord, influencing element
T	R	9	Transformation, influencing element
XR	R	1	X-coord, receiving element
YR	R	1	Y-coord, receiving element
ZR	R	1	Z-coord, receiving element
XK	R	1	Frequency
ZMACH	R	1	Mach number
BEETA	R	1	beta
PIE	R	1	Pi
XKOM	R	1	Frequency / mach no.
USQRD	R	1	beta * beta
C1OPIV	R	1	1 / (pi * beta)

1. <u>Subroutine Name:</u> TRINTX

2. Purpose: To evaluate the following integrals across a triangular element using Gaussian quadrature technique.

$$I_1 = -\frac{1}{\pi}\int_{X_L}^{X_u} e^{-i\hat{k}(X_r-\xi)}\Big[\Omega(\xi,\eta_u)\Psi(\xi,\eta_u) - \Omega(\xi,\eta_L)\Psi(\xi,\eta)$$

$$-\Omega_\eta \int_{Y_L}^{Y_u} \Psi(\xi,\eta)\, d\eta\Big]\, d\xi$$

$$I_2 = -\int_{X_L}^{X_u} e^{-i\bar{k}(x_r-\xi)}\Big[\Omega_\xi\big(\Psi(\xi,\eta_L) - \Psi(\xi,\eta_u)\big) + \frac{1+i\vec{k}(x_r-\xi)}{(X_r-\xi)}$$

$$\big(\Omega(\xi,\eta_L)\Psi(\xi,\eta_L) - \Omega(\xi,\eta_u)\Psi(\xi,\eta_u)\big)\Big]\, d\xi$$

$$I_3 = \Omega_\eta \int_{X_L}^{X_u}\int_{Y_L}^{Y_u}\Psi(\xi,\eta)d\eta\, \frac{e^{-i\vec{k}(x_r-\xi)}}{(X_r-\xi)}\left(\frac{1+i\vec{k}(x_r-\xi)}{(X_r-\xi)}\right) d\xi$$

where ξ is the X-coord variable, η the Y-coord variable and Xr is the X-coord reference point (called the receiving point in formulation)

3. <u>Equations and Procedures:</u>

In the above equations, Ω is a vector and is written (in rectangular coordinates) as

$$\Omega = [X, Y, 1]\,[T]$$

where the area matrix T is given by

$$T = \begin{bmatrix} X_1 & X_2 & X_3 \\ Y_1 & Y_2 & Y_3 \\ 1 & 1 & 1 \end{bmatrix}^{-1}$$

The X_i, Y_i relate to the node points of the triangle in question. If the determinant of T is 1E-06, no calculations are made, and the value of zero is returned for the integral. A 5-point Gaussian integration technique is then applied as a numerical procedure. Checks are made during each step on the upper and lower limits to ascertain that the total strip area remains within the mach cone area (i.e. it may be a partial element configuration). Since Ω is a 1*3 array, the potentials returned are stored in a 1*3 array (complex). When possible, calculations are made in real numbers to computation time. A branch is provided to bypass calculation of integral b (above) when necessary.

4. Input Arguments:

X1, X2, X3 = ordered set of X-coord values of the triangle (smallest to largest)

Z = Z-coord of triangle (Z1+Z2+Z3/3)

A12, A13, A23 = slopes of sides of triangle

B12, B13, B23 = X-intercept of slopes

CASE = (1 or 2) relative location of the base of triangle

5. Output Arguments: None

6. Error Returns: None

7. Calling Sequence:

CALL TRINTX (X1, X2, X3, A12, A13, A23, B12, B13, CASE, Z)

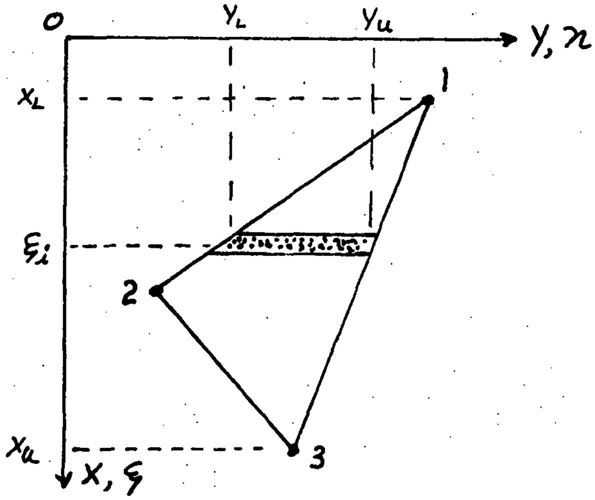
8. Storage Required: 3956 bytes (989 words)

9. Subroutine User: MASTER

10. Subroutine Required: PSI

11. Remarks:

A typical element is shown here to indicate what points are chosen for integration purposes



DECK 001

```
      COMPLEX    WCRC  ( 16000  )                                             00100010
     0INTEGER    ISTART(   28    ) ,RESTAR            ,SYSIN                  00100020
     1          ,SYSCLT             ,TYSIN            ,TYSOUT                 00100030
      DIMENSICN  WCRK  (  2650  ) ,RECORD(    14    )                         00100040
      COMMON     /BASIC / KCNTRL(100)                                         00100050
      COMMON     /CIMENS/ N1, N2, N3, N4, N5, N6, N7, N8,                     00100060
     *                    J1, J2, J3,                                         00100070
     *                    M1, M2, M3, M4, M5, M6,                             00100080
     *                    L1, L2, L3, L4, L5, L6, L7, L8, L9                  00100090
      EQUIVALENCE                                                             00100100
     *            (KCNTRL(46),   PI        )                                  00100110
     *           ,(KCNTRL(58),SYSOUT       )                                  00100120
     *           ,(KCNTRL(59),SYSIN        )                                  00100130
     *           ,(KCNTRL(60),TYSIN,TYSOUT)                                   00100140
     *           ,(KCNTRL(54),KTAPE        )                                  00100150
     *           ,(KCNTRL(55),LTAPE        )                                  00100160
     *           ,(KCNTRL(56),MTAPE        )                                  00100170
     *           ,(KCNTRL(57),NTAPE        )                                  00100180
     *           ,(KCNTRL(47),NP           )                                  00100190
     *           ,(KCNTRL( 5),NP1          )                                  00100200
     *           ,(KCNTRL(20),NP2          )                                  00100210
     *           ,(KCNTRL( 8),NC1          )                                  00100220
     *           ,(KCNTRL(23),NC2          )                                  00100230
     *           ,(KCNTRL( 4),NMCD         )                                  00100240
     *           ,(KCNTRL(86),RESTAR       )                                  00100250
     *           ,(KCNTRL(97),SUMEPS       )                                  00100260
     *           ,(RECCRD( 1),KCDE         )                                  00100270
     *           ,(ISTART( 1),N1           )                                  00100280
     *           ,(KAPPA      ,CAPPA       )                                  00100290
      DATA    IZR,IM20 / 0, -20 /                                             00100300
C                                                                             00100310
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  00100320
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  00100330
C                                                                             00100340
C     THIS  PROGRAM  EVCLVED  AT  BELL AEROSPACE CO.  UNDER  THE  TITLE       00100350
C      OF  'DEVELCPMENT  AND  APPLICATIONS  OF  SUPERSONIC  UNSTEADY          00100360
C     CONSISTENT  AERODYNAMICS  FCR  INTERFERING  PARALLEL  WINGS'            00100370
C     (BAC  REPCRT  NC. 2471-941001,    CONTRACT  NAS1 - 10880 (03/72) )      00100380
C                                                                             00100390
C                                                                             00100400
C                  MNEMONIC      ' A I C  / I N T '                           00100410
C                                                                             00100420
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  00100430
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  00100440
C                                                                             00100450
C        DEFINE  ALL  CCNTRLS.  TAPE  UNITS,  MAXIMUMS,  PI                   00100460
C                                                                             00100470
      DO 6 JP=1,100                                                           00100480
    6 KCNTRL(JP) = 0                                                          00100490
C                                                                             00100500
      ISTART(11) = 100                                                        00100510
      ISTART(12) =  20                                                        00100520
      ISTART(13) =  10                                                        00100530
      ISTART(14) =  50                                                        00100540
      ISTART(15) =  80                                                        00100550
      ISTART(27) = 16000                                                00100580
      ISTART(28) = 2650                                                 00100590
C                                                                       00100600
      PI          = 3.141592654                                         00100610
      SYSOUT      = 6                                                   00100620
      SYSIN       = 5                                                   00100630
      TYSOUT      = 9                                                   00100640
      KONTRL(54) = 4                                                    00100650
      KONTRL(55) = 2                                                    00100660
      KONTRL(56) = 3                                                    00100670
      KONTRL(57) = 1                                                    00100680
      KONTRL(84) = 7                                                    00100690
      MREST       = 8                                                   00100700
      SUMEPS      = 1.00E-04                                            00100710
C                                                                       00100720
C-----------------------------------------------------------------------00100730
C                                                                       00100740
      KEY     = 1                                                       00100750
      CALL  SIZE( KONTRL , ISTART , KEY )                               00100760
C                                                                       00100770
      CALL  INPUT                                                       00100780
     *    ( WCRK(N2), WORK(N3), WORK(N1), WORK(N5), WORK(N4)            00100790
     *      ,WCRK(N6) )                                                 00100800
C                                                                       00100810
      KEY     = 2                                                       00100820
      CALL  SIZE( KONTRL , ISTART , KEY )                               00100830
C                                                                       00100840
C-----------------------------------------------------------------------00100850
C                                                                       00100860
      IF( KCNTRL(100)  .LE. 0 ) CALL  EXIT                              00100870
C                                                                       00100880
  555 READ(SYSIN,1000) CAPPA                                            00100890
         IF( ENDFILE SYSIN )  900,556                                   00100900
  556 KONTRL(17) = KAPPA                                                00100910
      KONTRL( 2) = KAPPA                                                00100920
C                                                                       00100930
      REWIND  KTAPE                                                     00100940
      REWIND  LTAPE                                                     00100950
      REWIND  MTAPE                                                     00100960
      REWIND  NTAPE                                                     00100970
      CALL  MASTER                                                      00100980
     *    (WORC(J1), WORK(N1), WORK(N2), WORK(N4),                      00100990
     *     WORK(N3), WORC(J2), WORK(N7) , WORK(N5) )                    00101000
      IERR  = KCNTRL(99)                                                00101010
         IF( IERR  .GT.  0 )  WRITE(SYSOUT,9002)  IERR                  00101020
C                                                                       00101030
         IF(  RESTAR .GE. 0  )  GOTO 3006                               00101040
C-----------------------------------------------------------------------00101050
C........READ  'MTAPE'  THRU  2-ND  '-20'  AND  PARK.                   00101060
C        IMPORTANT  THAT  TAPE  BE  PARKED  AT END  FOR  'ASSPOT'       00101070
C          RESTART  TECHNIQUE  MUST  TRANSFER  INFO  ON  'MTAPE'  TO    00101080
C          ANOTHER  UNIT.    THIS  PROTECTS  ANY  FOLLOWIN  FILES       00101090
C          BECAUSE  OF   'WRITE(MTAPE)  IN   ASSPOT, ASSDWN   ROUTINES  00101100
      REWIND  MREST                                                     00101110
      DO  3004  J=1,2                                                   00101120
 3005 READ(MTAPE         ) RECORD                                       00101130
      WRITE(MREST)  RECORD                                              00101140
      IF(KODE .NE. IM20 ) GC TO 3005                                    00101150
      REWIND  MTAPE                                                     00101170
      MTAPE  = MREST                                                    00101180
C                                                                       00101190
C-----------------------------------------------------------------------00101200
 3006 CONTINUE                                                          00101210
C                                                                       00101220
      CALL  ASSPCT                                                      00101230
     1     (WORC(M2) , WORC(M3) , WORC(M4) , WORC(M5)                   00101240
     2   ,   NP1     ,   NP2    ,   ND1    ,   ND2                      00101250
     3   ,  NTAPE    ,  KTAPE   ,  MTAPE   )                            00101260
      CALL  ASSDWN                                                      00101270
     1     (WORC(L2) , WORC(L3) , WORC(L4) , WORC(L5)                   00101280
     2     ,WORC(L8) , WORC(L6) , WORC(L1) ,   NP                       00101290
     3     , NP1     ,   NP2    ,   ND1    ,   ND2                      00101300
     4     ,KONTRL(50)          , MTAPE    , KTAPE                      00101310
     5     ,KONTRL( 4)          , KONTRL(49)                            00101320
     6     ,KONTRL(48)          , WORC(L7)            )                 00101330
C                                                                       00101340
      CALL  DISPLA( WORC,WORC(L4),NP1, NP2, NP , NMOD )                 00101350
      NO1      = L2                                                     00101360
      NO2      = L4                                                     00101370
      NO3      = NO2 + NP1 * NP1                                        00101380
      NO4      = NO3 + NP1 * NMOD                                       00101390
      NO5      = NO4 + 36                                               00101400
      KEY      = 1                                                      00101410
      CALL  OUTPLT                                                      00101420
     1     ( WORC(NO1) , WORC(NO2) , WORK( N1) , WORC(NO3)              00101430
     2     , WORC(NO4) ,    NP1,NMOD,  LTAPE    ,    KEY        )       00101440
C                                                                       00101450
      NO1      = L3                                                     00101460
      NO3      = NO2 + NP2 * NP2                                        00101470
      NO4      = NO3 + NP2 * NMOD                                       00101480
      NO5      = NO4 + 36                                               00101490
      KEY      = 2                                                      00101500
      CALL  OUTPLT                                                      00101510
     1     ( WORC(NO1) , WORC(NO2) , WORK( N1) , WORC(NO3)              00101520
     2     , WORC(NO4) ,    NP2,NMOD,  LTAPE    ,    KEY        )       00101530
C                                                                       00101540
         IF( KONTRL(70) .GT. 1 )  CALL  TAPES( WORC(L4) )               00101550
        GO TO 555                                                       00101560
  900 CALL  ENDJOB                                                      00101570
      CALL  EXIT                                                        00101580
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 00101590
1000  FORMAT( 10X, 4E15.6 )                                             00101600
9002 0FORMAT(1H1 ////   20X,24H*** W A R N I N G ***       //           00101610
     1        15X, I5, 44H  ERRORS  ACCUMULATED  DURING  INTEGRATION.   )00101620
      END                                                               00101630
```

DECK 002

```
      OSUBROUTINE  INPUT                                                 00200010
      *   ( XYZ, IBLN, XMOD, LND, MLINE, TITLE )                         00200020
      OREAL        MACH          ,XYZ   ( 3,80,2 ) ,XMOD  (50,10,2 )     00200030
      1           ,TITLE (  20,1  )                                      00200040
      OINTEGER     SYSIN         ,SYSOUT           ,TYSIN                00200050
      1           ,TYSOLT        ,KARD  (   20   ) ,KCOL  (    5   )     00200060
      2           ,IBLN  ( 3,100,2) ,LND (  20,2  ) ,MLINE (  20,2  )    00200070
       COMMON     /    BASIC     /  KONTRL(  100    )                    00200080
       EQUIVALENCE                                                       00200090
      1            (KCNTRL(46),   PI         )                           00200100
      2           ,(KCNTRL(58),SYSOUT        )                           00200110
      3           ,(KCNTRL(59),SYSIN         )                           00200120
      4           ,(KCNTRL(60),TYSIN,TYSOUT)                             00200130
      6           ,(KCNTRL(87),MACH          )                           00200140
      7           ,(KCNTRL(88),VEETA         )                           00200150
      8           ,(KCNTRL(90),ISCLAT        )                           00200160
      9           ,(KCNTRL(91),NEWPTS        )                           00200170
      A           ,(KCNTRL(92),XMU           )                           00200180
      B           ,(KCNTRL(14),THETA1        )                           00200190
      C           ,(KCNTRL(29),THETA2        )                           00200200
      ODATA        KCCL          /4H1234          ,4H5678                00200210
      1           ,4H9012        ,4H3456          ,4H7890        /       00200220
       DATA        KEND          /4HENDA      /                          00200230
C******                                                                  00200240
C         READ  ALL  CARD  IMAGES  AND  STORE  ON  'TYSOUT'.   DISPLAY   00200250
C         AS  GO.    SEARCHING  FOR  ''ENDATA''  CARD  IN  INPUT  STREAM.00200260
C         THIS  DEFINES  E-O-F .                                         00200270
C         (NOTE.  LINCT  IS  VARIABLE  BY  USER  TO  GET  MAXIMUM  USE   00200280
C          OF  PAGE.   IT  MUST  BE  CHANGED  IN  2 PLACES.              00200290
C******                                                                  00200300
C                                                                        00200310
       REWIND   TYSIN                                                    00200320
       WRITE(SYSCLT,1080)                                                00200330
        WRITE(SYSCLT,1020)   ((KCOL(K),K=1,5),KK=1,4)                    00200340
       LINCT = 38                                                        00200350
       ILINE =  0                                                        00200360
     1 READ (SYSIN ,1010       )  KARD                                   00200370
         IF( ENDFILE SYSIN ) 8,9123                                      00200380
  9123 CONTINUE                                                          00200390
       WRITE(TYSCLT,1010)           KARD                                 00200400
       WRITE(SYSCLT,1020)           KARD                                 00200410
       ILINE = ILINE + 1                                                 00200420
         IF( KARD(1) .EQ. KEND )  GO TO 2                                00200430
         IF(ILINE .LT. LINCT)  GO TO 1                                   00200440
       LINCT = LINCT + 38                                                00200450
       WRITE(SYSCLT,1020)    ((KCOL(K),K=1,5),KK=1,4)                    00200460
       WRITE(SYSCLT,1080)                                                00200470
       WRITE(SYSCLT,1020)    ((KCCL(K),K=1,5),KK=1,4)                    00200480
         GO TC 1                                                         00200490
     8 CALL  CRASH                                                       00200500
       WRITE(SYSCLT,1091)                                                00200510
       CALL  EXIT                                                        00200520
     2 END FILE TYSIN                                                    00200530
       REWIND   TYSIN                                                    00200540
        WRITE(SYSCLT,1020)   ((KCOL(K),K=1,5),KK=1,4)                    00200550
```

```
C                                                                       00200570
      IW = 1                                                            00200580
    3 CONTINUE                                                          00200590
      CALL  DATA                                                        00200600
     1      ( XYZ(1,1,IW), IBLN(1,1,IW), XMOD(1,1,IW)                   00200610
     2         ,LND(1,IW), TITLE, IW                                    00200620
     3         )                                                        00200630
      GO TO( 4,5 ), IW                                                  00200640
    4 IW = 2                                                            00200650
      GO TO 3                                                           00200660
    5 CONTINUE                                                          00200670
         IF( KONTRL(99)  .LE.  0 )  GO TO 6                             00200680
C        SERIOUS  ERRORS  WERE  DETECTED.   MUST  STOP                  00200690
      CALL  CRASH                                                       00200700
      CALL  EXIT                                                        00200710
    6 CONTINUE                                                          00200720
C                                                                       00200730
      VEETA = SQRT(MACH*MACH - 1.00)                                    00200740
      XMU =  ASIN(1.00 / MACH)                                          00200750
      I1     = LND(1,1)                                                 00200760
      I2     = LND(2,1)                                                 00200770
      THETA1 = ATAN((XYZ(2,I2,1) - XYZ(2,I1,1))/                        00200780
     *              ( XYZ(1,I2,1) - XYZ(1,I1,1)))                       00200790
      I1     = LND(1,2)                                                 00200800
      I2     = LND(2,2)                                                 00200810
      THETA2 = ATAN((XYZ(2,I2,2) - XYZ(2,I1,2))/                        00200820
     *              ( XYZ(1,I2,2) - XYZ(1,I1,2)))                       00200830
C                                                                       00200840
C.........PLACE  TITLE  CARDS  ON  DISK  FOR  LATER  DISPLAY            00200850
C                                                                       00200860
      REWIND  TYSIN                                                     00200870
      NT  = KONTRL(89)                                                  00200880
      IF( NT .LE. 0 )  GO TO 10                                         00200890
      DO 20 J=1,NT                                                      00200900
      WRITE(TYSOUT,1010)  (TITLE(JJ,J),JJ=1,20)                         00200910
   20 CONTINUE                                                          00200920
      END FILE TYSIN                                                    00200930
      REWIND   TYSIN                                                    00200940
C                                                                       00200950
   10 CONTINUE                                                          00200960
      IW = 1                                                            00200970
      MB = 0                                                            00200980
      NGP = KONTRL(MB+5)                                                00200990
       IF( XMU .LE. THETA1 ) GO TO 30                                   00201000
      WRITE(SYSOUT,2000)                                                00201010
12    CALL  DIAPH3                                                      00201020
     *     ( XYZ(1,1,IW) ,  IBLN(1,1,IW) ,  LND(1,IW)                   00201030
     *     , MLINE(1,IW) ,   TITLE                                      00201040
     *     ,KONTRL(MB+13),KONTRL(MB+3)    , KONTRL(MB+1)                00201050
     *     , NGP  ,        KONTRL(MB+9)    , KONTRL(MB+11)              00201060
     *     ,MACH,  XMU,  SYSOUT, PI, IERR                               00201070
     *      )                                                           00201080
      KONTRL(MB+1) = KONTRL(MB+6) + KONTRL(MB+9)                        00201090
      KONTRL(MB+7) = NGP                                                00201100
      KONTRL(MB+8) = NGP - KONTRL(MB+5)                                 00201110
C                                                                       00201120
      GO TO( 30,40 ), IW                                                00201130
30    CONTINUE                                                          00201140
      MB = 15                                                           00201150
      IF( XMU .LE. THETAZ ) GO TO 40                                    00201170
      WRITE(SYSOUT,2002)                                                00201180
      NGP=KONTRL(MB+5)                                                  00201190
       IF(ISOLAT .EQ. 0)  GO TO 12                                      00201200
C******                                                                 00201210
C         THIS  SECTION  HANDLES  'EXTRAPOLATION'  OF  LEADING-EDGE     00201220
C         ON  WING 2.                                                   00201230
C******                                                                 00201240
      WRITE(SYSOUT,2004)                                                00201250
C                                                                       00201260
      NLND    = KONTRL(MB+13)                                           00201270
      DO 32 J=1,NLND                                                    00201280
32    KARD(J) = LND(J,2)                                                00201290
C                                                                       00201300
C     COMPUTE  NEW  POINT  BY INTERSECTION  OF  MACH CONE WITH          00201310
C     Z-PLANE  OF  WING-2.  DIVIDE  EQUALLY  BY  NEWPTS                 00201320
C                                                                       00201330
      Z    = XYZ(3,1,2)                                                 00201340
      XOLD = XYZ(1,1,2)                                                 00201350
      XNEW = Z / TAN(XMU)                                               00201360
      DELX = (XOLD - XNEW) / FLOAT(NEWPTS)                              00201370
C                                                                       00201380
      JNEW = KONTRL(20)                                                 00201390
      DO 34 J=1,NEWPTS                                                  00201400
      JNEW = JNEW + 1                                                   00201410
      XYZ(1,JNEW,2) = XNEW                                              00201420
      XNEW = XNEW + DELX                                                00201430
      XYZ(2,JNEW,2) = 0.0                                               00201440
      XYZ(3,JNEW,2) =  Z                                                00201450
      WRITE(SYSOUT,2006) JNEW, XYZ(1,JNEW,2), XYZ(2,JNEW,2), Z          00201460
34    LND(J,2) = JNEW                                                   00201470
      JR    = NEWPTS + 1                                                00201480
      JS    = NLND + NEWPTS                                             00201490
      KONTRL(28) = JS                                                   00201500
      J1    = 0                                                         00201510
      DO 36 J=JR,JS                                                     00201520
      J1    = J1 + 1                                                    00201530
36    LND(J,2) = KARD(J1)                                               00201540
      NGP = NGP + NEWPTS                                                00201550
      WRITE(SYSOUT,2008)(LND(J,2),J=1,JS)                               00201560
      GO TO 12                                                          00201570
C-----------------------------------------------------------------------00201580
C                                                                       00201590
  40  CONTINUE                                                          00201600
      NP1 = KONTRL(5)                                                   00201610
      NP2 = KONTRL(20)                                                  00201620
      KONTRL(47) = NP1 + NP2                                            00201630
      KONTRL(48) = KONTRL( 7)                                           00201640
      KONTRL(49) = KONTRL(22)                                           00201650
      KONTRL(50) = KONTRL( 8)+ KONTRL(23)                               00201660
      KONTRL(51) =(NP1 +NP2)**2                                         00201670
      KONTRL(52) = KONTRL(50) **2                                       00201680
      KONTRL(53) = KONTRL(47) * KONTRL(50)                              00201690
      RETURN                                                            00201700
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .00201710
1010  FORMAT( 20A4)                                                     00201720
1020  FORMAT(10X,20A4)                                                  00201730
1080  FORMAT(1H1/// 28X,44HA I C   /   I N T......D A T A     D E C K )00201740
1091 CFORMAT(// 10X,  40HAN  END-OF-FILE  IN  THE  INPUT  STREAM    /   00201750
                      [illegible] AN  END  CARD  IS  MISSING.  '     00201770
2000  FORMAT(1H1///24X,20HW I N G....O N E     )                 00201780
2002  FORMAT(1H1///24X,20HW I N G....T W O     )                 00201790
2004  FORMAT(1H0,  20X,24HUPDATED  LEADING  DATA. /              00201800
     1              9X, 12HGRID  POINT   , 6X,1HX,19X,1HY,19X,1HZ )  00201810
2006  FORMAT(       11X, I5, 4E20.6  )                           00201820
2008  FORMAT(1H0,  20X,24HUPDATED  LEDGE  DATA.   /15X,20I5 )    00201830
      END                                                        00201840
      SUBROUTINE  DATA                                                  00300010
     *     ( XYZ, IBLN, XMOD, LND, TITLE, IW  )                         00300020
C                                                                       00300030
      INTEGER                                                           00300040
     1            SYSOUT           ,TYSIN            ,KARD  (   20   )  00300050
     2            ,IBLN   (  3,1   ) ,ICONT (    9    ) ,IPRINT(   12   ) 00300060
     3            ,LND    (   1    ) ,TITLE ( 20,1    )                  00300070
      DIMENSION  XYZ    (  3,1   ) ,XMOD  ( 50,1    ) ,HDR   (    9   ) 00300080
      COMMON     /    BASIC      /  KONTRL(  100    )                   00300090
      EQUIVALENCE                                                       00300100
     1            (KONTRL(58),SYSOUT)                                   00300110
     2            ,(KONTRL(60),TYSIN )                                  00300120
     3            ,(KONTRL(61),IPRINT(1))                               00300130
     *            ,(KONTRL(99),NERR   )                                 00300140
     4            ,(SHIFT      ,ISHIFT)                                 00300150
     5            ,(CHCRD      ,IREF  )                                 00300160
     6            ,(SYNC       ,ISYNC )                                 00300170
     7            ,(CAPPA      ,KAPPA )                                 00300180
     8            ,(XMACH      ,MACH  )                                 00300190
     9            ,(REPE       ,IYES  )                                 00300200
      DATA        NAMES /   9    / ,BLANK / 4H      / ,IYES  / 4HREPE / 00300210
     1            ,HDR   / 4HRUN    ,       4HTITL    ,        4HPRIN     00300220
     2            ,        4HSYST   ,       4HCOOR    ,        4HELEM     00300230
     3            ,        4HLEDG   ,       4HMODE    ,        4HEND /    00300240
C                                                                       00300250
C******                                                                 00300260
C        SET  INITIAL  CONSTANTS/CONTROLS.   'MB'  IS  A  BASE  FOR     00300270
C        'KONTRL'  ARRAY.    'ICONT'  TRACKS  LABEL-DATA  (0=NO, 1=YES).00300280
C******                                                                 00300290
      [illegible]                                                       00300320
      IF( IW .LE. 2 )  MB = 15                                          00300330
      DO  3  K=4,NAMES                                                  00300340
    3 ICONT(K) =  0                                                     00300350
      ICONT(1) = -1                                                     00300360
      ICONT(2) = -1                                                     00300370
      ICONT(3) = -1                                                     00300380
C******                                                                 00300390
C        ALL  LABEL-DATA  SECTIONS  RETURN  CONTROL  HERE.              00300400
C        A  LABEL-DATA-CARD  IS  ALWAYS  EXPECTED.                      00300410
C******                                                                 00300420
    5 READ (TYSIN ,1030) CHAR, CUM, NBR                                 00300430
        IF( ENDFILE  TYSIN )  298,505                                   00300440
  505 CONTINUE                                                          00300450
      DO  6  K=1,NAMES                                                  00300460
      KBR     = K                                                       00300470
      IF( HDR(K) .EQ. CHAR ) GO TO 9                                    00300480
    6 CONTINUE                                                          00300490
C******                                                                 00300500
C        INPUT  ERROR.    EITHER  CARD  IS  1. MISPELLED  OR  2. OUT OF 00300510
C        ORDER.    IN  EITHER  CASE  AN  ATTEMPT  TO  MATCH  HAS  FAILED00300520
C        AND  THEREFORE  WE  MUST  STOP  PROCESSING.                    00300530
C******                                                                 00300540
      BACKSPACE  TYSIN                                                  00300550
```

```
C******                                                                 00300580
C        RECORD  ERROR  TC  STOP  EXECUTION  LATER.     FLUSH  INPUT-   00300590
C        STREAM  TC  'END'.    CONTINUE  NEXT  BLOCK( WING).            00300600
C******                                                                 00300610
  299 CONTINLE                                                          00300620
      NERR    = IERR + NERR                                             00300630
   15 READ (TYSIN ,1030) CHAR                                           00300640
        IF( ENDFILE  TYSIN )  298,16                                    00300650
   16 CONTINUE                                                          00300660
        IF( CHAR .NE. HDR(9) )  GO TO 15                                00300670
      GO TO 99                                                          00300680
  298 CALL CRASH                                                        00300690
      WRITE(SYSCLT,1091)                                                00300700
      CALL  EXIT                                                        00300710
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 00300720
C                                                                       00300730
  9   GO TO( 10, 20, 30, 40, 13, 13, 13, 80, 90 ), KBR                  00300740
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 00300750
C                                                                       00300760
C******                                                                 00300770
C         LABEL  IS  RUN                                                00300780
C******                                                                 00300790
   10 ICONT( 1) = 1                                                     00300800
      KONTRL(100) = 1                                                   00300810
      KONTRL(E3) = NBR                                                  00300820
      KONTRL(E6) = NBR                                                  00300830
        GO TC 5                                                         00300840
C                                                                       00300850
C******                                                                 00300860
C        'SYSTEM'  CARD  MUST  APPEAR  BEFORE  ALL  NUMERICAL  DATA      00300870
C        SC  THAT   'NP'   MAY  BE  DEFINED.    ELSE  ALL-STOP.          00300880
C******                                                                 00300890
   13 K= KBR - 4                                                        00300900
      IF( ICONT(4) .EC. 0 ) GO TO 14                                    00300910
      GO TO( 50, 60, 70 ), K                                            00300920
C                                                                       00300930
   14 WRITE(SYSCLT,1099)                                                00300940
        GO TC 299                                                       00300950
C******                                                                 00300960
C         LABEL  IS  TITLE                                              00300970
C******                                                                 00300980
   20 ICONT( 2) = 1                                                     00300990
      K1 = KCNTRL(89)                                                   00301000
      K2 = K1 + NBR                                                     00301010
      KONTRL(E9) = K2                                                   00301020
      K1 = K1 + 1                                                       00301030
      DO 21 K=K1,K2                                                     00301040
      READ (TYSIN ,1010)  (TITLE(J,K),J=1,20)                           00301050
        IF( ENCFILE  TYSIN )  298,21                                    00301060
   21 CONTINUE                                                          00301070
        GC TO 5                                                         00301080
C******                                                                 00301090
C         LABEL  IS  PRINT                                              00301100
C******                                                                 00301110
   30 ICONT(3) = 1                                                      00301120
      GO TO( 31,32 ), IW                                                00301130
31    READ (TYSIN ,1040)  (IPRINT(K),K=1,12)                            00301140
        GO TO 5                                                         00301150
```

```
C******                                                                 00301180
C         LABEL  IS  SYSTEM                                             00301190
C******                                                                 00301200
   40 ICONT( 4) = 1                                                     00301210
      WRITE(SYSOUT,1005)                                                00301220
      READ (TYSIN ,1040)  NP, NELW, NMOD, NLND, NDIV, ISY, ISOLAT       00301230
     1                   ,NEWPTS                                        00301240
      READ (TYSIN ,1060)  XMACH, CAPPA, CHORD, SHIFT                    00301250
         IF( IW .EQ. 2 )  IREF  = KONTRL(12)                            00301260
         IF( IW .EQ. 2 )  SHIFT = SHIFT / CHORD                         00301270
      WRITE(SYSOUT,1045) IW,NP,NELW,NMOD,NLND,NDIV,ISY,CHORD            00301280
C                                                                       00301290
      SYMC  = FLOAT(ISY)                                                00301300
      KONTRL(MB+ 1) = NELW                                              00301310
      KONTRL(MB+ 7) = NP                                                00301320
      KONTRL(MB+ 2) = KAPPA                                             00301330
      KONTRL(MB+ 4) = NMOD                                              00301340
      KONTRL(MB+ 5) = NP                                                00301350
      KONTRL(MB+ 6) = NELW                                              00301360
      KONTRL(MB+10) = ISYMC                                             00301370
      KONTRL(MB+11) = NDIV                                              00301380
      KONTRL(MB+12) = IREF                                              00301390
      KONTRL(MB+13) = NLND                                              00301400
       IF( IW - 1 )  41, 41, 42                                         00301410
   41 CONTINUE                                                          00301420
      WRITE(SYSOUT,1046) XMACH,NMOD,SHIFT                               00301430
         IF( KONTRL(86)  .LT.  0 )  WRITE(SYSOUT,1047)                  00301440
      KONTRL(  85 ) = ISHIFT                                            00301450
      KONTRL(  87 ) = MACH                                              00301460
        GO TO 5                                                         00301470
   42 CONTINUE                                                          00301480
      WRITE(SYSOUT,1048) ISOLAT,SHIFT                                   00301490
      KONTRL(  90 ) = ISOLAT                                            00301500
      KONTRL(  91 ) = NEWPTS                                            00301510
      KONTRL ( 93 ) = ISHIFT                                            00301520
        GO TO 5                                                         00301530
C******                                                                 00301540
C         LABEL  IS  COORD                                              00301550
C******                                                                 00301560
   50 ICONT( 5) = 1                                                     00301570
      DO 53  K=1,NP                                                     00301580
      READ (TYSIN ,1050)  IGPN, XX, YY, ZZ                              00301590
        IF( ENDFILE  TYSIN )  298,51                                    00301600
   51 CONTINUE                                                          00301610
      XYZ(1,IGPN) = XX/CHORD                                            00301620
      XYZ(2,IGPN) = YY/CHORD                                            00301630
      XYZ(3,IGPN) = ZZ/CHORD                                            00301640
   53 CONTINUE                                                          00301650
      IF(IW .EQ. 1) GO TO 54                                            00301660
C         SHIFT  ALONG  X-AXIS.   'D'                                   00301670
      DO 52 K=1,NP                                                      00301680
   52 XYZ(1,K) = XYZ(1,K) + SHIFT                                       00301690
   54 CONTINUE                                                          00301700
      WRITE(SYSOUT,1051)                                                00301710
      DO  55  K=1,NP                                                    00301720
   55 WRITE(SYSOUT,1052)  K, XYZ(1,K), XYZ(2,K), XYZ(3,K)               00301730
        GO TO 5                                                         00301740
C******                                                                 00301750
```

```
                                                                        00301770
   60 ICONT( 6) = 1                                                     00301780
      K   = 0                                                           00301790
   61 K =K+1                                                            00301800
        IF( K - NELW ) 62, 62, 68                                       00301810
   62 CONTINUE                                                          00301820
      READ (TYSIN ,1042)  IREPT, IELM, I1, I2, I3                       00301830
        IF( ENDFILE  TYSIN )  298,63                                    00301840
C                                                                       00301850
   63 CONTINUE                                                          00301860
        IF( IREPT  .EQ.  IYES )  GO TO 64                               00301870
      KELM   = IELM                                                     00301880
      IBLN(1,KELM) = I1                                                 00301890
      IBLN(2,KELM) = I2                                                 00301900
      IBLN(3,KELM) = I3                                                 00301910
      I11  = I1                                                         00301920
      I22  = I2                                                         00301930
      I33  = I3                                                         00301940
        GO TO 61                                                        00301950
   64 CONTINUE                                                          00301960
        IF( K  .LE.  1 )  GO TO 69                                      00301970
      K = K + IELM - 1                                                  00301980
      DO 66 K6=1,IELM                                                   00301990
      I11 = I11 + I1                                                    00302000
      I22 = I22 + I2                                                    00302010
      I33 = I33 + I3                                                    00302020
      KELM = KELM + 1                                                   00302030
      IBLN(1,KELM) = I11                                                00302040
      IBLN(2,KELM) = I22                                                00302050
      IBLN(3,KELM) = I33                                                00302060
   66 CONTINUE                                                          00302070
        GO TO 61                                                        00302080
C******                                                                 00302090
C         ELEMENT  'REPEAT'  BLUNDER./    FLUSH  INPUT  STREAM  TO      00302100
C         NEXT  LABEL-DATA  SECTION.      COUNT  THIS  SECTION  MISSING. 00302110
C******                                                                 00302120
   69 WRITE(SYSOUT,1067)                                                00302130
      ICONT(6) = 0                                                      00302140
      NERR   = IERR + NERR                                              00302150
  696 READ (TYSIN ,1030) CHAR                                           00302160
        IF( ENDFILE  TYSIN )  298,697                                   00302170
  697 CONTINUE                                                          00302180
        IF( CHAR .EQ. BLANK  .OR.  CHAR .EQ.  REPE )  GO TO 696         00302190
      BACKSPACE  TYSIN                                                  00302200
        GO TO 5                                                         00302210
   68 CONTINUE                                                          00302220
      K = K-1                                                           00302230
      IF( K  .EQ.  NELW ) GO TO 67                                      00302240
      WRITE(SYSOUT,1068) K,NELW                                         00302250
        GO TO 299                                                       00302260
   67 CONTINUE                                                          00302270
      WRITE(SYSOUT,1061)                                                00302280
      DO  65  K=1,NELW                                                  00302290
      WRITE(SYSOUT,1062) K, IBLN(1,K), IBLN(2,K), IBLN(3,K)             00302300
   65 CONTINUE                                                          00302310
        GO TO 5                                                         00302320
C******                                                                 00302330
C         LABEL  IS  LEDGE                                              00302340
C******                                                                 00302350
```

```
      WRITE(SYSCLT,1065)  (  LND(K),K=1,NLND)                           00302380
        GO TO 5                                                         00302390
C******                                                                 00302400
C         LABEL  IS  MODE                                               00302410
C******                                                                 00302420
   80 ICONT(8) = 1                                                      00302430
      WRITE(SYSCLT,1080)                                                00302440
      DO 82 I1=1,NP                                                     00302450
      READ (IYSIN ,1060)     (XMCD(I1,I2),I2=1,NMOD)                    00302460
      WRITE(SYSCLT,1053) I1,(XMCD(I1,I2),I2=1,NMOD)                     00302470
        IF( ENDFILE  IYSIN )  298,82                                    00302480
   82 CONTINUE                                                          00302490
        GO TO 5                                                         00302500
C******                                                                 00302510
C         LABEL  IS  END                                                00302520
C******                                                                 00302530
   9C ICONT(9) = 1                                                      00302540
      IF( IW .EC. 2 )  WRITE(SYSOUT,1090)                               00302550
C-----------------------------------------------------------------------00302560
C                                                                       00302570
C . . . . . . .D A T A     C H E C K     S E C T I O N. . . . . . . .  00302580
C                                                                       00302590
C-----------------------------------------------------------------------00302600
      IERR = C                                                          00302610
C******                                                                 00302620
C         HAS  ALL  RECUIRED  DATA  BEEN  INPUT.    RETURN  IF  YES.    00302630
C         ( 1=YES,  0=NO,  -1=OPTIONAL )                                00302640
C******                                                                 00302650
      DO 210  K=1,NAMES                                                 00302660
         IF( ICCNT(K)  .NE. 0 )  GO TO 210                              00302670
      IERR = IERR + 1                                                   00302680
      WRITE(SYSCLT,2020)  HCR(K)                                        00302690
  210 CONTINUE                                                          00302700
      NERR    = IERR + NERR                                             00302710
      IF( IERR .GT. 0 ) GC TO 99                                        00302720
C******                                                                 00302730
C         ALL  BCCLEAN  CAN  BE  CHECKED  FOR  RANGE,   0  .LE.  NP     00302740
C         WHERE  'NP'  IS  THE  TOTAL  NUMBER  OF  GRID  POINTS  ON     00302750
C         THE  SYSTEM.    IT  IS  ALSO  THE  LARGEST  GRID-POINT NBR.   00302760
C         TO  REACH  THIS  TEST  SECTION,   NP  MUST  BE  CORRECT.      00302770
C******                                                                 00302780
      DO 221 K=1,NELW                                                   00302790
      DO 221 J=1,3                                                      00302800
      KGP   = IBLN(J,K)                                                 00302810
        IF( KGP.GT.0  .AND  .KGP.LE.NP ) GO TO 221                      00302820
      IERR = IERR + 1                                                   00302830
      WRITE(SYSCLT,2030) K,J,KGP                                        00302840
  221 CONTINUE                                                          00302850
C                                                                       00302860
      DO 23C  K=1,NLND                                                  00302870
      IF(LND(K)  .GT.  0  .AND.  LND(K)  .LE.  NP )  GO TO 230          00302880
      WRITE(SYSCLT,2035) LND(K)                                         00302890
      IERR = IERR + 1                                                   00302900
  230 CONTINUE                                                          00302910
      NERR    = IERR + NERR                                             00302920
   99 CONTINUE                                                          00302930
      RETURN                                                            00302940
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 00302950
```

```
103C FORMAT(     A4,   A2,   I4   )
104C FORMAT(     5X, 12I5         )
1042 FORMAT(     A4,2X,I4,  11I5  )
1050 FORMAT(     5X,   I5, 4E15.6 )
1053 FORMAT(     5X,   I5, 4E15.6 /10X,4E15.6/10X,2E15.6)
1052 FORMAT(   11X, I5,  4E20.6  )
106C FORMAT(    10X,        4E15.6 )
1062 FORMAT(    16X,  4I8          )
10450FORMAT(// 1CX,  12H...W I N G..., I1, 24H.....C O N T R L S...
    1       // 1CX,  30HNBR  GRID  POINTS  ON  WING...,  I4 /
    *          10X,  30HNBR  ELEMENTS ON  PLANFURM.....  I4 /
    2          10X,  30HNBR  DEGREES  OF  FREEDOM.....,  I4 /
    3          10X,  30HNBR  LEADING  EDGE  POINTS.....  I4 /
    4          10X,  30HNBR  DIVISIONS  IN  DIAPHRAGM.,  I4 /
    5          1CX,  30HSYMMETRY   FACTOR.............,  I4 /
    6          10X,  22HREFERENCE  LENGTH.....         , E12.5 /
C   7          10X,  22HFREQUENCY  (IMAG).....         , E12.5 /
    8     //// 1CX,  36H..G E N E R A L....C O N T R L S..    //)
10460FORMAT(   1CX,  22HM A C H  NUMBER.......         , E12.5 /
    1          1CX,  30HNBR  MCDE  SHAPES.............,  I4 /
    2          10X,  22HTRUNCATICN  (EPS).....         , E12.5 / )
1047CFORMAT(/////
    1          1CX,  30H*RESTART   OPTION   REQUESTED* )
10480FORMAT(   10X,  30HEXTRAPCLATION REQUIRED(1=YES)., I4 /
    1          10X,  22HSTAGGER     (LE.TO.LE)         , E12.5 /)
10510FORMAT(1H1  //
    1          2CX,  24HGRID  POINT  COORDINATES       //
    2           9X,  12HGRID  PUINT , 6X, 1HX, 19X, 1HY ,19X,1HZ)
10610FORMAT(1H1  //
    1          23X,  22HELEMENT  GRID  POINTS          //
    2          2CX,  30HELEM        A        B        C   / )
1065 FORMAT(///1CX,  30HLEADING  EDGE  GRID  PUINTS.  / 5X,20I5 /)
10670FORMAT(// 1CX,  20H*** E R R O R ***   //
    1          14X,  50HAN  ELEMENT  'REPEAT'  CARD  MAY  NOT  APPEAR
    2         /14X,  50HBEFORE  ANY  ELEMENTS  HAVE  BEEN  DEFINED.
    3         /24X,  4CHSHUFFLE  DECK  AND  TRY  AGAIN ./
    4        //14X,  50H(I  WILL  IGNORE  ALL  ELEMENT  DATA  THIS PASS
    5)          )
10680FORMAT(/// 20X, 36HTHE  NUMBER  OF  ELEMENTS  GENERATED  , I6  /
    1           2CX, 36HDISAGREES  WITH  YOUR  REQUEST  FOR   , I6 / )
107C0FORMAT(1H1 ////  10X, 20A4, //
    1          1CX,  36HNO  MATCH  FOR  ABOVE  LABEL  CARD.  )
108C0FORMAT(1H1/10X, 20HMCDE  SHAPE  DATA
    1          / 9X, 20H******************** /)
1090 FORMAT(////11X, 34H+++++ END  CARD  ENCOUNTERED +++++ /)
1091CFORMAT(// 1CX,  40HAN  END-CF-FILE  IN  THE  INPUT  STREAM    /
    1          10X,  40HHAS  BEEN  ENCOUNTERED.     IT  APPEARS    /
    2          10X,  40HTHAT  AN  'END' ,  OR  'ENDATA'  CARD IS   /
    3          10X,  40HMISSING   OR  A  MISCOUNT  OF  DATA....    )
1099CFORMAT(// 1CX,  20H*** E R R O R ***   //
    1          14X,  5CHSYSTEM  CARD  MUST  PRECEDE  ALL  LABELED  DATA
    2         /14X,  50HINFUT  STREAM  WILL  BE  FLUSHED  TO  NEXT  WING
    3         /26X,  25HBETTER-LUCK-NEXT-FLIGHT./    )
2020 FORMAT(// 10X,  40H**E R R O R**   MISSING  LABEL  DATA....,2X,A4)
20300FORMAT(// 1CX,  24H**E R R O R**   ELEMENT , I4
    1               ,  14H   GRID  POINT  ,I3 , 4H  = , I6  )
2035 FORMAT( // 5X,  28HLEDGE  DATA,  OUT OF RANGE  , I4 )
     END
```

DECK 004

```
      SUBROUTINE SIZE                                                   00400010
     *     ( KCN , IS , KEY )                                           00400020
      INTEGER     KCN( 1 ) , IS( 1 ) , OUT                              00400030
      DATA        NXYZ, NXMOD1, NXMOD2, NELEM, N20 /                    00400040
     *             80,     50,     10,   100,  20 /                     00400050
C                                                                       00400060
      OUT = KCN(58)                                                     00400070
      IF( KEY .EQ. 2 )  GOTO 2                                          00400080
C       KEY =1  IMPLIES  NOT  YET  ENTERED  'INPUT'                     00400090
C       BUT  WORK  SPACE  MUST  BE  DEFINED                             00400100
C******                                                                 00400110
C                                                                       00400120
C...MAIN....IS(  )...SUB.INPUT...LENGTH....................             00400130
C     N1          1        XMOD      NXMOD1 * NXMOD2 * 2                00400140
C     N2          2        XYZ       3       * NXYZ   * 2               00400150
C     N3          3        IBLN      3       * NELEM  * 2               00400160
C     N4          4        MLINE     N20     * 2                        00400170
C     N5          5        LND       N20     * 2                        00400180
C     N6          6        TITLE     N20     * N20                      00400190
C     N7          7        PTUSED    NXYZ                               00400200
C     N8          8         -        TOTAL  LENGTH                      00400210
C                                                                       00400220
C******                                                                 00400230
      IS( 1) = 1                                                        00400240
      IS( 2) = IS(1) + NXMOD1 * NXMOD2 * 2                              00400250
      IS( 3) = IS(2) + 3 * NXYZ * 2                                     00400260
      IS( 4) = IS(3) + 3 * NELEM * 2                                    00400270
      IS( 5) = IS(4) + N20 * 2                                          00400280
      IS( 6) = IS(5) + N20 * 2                                          00400290
      IS( 7) = IS(6) + N20 * N20                                        00400300
      IS( 8) = IS(7) + NXYZ                                             00400310
      GOTO 99                                                           00400320
    2 CONTINUE                                                          00400330
      KPRINT = KCN(72)                                                  00400340
C     KPRINT = 1                                                        00400350
C******                                                                 00400360
C         CHECK  INDIVIDUAL  VALUES  FIRST.                             00400370
C******                                                                 00400380
C         WING  1                                                       00400390
      IF( KCN( 1)  .GT.  IS(11) )  GOTO 999                             00400400
      IF( KCN( 3)  .GT.  IS(12) )  GOTO 999                             00400410
      IF( KCN( 4)  .GT.  IS(13) )  GOTO 999                             00400420
      IF( KCN( 5)  .GT.  IS(14) )  GOTO 999                             00400430
      IF( KCN( 7)  .GT.  IS(15) )  GOTO 999                             00400440
      IF( KCN(13)  .GT.  IS(16) )  GOTO 999                             00400450
C         WING  2                                                       00400460
      IF( KCN(16)  .GT.  IS(11) )  GOTO 999                             00400470
      IF( KCN(18)  .GT.  IS(12) )  GOTO 999                             00400480
      IF( KCN(19)  .GT.  IS(13) )  GOTO 999                             00400490
      IF( KCN(20)  .GT.  IS(14) )  GOTO 999                             00400500
      IF( KCN(22)  .GT.  IS(15) )  GOTO 999                             00400510
      IF( KCN(28)  .GT.  IS(16) )  GOTO 999                             00400520
C******                                                                 00400530
C         OK  HERE.                                                     00400540
C         COMPLETE  ALL  WORK  SPACE  FOR  'MASTER'                     00400550
```

```
C     J1      9       INDWN     NXMOD1 * NXMOD2 * 2                           00400570
C     J2      10      PM        MAXO(NP1**2,NP2**2)                           00400580
C     J3      11      -         TOTAL  LENGTH                                 00400590
C******                                                                       00400600
      INDWN  = NXMOD1 * NXMOD2 * 2                                            00400610
      IS( 9) = 1                                                              00400620
      IS(10) = IS (9) + INDWN                                                 00400630
      NT  = KON( 5 )                                                          00400640
      IF( NT .LT.  KON(20) )   NT = KON(20)                                   00400650
      IS(11) = IS(10)+ NT*NT                                                  00400660
C******                                                                       00400670
C        COMPLTE  ALL  WORK  SPACE  FOR   'ASSPOT'                            00400680
C...MAIN....IS(  )...SUB.ASSPOT..LENGTH....................                   00400690
C     M1      12      INDWN     NXMOD1 * NXMOD2 * 2                           00400700
C     M2      13      PHI       NP     * NP                                   00400710
C     M3      14      PHIXX     NP     * ND                                   00400720
C     M4      15      PHI22     ND     * ND                                   00400730
C     M5      16      T         ND     * NP                                   00400740
C     M6      17      -         TOTAL  LENGTH                                 00400750
C                                                                             00400760
C******                                                                       00400770
      IS(12) = 1                                                              00400780
      IS(13) = IS(12)+ INDWN                                                  00400790
      IS(14) = IS(13)+ KON(51)                                                00400800
      IS(15) = IS(14) + KON(47) * KON(50)                                     00400810
      IS(16) = IS(15) + KON(52)                                               00400820
      IS(17) = IS(16) + KON(47) * KON(50)                                     00400830
C******                                                                       00400840
C        COMPLTE  ALL  WORK  SPACE  FOR   'ASSDWN'                            00400850
C...MAIN....IS(  )...SUB.ASSDWN..LENGTH....................                   00400860
C     L1      18      INDWN     NXMOD1 * NXMOD2 * 2                           00400870
C     L2      19      W12       NP1    * NP2+ND2                              00400880
C     L3      20      W21       NP2    * NP1+ND1                              00400890
C     L4      21      A         NP     * NP                                   00400900
C     L5      22      WT        NP     * MAXO(ND,NMOD)                        00400910
C     L6      23      T         ND     * NP                                   00400920
C     L7      24      C         NP     * NMOD                                 00400930
C     L8      25      WORK      NOT  USED SET = 2                             00400940
C     L9      26      -         TOTAL  LENGTH                                 00400950
C                                                                             00400960
C******                                                                       00400970
      IS(18) = 1                                                              00400980
      IS(19) = IS(18)+ INDWN                                                  00400990
      IS(20) = IS(19) + KON( 5) * KON(49)                                     00401000
      IS(21) = IS(20) + KON(20) * KON(48)                                     00401010
      IS(22) = IS(21) + KON(51)                                               00401020
      NT     = KON( 4)                                                        00401030
      IF( KON(50) .GT. NT )  NT = KON(50)                                     00401040
      IS(23) = IS(22) + KON(47) * NT                                          00401050
      IS(24) = IS(23) + KON(47) * KON(50)                                     00401060
      IS(25) = IS(24) + KON(47) * KON( 4)                                     00401070
      NT     = KON( 5)                                                        00401080
      IF( KON(20) .GT. NT )  NT = KON(20)                                     00401090
      IS(26) = IS(25) +    2                                                  00401100
C-----------------------------------------------------------------------------00401110
C                                                                             00401120
C        CHECK  TOTAL  LENGTH OF  WORK SPACE  HERE                            00401130
C        IS(28)  =  MAXIMUM  ALLOTED   'REAL'                                 00401140
C        IS(27)  =  MAXIMUM  ALLOTED   'COMPLEX'                              00401150
```

```
      IF( IS(27).GT. IS(28) ) GC TO 999                                 00401170
      IF( IS(11).GT. IS(27) ) GC TO 999                                 00401180
      IF( IS(17).GT. IS(27) ) GC TO 999                                 00401190
      IF( IS(26).GT. IS(27) ) GC TO 999                                 00401200
C-----------------------------------------------------------------------00401210
   98    IF(KPRINT .GT. 0) WRITE(CUT,1000) (IS(J),J=1,28)               00401220
   99 RETURN                                                            00401230
  999 CONTINUE                                                          00401240
      KON(ICO) = 0                                                      00401250
      KPRINT = 2                                                        00401250
      CALL  CRASH                                                       00401270
      WRITE(CUT,9001)                                                   00401280
      GO TO 98                                                          00401290
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 00401300
1000 OFORMAT(///    20X,30HP R C B L E M ........ S I Z E               00401310
     1        //    10X, 8I7 // 10X, 3I7 //  10X, 6I7 //                00401320
     2              10X, 9I7  ///                                       00401330
     3              10X,20HMAXIMUM  COMPLEX  , ,I7                      00401340
     4       ///    10X,20HMAXIMUM  REAL     , ,I7        )             00401350
9001  FORMAT(1H1////20X,40HP R C B L E M........T O.......L A R G E )   00401360
      END                                                               00401370
```

DECK 009

```
      SUBROUTINE  FAKTOR( FAC, IBLN, NEL, NGP )                         00900010
      DIMENSION  FAC   (  1    ) ,IBLN  (   1    )                      00900020
C                                                                       00900030
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00900040
C     PURPOSE..  TO  DETERMINE  WEIGHT-FACTORS  AT  EACH  NODE  POINT.  00900050
C                ON  THE  WING(PLANFORM)  ONLY.    THE  FACTOR  EQUALS  00900060
C                THE  NUMBER  OF  ELEMENTS  HAVING  THAT  NODE  POINT   00900070
C                AS  COMMON.                                            00900080
C                                                                       00900090
C     LOGIC..    THE  BOOLEAN  ARRAY  CONTAINING  ELEMENT  INFO. ,      00900100
C                EXTERNAL  TO  THIS  SUB.PROGRAM,  IS  'IBLN(3,100,2)'. 00900110
C                INDIVIDUAL  GRID-POINT  INFO. (3-EACH ELEMENT)  IS     00900120
C                STORED  COLUMN-WISE  AND  SEQUENTIALLY.   THEREFORE,   00900130
C                NEED  ONLY  ZIP  THRU  3*NEL  NODE-POINTS,  COUNTING'  00900140
C                AS  YOU  GO.                                           00900150
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 00900160
C                                                                       00900170
      NBR    = 3 * NEL                                                  00900180
      DO 2 I= 1,NGP                                                     00900190
      K      = 0                                                        00900200
      DO 4 J= 1,NBR                                                     00900210
        IF( IBLN(J) - I ) 4,3,4                                         00900220
    3 K      = K + 1                                                    00900230
    4 CONTINUE                                                          00900240
      FAC(I)= 1.0 / FLOAT(K)                                            00900250
    2 CONTINUE                                                          00900260
C     WRITE(  6,1000) NEL,NGP, (J,FAC(J),J=1,NGP)                       00900270
C1000 FORMAT(1H1,  10X,12HNEL    NGP    ,    2I6   //                   00900280
C    *         12X,12H1 / FACTOR  /  100(5X,I5,E18.6/))                 00900290
      RETURN                                                            00900300
      END                                                               00900310
```

DECK 005

```
      SUBROUTINE  DIAPH3                                                00500010
     *          ( XY        ,IBLN     ,LND      ,LDNW     ,IAR     ,NLND    00500020
     *           ,NLDNW     ,NEL      ,NGP      ,NELD1    ,NDIV    ,MACH    00500030
     *           ,MU        ,OUT      ,PI       ,NERR     )                 00500040
     OINTEGER    OUT                  ,IBLN  (  3,1   ) ,LND    (   1    ) 00500050
     1           ,LDNW  (   1    ) ,IAR   ( 20,1   )                      00500060
      REAL       MACH                 ,MU                ,XY    (  3,1   ) 00500070
      COMMON     /    BASIC      /  KONTRL(  98    ) ,IERR                00500080
C                                                                       00500090
      IERR   = 0                                                        00500100
      XDIV   = FLOAT(NDIV)                                              00500110
      NN     = NGP + 1                                                  00500120
      IGP    = NN                                                       00500130
      IEL    = NEL + 1                                                  00500140
      I      = LND(1)                                                   00500150
      ISTOP  = NLND - 1                                                 00500160
      COSMUD = COS(MU) / XDIV                                           00500170
      SINMUD = SIN(MU) / XDIV                                           00500180
      ANG3   = PI - 2.0 * MU                                            00500190
      SINA3  = SIN(ANG3)                                                00500200
C                                                                       00500210
      DO 10 J=1,ISTOP                                                   00500220
      I1     = LND(J+1)                                                 00500230
      DY     = XY(2,I1) - XY(2,I)                                       00500240
      DX     = XY(1,I1) - XY(1,I)                                       00500250
      ANG4   = ATAN( DY / DX )                                          00500260
      ANG1   = MU - ANG4                                                00500270
C     ANG2   = MU + ANG4                                                00500280
      RATIO  = SIN(ANG1) / SINA3                                        00500290
      C      = SQRT( DX * DX  +  DY * DY )                              00500300
      A      = C * RATIO                                                00500310
      DELX   = -A * COSMUD                                              00500320
      DELY   =  A * SINMUD                                              00500330
      XY(1,NN) = XY(1,I1) + DELX                                        00500340
      XY(2,NN) = XY(2,I1) + DELY                                        00500350
      NN     = NN + 1                                                   00500360
        IF( NDIV .LE. 1 )  GO TO 10                                     00500370
      DO 11 J1=2,NDIV                                                   00500380
      XY(1,NN) = XY(1,NN-1) + DELX                                      00500390
      XY(2,NN) = XY(2,NN-1) + DELY                                      00500400
      NN     = NN + 1                                                   00500410
   11 CONTINUE                                                          00500420
  10  CONTINUE                                                          00500430
      NGP   = NN - 1                                                    00500440
C                                                                       00500450
C.......BEGIN ASSIGNMENT OF BOOLEAN                                     00500460
C                                                                       00500470
      N1    = NLND - 1                                                  00500480
      N2    = NLND - 2                                                  00500490
      N3    = NDIV + 1                                                  00500500
      NEL1 = NEL                                                        00500510
C                                                                       00500520
C.......STUFF  WORK  ARRAY...                                           00500530
C                                                                       00500540
      IB      = IGP - 1                                                 00500550
      IB          = IB + 1
      IAR(J,K)= IB
   49 CONTINUE
      IAR(J,1)=LNC(J+1)
   50 CONTINUE
C
C.......SELECT  ELEMENT  GRID  POINTS  ONE  AT  A  TIME
C
      DO 60 K=1,NDIV
C
C.......STORE ONE UPPER ELEMENT FOR EACH NDIV
C
      NEL1         = NEL1 + 1
      IBLN(1,NEL1) = IAR(  1,K  )
      IBLN(2,NEL1) = IAR(  1,K+1)
      IBLN(3,NEL1) = LND(   1   )
C
C.......STORE LEFT SIDE ELEMENTS   GO ING DOWN
C
      DO 61 KL=1,N2
      NEL1         = NEL1 + 1
      IBLN(1,NEL1) = IAR(KL+1,K  )
      IBLN(2,NEL1) = IAR(KL  ,K+1)
      IBLN(3,NEL1) = IAR(KL  ,K  )
   61 CONTINUE
C
C.......STORE RIGHT SIDE ELEMENTS   GOING DOWN
C
      DO 62 KR=1,N2
      NEL1         = NEL1 + 1
      IBLN(1,NEL1) = IAR(KR+1,K  )
      IBLN(2,NEL1) = IAR(KR+1,K+1)
      IBLN(3,NEL1) = IAR(KR  ,K+1)
   62 CONTINUE
 60   CONTINUE
      NEL   = NEL1
      NELD2 = NDIV * (2 * NLND - 3)
      NELD1 = NELD1 + NELD2
C
      NLDNW      = NLND - 1
      DO 90 K=1,NLDNW
      LDNW(K) = IAR(K,N3)
   90 CONTINUE
C
C-----------------------------------------------------------------------
      DO 198 K=  1,NGP
      IF(XY(1,K).GE.0.0    .AND. XY(2,K) .GE. 0.0 )  GO TO 198
      IERR = IERR + 1
  198 CONTINUE
      WRITE(OLT,1000)
      WRITE(OLT,1001)
C
C.....GET  AN  'Z'  VALUE  AND  STUFF
C
      Z  = XY(3,IGP-1)
      DO 200 K=IGP,NGP
      XY(3,K) = Z
      WRITE(OLT,1002)  K, XY(1,K), XY(2,K), XY(3,K)
      DO 210 J=IEL,NEL
      WRITE(CLT,1C04)  J, IBLN(1,J), IBLN(2,J), IBLN(3,J)
210   CONTINUE
      WRITE(CLT,1C05)  NLDNW, (LDNW(K),K=1,NLDNW)
        IF( IERR .LE. 0 ) GC TO 99
      CALL  CRASH
      WRITE(OLT,9001)
      CALL  CCRED
   99 RETURN
C
1000 OFORMAT(1HC//20X, 26HDIAPHRAGM  ELEMENTS  WERE    /
     1            20X, 20HACDED  AS  FOLLOWS.       //)
1001 OFORMAT(1HC//20X, 24HGRID  POINT  COORDINATES   /
     1             9X, 12HGRID  PCINT  , 6X,1HX,19X,1HY,19X,1HZ )
1002  FORMAT(     11X, I5, 4E20.6  )
1003 OFORMAT(1HO//23X, 24HELEMENT  GRID  POINTS      /
     1            20X, 30HELEM        A       B       C   /)
1004  FORMAT(     16X,4I8  )
1005  FORMAT(//18X,I4, 24H  MACH LINE GRID POINTS. /3(6X,9I5/))
9001  FORMAT(1H1,10X, 38H**NEGATIVE  GRID  POINTS  ENCOUNTERED. )
      END
```

DECK 006

```
 SUBROUTINE MASTER                                                      00600010
*    ( INDWN  ,XMOD   ,XYZ    ,MLINE                                    00600020
*     ,IBLN   ,PM     ,PTUSED ,LND    )                                 00600030
0COMPLEX                                                                00600040
1         INDWN (100,10  ) ,PM     (    1    ) ,FPOT   (   3    )       00600050
2        ,DWN   (   3    ) ,DWNI   (    3    ) ,POT1                    00600060
3        ,POT2             ,POT3               ,DWN1                    00600070
4        ,DWN2             ,DWN3               ,TSX                     00600080
5        ,VALUE                                                         00600090
0INTEGER                                                                00600100
1         IBLN  ( 3,100,2) ,MLINE ( 20,1     ) ,LND    (  20,1  )       00600110
2        ,PTUSED(   80   ) ,ILB    (    3    ) ,IMB    (    3   )       00600120
3        ,OUT              ,CASE               ,SWITCH                  00600130
4        ,RESTAR                                                        00600140
0DIMENSION                                                              00600150
1         XYZ   ( 3,80,2 ) ,XMOD   (50,10,2 ) ,XKVEC (    2   )        00600160
2        ,FACTOR(  100   ) ,REC1   (   14    ) ,R      (    3   )       00600170
3        ,X     (    3   ) ,Y      (    3    ) ,Z      (    3   )       00600180
4        ,BETA  (    3   ) ,OMX    (    3    ) ,C      (  3,3   )       00600190
5        ,CEQ   (   9    )                                              00600200
 COMMON   /    BASIC    /  KONTRL(  100   )                             00600210
0COMMON   /    TRINT    /  FPOT                ,DWN                     00600220
1        ,DWNI             ,UU     (    5    ) ,CC     (    5   )       00600230
2        ,XI    (    3   ) ,YI     (    3    ) ,ZI     (    3   )       00600240
3        ,T     (   3,3  ) ,XR                 ,YR                      00600250
4        ,ZR               ,XK                 ,ZMACH                   00600260
5        ,BEETA            ,PIE                ,XKOM                    00600270
6        ,VSQRD            ,CIOPIV                                      00600280
 EQUIVALENCE                                                            00600290
1         (BETA(1),BETA1),(BETA(2),BETA2),(BETA(3),BETA3)               00600300
2        ,( ILB(1),ILB1 ),( ILB(2),ILB2 ),( ILB(3),ILB3 )               00600310
3        ,( OMX(1),OMX1 ),( OMX(2),OMX2 ),( OMX(3),OMX3 )               00600320
5        ,(   R(1),R1   ),(   R(2),R2   ),(   R(3),R3   )               00600330
6        ,(C(1,1) ,CEQ(1)),(KAPPA,CAPPA),(ISYMC,SYMC),(KX,XK)           00600340
 EQUIVALENCE                                                            00600350
1         (REC1( 1),KPT   ), (REC1( 2),IMB(1),IMB1 )                    00600360
2        ,(IMB ( 2),IMB2  ), (IMB ( 3),IMB3  )                          00600370
3        ,(REC1( 5),IWING ), (REC1( 6),KWING )                          00600380
3        ,(REC1( 7),L     ), (REC1( 8),M     ),(REC1( 9),POT1  )        00600390
4        ,(REC1(11),POT2  ), (REC1(13),POT3 )                           00600400
 EQUIVALENCE                                                            00600410
1         (KONTRL(75),IW    )                                           00600420
2        ,(KONTRL(76),KW    )                                           00600430
3        ,(KONTRL(77),LRECL )                                           00600440
4        ,(KONTRL(78),MINFL )                                           00600450
5        ,(KONTRL(79),INCDE )                                           00600460
6        ,(KONTRL(80),SWITCH)                                           00600470
7        ,(KONTRL(81), ISW  )                                           00600480
8        ,(KONTRL(87),XMACH )                                           00600490
9        ,(KONTRL(88),VEETA )                                           00600500
B        ,(KONTRL(46),  PI  )                                           00600510
C        ,(KONTRL(63),KPRINT)                                           00600520
D        ,(KONTRL(95),KON95 )                                           00600530
E        ,(KONTRL(86),RESTAR)                                           00600540
F        ,(KONTRL(55),LTAPE )                                           00600550
```

```
C
C
      DATA      IM10, IM20, IZR/ -10, -20, 0 /
      DATA      CEQ / 2.,1.,1.,1.,2.,1.,1.,1.,2. /
C
C        NOTE.. IF  'INDWN'  EVER  GENERATED  ELSEWHERE(LIKE MAYBE A
C               SPLINE-FIT-PROGRAM,   THEN  DURING 'RESTART'  THIS
C               ROUTINE  WILL NOT  BE  ENTERED
C
      OUT    = KCNTRL(58)
      NMOD   = KCNTRL( 4)
      NP1    = KCNTRL( 5)
      ND1    = KCNTRL( 8)
      NP2    = KCNTRL(20)
      ND2    = KCNTRL(23)
      NPMAX  = KCNTRL(47)
      ISOLAT= KCNTRL(90)
      NEXTR1= LND(1,2)
      NEXTR2= LND(2,2)
      C106PI= 1.0 / 6.0
      C1024P= C106PI*0.25
      C10PIV= 1.00 / (PI * VEETA)
      ZMACH  = XMACH
      BEETA  = VEETA
      VSQRD  = VEETA * VEETA
      KONS5  = 0
C
      PIE    = PI
      UU(5)  = 0.90617984
      UU(4)  = 0.53846931
      UU(3)  = 0.0
      UU(2)  = -UU(4)
      UU(1)  = -UU(5)
      CC(1)  = 0.11846344
      CC(2)  = 0.23931434
      CC(3)  = 0.28444444
      CC(4)  = CC(2)
      CC(5)  = CC(1)
C
      SX1    = XMACH * XMACH / VSQRD
      KX     = KCNTRL( 2)
      XKVEC(1)= XK * SX1
      KX     = KCNTRL(17)
      XKVEC(2)= XK * SX1
C
      DO 4 K=1,NMOD
      DO 4 J=1,NPMAX
    4 INDWN(J,K) = (0.0,0.0)
C • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • • •
C
      IBP    = -15
      IBUR = 0
      DO 1 IWING=1,2
      IW     = IWING
      IF( IWING .EQ. 2 )  IBUR = NP1
      IBP    = IBP + 15
      NEL    = KCNTRL(IBP + 1 )
      KAPPA = KCNTRL(IBP + 2 )
      NF     = KCNTRL(IEP + 5 )
      NELW   = KCNTRL(IEP + 6 )
      NGPTS  = KCNTRL(IEP + 7 )
      ND     = KCNTRL(IEP + 8 )
      NELD   = KCNTRL(IEP + 9 )
      NDIV   = KCNTRL(IEP + 11)
      COMDIV= CAPPA * C1C24P
      CALL  FAKTCR ( FACTOR , IBLN(1,1,IWING), NELW,NP )
         IF( RESTAR  .LT.  0 )  GOTO 401
C
C        INITIALIZE  ARRAY  CF  NODE POINTS  USED
C        (0 = NC   ,   1 = YES)
      DO 11 J11 = 1,NGPTS
11    PTUSED(J11) = 0
      LIMIT  = NP * NP
      DO 105 J1C5=1,LIMIT
105   PM(J1C5) = (0.0,0.0)
  401 CONTINUE
C
C        SELECT  RECEIVING  ELEMENT  FROM  CURRENT  WING
C
      DO 1000 L=1,NEL
      LRECL = L
      ILB1   = IBLN(1,L,IWING)
      ILB2   = IBLN(2,L,IWING)
      ILB3   = IBLN(3,L,IWING)
      X(1)   = XYZ(1,ILB1,IWING)
      Y(1)   = XYZ(2,ILB1,IWING)
      Z(1)   = XYZ(3,ILB1,IWING)
      X(2)   = XYZ(1,ILB2,IWING)
      Y(2)   = XYZ(2,ILB2,IWING)
      Z(2)   = XYZ(3,ILB2,IWING)
      X(3)   = XYZ(1,ILB3,IWING)
      Y(3)   = XYZ(2,ILB3,IWING)
      Z(3)   = XYZ(3,ILB3,IWING)
C
C        IF CURRENT ELEMENT  IS  NCT  ON  WING,   SKIP
C
      IF( L .GT. NELW )  GO TO 1090
C
C        CCMPUTE  FRESSURE  MATRIX
C
      BETA1   = Y(2) - Y(3)
      BETA2   = Y(3) - Y(1)
      BETA3   = Y(1) - Y(2)
      DET     = X(1) * BETA1 + X(2) * BETA2 + X(3) * BETA3
      OMX1    = BETA1 / DET
      OMX2    = BETA2 / DET
      OMX3    = BETA3 / DET
         IF( RESTAR  .LT.  0 )  GOTO 402
      ABSDET = ABS( DET )
      CCNST = ABSDET * COMDIV
      SIGN    = C1C6PI
        IF( DET .LT. 0.0 )  SIGN = -SIGN
      BETA1   = BETA1 * SIGN
      BETA2   = BETA2 * SIGN
      BETA3   = BETA3 * SIGN
C
      DO 101 L1C1=1,3
      PM(LOC)= PM(LOC) + CMPLX(BETA1 , CONST * C(L101,1))
      LOC    = IC + NF * (ILB2-1)
      PM(LOC)= PM(LOC) + CMPLX(BETA2 , CONST * C(L101,2))
      LOC    = IC + NF * (ILB3-1)
      PM(LOC)= PM(LOC) + CMPLX(BETA3 , CONST * C(L101,3))
101   CONTINUE
  402 CONTINUE
C
C........COMPLTE  INPUT  DOWNWASH
C
      DO 100 J100 = 1,NMOD
      XMOD1 = XMOD(ILB1,J100,IWING)
      XMOD2 = XMOD(ILB2,J100,IWING)
      XMOD3 = XMOD(ILB3,J100,IWING)
      SX1   = CMX1 * XMOD1
      SX2   = CMX2 * XMOD2
      SX3   = CMX3 * XMOD3
C
      TSX   = CMPLX(CMX1,CAPPA)
      VALUE                 = TSX * XMOD1 + SX2 + SX3
      INDWN(ILB1+IBUR,J100) = INDWN(ILB1+IBUR,J100)
     *                      + VALUE * FACTOR(ILB1)
C
      TSX   = CMPLX(CMX2,CAPPA)
      VALUE                 = TSX * XMOD2 + SX1 + SX3
      INDWN(ILB2+IBUR,J100) = INDWN(ILB2+IBUR,J100)
     *                      + VALUE * FACTOR(ILB2)
C
      TSX   = CMPLX(CMX3,CAPPA)
      VALUE                 = TSX * XMOD3 + SX1 + SX2
      INDWN(ILB3+IBUR,J100) = INDWN(ILB3+IBUR,J100)
     *                      + VALUE * FACTOR(ILB3)
100   CONTINUE
1090  CONTINUE
        IF( RESTAR  .LT.  0 )  GOTO 403
C
C........DO LOOP 2000 WILL COVEL ALL NODE POINTS
C        OF THE RECEIVING  ELEMENT(UNLESS PREVIOUSLY)
C
      DO 2000 IN = 1,3
      INODE = IN
      KPT   = ILE(IN)
        IF(PTUSED(KPT) .EQ. 1)  GO TO 2000
      PTUSED(KPT) = 1
      XR    = X( IN)
      YR    = Y( IN)
      ZR    = Z( IN)
        IF(  ISCLAT  .EQ.  0       )  GOTO 18
       IF (  IWING   .EQ.  1       )  GOTO 18
      IF  (  KPT     .NE.  NEXTR1  )  GOTO 18
C
C         SPECIAL  CASE  DURING  EXTRAPOLATION  PROCEEDURE   *ONLY*
C
      XR    = (XYZ(1,NEXTR1,2) + XYZ(1,NEXTR2,2)) * 0.50
      GOTO 24
   18 CONTINUE
        IF( MLINE1  .EQ.  0 )  GO TO 24
C******
```

```
C-------                                                                00602370
      DO 20 J20=1,MLINE1                                                00602380
      IF(KPT .EQ. MLINE(J20,IWING))  GO TO 22                           00602390
   20 CONTINUE                                                          00602400
      GO TO 24                                                          00602410
C-----------------------------------------------------------------------00602420
C         *** LOGIC  IS  DIAPHRAGM  DEPENDENT ***                       00602430
C              DUE TO SEQUENTIAL GRID-GEN  IN  'DIAPH3'  WHEN NDIV.GT.1 00602440
C              TAKING NEXT POINT BACK IS  EASY.    OTHERWISE  MUST GET  00602450
C              FROM CORRESPONDING  ELEMENT  IN  'LND-ARRAY'             00602460
   22 KPT1  = KPT  - 1                                                  00602470
         IF( NDIV .LE. 1 )  KPT1 = LND(J20+1,IWING)                     00602480
      XR     = ( XYZ(1,KPT,IWING)  +  XYZ(1,KPT1,IWING) )  * 0.500      00602490
      YR     = ( XYZ(2,KPT,IWING)  +  XYZ(2,KPT1,IWING) )  * 0.500      00602500
C-----------------------------------------------------------------------00602510
 24   CONTINUE                                                          00602520
C                                                                       00602530
C........BEGIN SEARCH THRU INFLUINCING ELEMENTS.  BOTH                  00602540
C        WINGS MUST BE COVERED                                          00602550
C                                                                       00602560
      IBP3    = -15                                                     00602570
      DO 3 KWING=1,2                                                    00602580
      KW       = KWING                                                  00602590
      IBP3    = IBP3 + 15                                               00602600
      NELT    = KCNTRL(IBP3 +  1)                                       00602610
      ISYMC   = KCNTRL(IBP3 + 10)                                       00602620
      XK      = XKVEC(KWING)                                            00602630
      XKOM    = XK / XMACH                                              00602640
      ISW     = 2                                                       00602650
      IF( IWING .NE. KWING   .AND.  KPT .LE. NP ) ISW = 1               00602660
C                                                                       00602670
C       COMPUTE DOWNWASH (IWS-1) ONLY IF  RECEIVING NODE  POINT         00602680
C       LIES  ON  WING   AND  INFLUENCING  ELEMENT  ON  OPPOSITE  WING  00602690
C                                                                       00602700
      DO 3000 M=1,NELT                                                  00602710
      MINFL    = M                                                      00602720
      IMB(1)  = IBLN(1,M,KWING)                                         00602730
      IMB(2)  = IBLN(2,M,KWING)                                         00602740
      IMB(3)  = IBLN(3,M,KWING)                                         00602750
      XI (1)  = XYZ(1,IMB1,KWING)                                       00602760
      YI (1)  = XYZ(2,IMB1,KWING)                                       00602770
      ZI (1)  = XYZ(3,IMB1,KWING)                                       00602780
      XI (2)  = XYZ(1,IMB2,KWING)                                       00602790
      YI (2)  = XYZ(2,IMB2,KWING)                                       00602800
      ZI (2)  = XYZ(3,IMB2,KWING)                                       00602810
      XI (3)  = XYZ(1,IMB3,KWING)                                       00602820
      YI (3)  = XYZ(2,IMB3,KWING)                                       00602830
      ZI (3)  = XYZ(3,IMB3,KWING)                                       00602840
      ZZ      =  ( ZI(1) + ZI(2) + ZI(3) )  / 3.0                       00602850
C                                                                       00602860
      POT1 = (0.0,0.0)                                                  00602870
      POT2 = (0.0,0.0)                                                  00602880
      POT3 = (0.0,0.0)                                                  00602890
      DWN1 = (0.0,0.0)                                                  00602900
      DWN2 = (0.0,0.0)                                                  00602910
      DWN3 = (0.0,0.0)                                                  00602920
      SWITCH = 1                                                        00602930
      PMULT  = 1.0                                                      00602940
        GO TO 301                                                       00602950
      YI(1)   = -YI(1)                                                  00602970
      YI(2)   = -YI(2)                                                  00602980
      YI(3)   = -YI(3)                                                  00602990
      PMULT   = SYMC                                                    00603000
301   CONTINUE                                                          00603010
C                                                                       00603020
C........COMPUTE  'R'  LOCATION.  I.E.   IS ELEMENT                     00603030
C         WITHIN  THE MACH CONE                                         00603040
C                                                                       00603050
      DO 28 J28=1,3                                                     00603060
      RDX = XR - XI(J28)                                                00603070
      SDX = VEETA* SQRT( (YR-YI(J28))**2 +  (ZR-ZI(J28))**2 )           00603080
      R1  = RDX - SDX                                                   00603090
      R2  = RDX + SDX                                                   00603100
        IF( R1 .GT. 0.0   .AND.  R2 .GT. 0.0 )  GO TO 30                00603110
   28 CONTINUE                                                          00603120
C                                                                       00603130
C       HERE IF  ALL NODES  OUTSIDE  MACH  CONE                         00603140
C                                                                       00603150
      GO TO( 3000,303 ), SWITCH                                         00603160
30    CONTINUE                                                          00603170
         IF( KPRINT  .GT.  0)                                           00603180
     *WRITE(OUT,1010) IWING,L,IN,KWING,M, XR,YR,ZR, R1,R2,SDX,          00603190
     *               (XI(JZR), YI(JZR), ZI(JZR), JZR=1,3)               00603200
      CALL  ORIENT                                                      00603210
     *   (XI,YI,A12,A13,B12,B13, CASE,                                  00603220
     *    X1,X2,X3,A23,B23,Y1,Y2,Y3 )                                   00603230
      CALL  TRINTX                                                      00603240
     * (X1,X2,X3,A12,A13,A23,B12,B13,B23,CASE,ZZ)                       00603250
         IF( KCN95 .EQ. 0 ) GO TO 3040                                  00603260
      KCN95 = 0                                                         00603270
      GO TO 302                                                         00603280
3040  CONTINUE                                                          00603290
      POT1 = POT1 + PMULT * FPOT(1)                                     00603300
      POT2 = POT2 + PMULT * FPOT(2)                                     00603310
      POT3 = POT3 + PMULT * FPOT(3)                                     00603320
        GO TO ( 304,302 ), ISW                                          00603330
304   CONTINUE                                                          00603340
      DWN1 = DWN1 + PMULT * DWN(1)                                      00603350
      DWN2 = DWN2 + PMULT * DWN(2)                                      00603360
      DWN3 = DWN3 + PMULT * DWN(3)                                      00603370
      CALL  ORIENT                                                      00603380
     *  ( YI,XI,A12,A13,B12,B13, CASE, Y1,Y2,Y3,A23,B23,                00603390
     *    X1,X2,X3 )                                                    00603400
      CALL  DWASH Y                                                     00603410
     * (Y1,Y2,Y3,A12,A13,A23,B12,B13,B23,CASE, ZZ)                      00603420
      DWN1 = DWN1 - PMULT * DWNI(1)                                     00603430
      DWN2 = DWN2 - PMULT * DWNI(2)                                     00603440
      DWN3 = DWN3 - PMULT * DWNI(3)                                     00603450
302   GO TO( 300,303 ), SWITCH                                          00603460
303   CONTINUE                                                          00603470
      WRITE(NTAPE)  REC1                                                00603480
        GO TO ( 305,3000 ), ISW                                         00603490
305   POT1 = DWN1                                                       00603500
      POT2 = DWN2                                                       00603510
      POT3 = DWN3                                                       00603520
      WRITE(MTAPE)  REC1                                                00603530
3000  CONTINUE                                                          00603540
   3  CONTINUE                                                          00603550
```

```
 403 CONTINUE                                                          00603570
1000  CONTINUE                                                         00603580
         IF( RESTAR  .LT.  0 )  GOTO 404                               00603590
      WRITE(MTAPE)  IM20, (IZR,JZR=1,13)                               00603600
      ISW = 1                                                          00603610
      CALL  TINCLT( PP, NP, NP, 4HPMAT, ISW, LTAPE )                   00603620
  404 CONTINUE                                                         00603630
  1   CONTINUE                                                         00603640
         IF( RESTAR  .LT.  0 )  GOTO 405                               00603650
      WRITE(NTAPE)  IM20, (IZR,JZR=1,13)                               00603660
      REWIND  LTAPE                                                    00603670
  405 CONTINUE                                                         00603680
      RETURN                                                           00603690
1010  FORMAT(1H0, 5I5, 2(4X,3E12.4) / 3(4X,3E12.4 / ) )                00603700
      END                                                              00603710
      SUBROUTINE  TRINTX                                                00700010
     *        (X1,X2,X3,A12,A13,A23,B12,B13,B23,CASE,Z)                 00700020
C                                                                       00700030
C........COMBINED  'X' - INTEGRATION ROUTINE                             00700040
C                                                                       00700050
      COMPLEX      POT1              ,POT2            ,POT3             00700060
     1            ,DWN1              ,DWN2            ,DWN3             00700070
     2            ,SUM1              ,SUM2            ,SUM3             00700080
     3            ,ARG               ,PREM            ,PREM1            00700090
     4            ,PREM2             ,FPOT   (   3   ) ,DWNX  (   3   ) 00700100
     5            ,DWN   (   3   )   ,POT    (   3   ) ,DUM   (   3   ) 00700110
      INTEGER      CUT               ,CASE                              00700120
      REAL         MU1               ,MU2             ,MU               00700130
     1            ,NUL               ,NUU             ,KAPPA            00700140
     2            ,KOM               ,MACH                              00700150
      COMMON      /     BASIC      /  KON(98)         ,IERR             00700160
      COMMON      /     TRINT      /  FPOT            ,DWNX             00700170
     1            ,CUM               ,U      (   5   ) ,C     (   5   ) 00700180
     2            ,X     (   3   )   ,Y      (   3   ) ,ZZ    (   3   ) 00700190
     3            ,T     (  3,3  )   ,XR              ,YR               00700200
     4            ,ZR                ,KAPPA           ,MACH             00700210
     5            ,VEETA             ,PI              ,KOM              00700220
     6            ,VESQ              ,C1OPIV                            00700230
      EQUIVALENCE                                                       00700240
     1             (DWN(1)  ,DWN1   ) ,(DWN(2)  ,DWN2 ) ,(DWN(3)  ,DWN3 ) 00700250
     2            ,(POT(1)  ,POT1   ) ,(POT(2)  ,POT2 ) ,(POT(3)  ,POT3 ) 00700260
     3            ,(KON(58) ,OUT    ) ,(KON(64),KPRINT) ,(KON(81) ,ISW  ) 00700270
     4            ,(KON(85) ,EPS    ) ,(KON(95),KON95 )                   00700280
C                                                                       00700290
          IF(KPRINT .GT. 0 )  WRITE(OUT,1000)                           00700300
     *       X1,X2,X3, CASE, A12,B12, A13,B13, A23,B23                  00700310
C                                                                       00700320
      EPS1   = 0.10 * EPS                                               00700330
      POT1   = (0.0,0.0)                                                00700340
      POT2   = (0.0,0.0)                                                00700350
      POT3   = (0.0,0.0)                                                00700360
      DXM    = (X3 - X1) * 0.500                                        00700370
      DXP    = (X3 + X1) * 0.500                                        00700380
      DELZ   = ZR - Z                                                   00700390
      D2MU   = VEETA * DELZ                                             00700400
          IF( ISW .EQ. 2 )  GOTO 3                                      00700410
    2 CONTINUE                                                          00700420
      DWN1   = (0.0,0.0)                                                00700430
      DWN2   = (0.0,0.0)                                                00700440
      DWN3   = (0.0,0.0)                                                00700450
      DZSQ   = DELZ * DELZ                                              00700460
      SLOPEL= A12                                                       00700470
      SLOPEU= A13                                                       00700480
    3 CONTINUE                                                          00700490
C                                                                       00700500
      DET    =  X(1) * (Y(2) - Y(3))                                    00700510
     *         +X(2) * (Y(3) - Y(1))                                    00700520
     *         +X(3) * (Y(1) - Y(2))                                    00700530
      KON95 =1                                                          00700540
          IF( ABS(DET) .LT.  1.00E-06 )  GO TO 52                       00700550
      DET    = 1.0/DET                                                  00700570
      T(1,1)= (Y(2) - Y(3)) * DET                                       00700580
      T(1,2)= (Y(3) - Y(1)) * DET                                       00700590
      T(1,3)= (Y(1) - Y(2)) * DET                                       00700600
      T(2,1)= (X(3) - X(2)) * DET                                       00700610
      T(2,2)= (X(1) - X(3)) * DET                                       00700620
      T(2,3)= (X(2) - X(1)) * DET                                       00700630
      T(3,1)= (X(2)*Y(3) - X(3)*Y(2)) * DET                             00700640
      T(3,2)= (X(3)*Y(1) - X(1)*Y(3)) * DET                             00700650
      T(3,3)= (X(1)*Y(2) - X(2)*Y(1)) * DET                             00700660
C                                                                       00700670
      IBR    = 1                                                        00700680
      DO 20 I=1,5                                                       00700690
      XI     = DXM * U(I) + DXP                                         00700700
      DELX   = XR - XI                                                  00700710
      MU1    = DELX - D2MU - EPS                                        00700720
      MU2    = DELX + D2MU - EPS                                        00700730
         IF(MU1 .LT. 0.0    .OR.    MU2 .LT. 0.0)  GOTO 20              00700740
         IF( CASE .EQ. 2 )  GOTO 22                                     00700750
         IF(XI .LE. X2 )  GOTO 22                                       00700760
         IF( IBR .LT. 2   .AND.   KPRINT .GT. 0 ) WRITE(OUT,102)        00700770
      IBR    = 3                                                        00700780
      A12    = A23                                                      00700790
      B12    = B23                                                      00700800
      SLOPEL= A12                                                       00700810
   22 YL     = A12 * XI  +  B12                                         00700820
      YU     = A13 * XI  +  B13                                         00700830
         IF(YU - YL) 23,25,25                                           00700840
   23 TEMP   = YU                                                       00700850
      YU     = YL                                                       00700860
      YL     = TEMP                                                     00700870
         IF( ISW .EQ. 2 )  GOTO 25                                      00700880
      TEMP   = SLOPEU                                                   00700890
      SLOPEU= SLOPEL                                                    00700900
      SLOPEL= TEMP                                                      00700910
   25 CONTINUE                                                          00700920
      TELX   = DELX - EPS                                               00700930
      MU     = SQRT( TELX*TELX - D2MU*D2MU )                            00700940
      NUL    = VEETA * (YR - YL)                                        00700950
      NUU    = VEETA * (YR - YU)                                        00700960
      RL     = MU - NUL                                                 00700970
      RU     = MU - NUU                                                 00700980
         IF(RL  .LT. 0.0   .AND.     RU .LT. 0.0)  GOTO 20              00700990
        IF (RL  .LT. 0.0                       )  YL  =  YR - MU/VEETA  00701000
      SL     = MU + NUL                                                 00701010
      SU     = MU + NUU                                                 00701020
         IF(SL  .LT. 0.0   .AND.     SU .LT. 0.0)  GOTO 20              00701030
        IF (SL  .LT. 0.0                       )  YU  =  YR + MU/VEETA  00701040
C                                                                       00701050
      DYM    = (YU - YL) * 0.500                                        00701060
      DYP    = (YU + YL) * 0.500                                        00701070
      ARG    = CMPLX(  0.0   ,  KAPPA * DELX  )                         00701080
      PREM   = C(I) * CEXP( -ARG )                                      00701090
      CALL  PSI(  XI, YU,  Z, SEIU  )                                   00701100
      CALL  PSI(  XI, YL,  Z, SEIL  )                                   00701110
      DSEI   = SEIU - SEIL                                              00701120
      POT1   = POT1 + PREM * XI * DSEI                                  00701130
      POT2   = POT2 + PREM * (YU*SEIU - YL*SEIL )                       00701140
      POT3   = POT3 + PREM * DSEI                                       00701150
      DXSQ  = DELX * DELX                                          00701180
      PREM1 = PREM / DELX                                          00701190
      PREM2 = (1.D0 + ARG) / DELX                                  00701200
      DSEI  = -DSEI                                                00701210
      SUM1  = DSEI * ( 1.0  +  XI * PREM2 )                        00701220
      SUM2  = PREM2 * ( YL*SEIL - YU*SEIU )                        00701230
      SUM3  = PREM2 * DSEI                                         00701240
      DWN1  = DWN1 + PREM1 * SUM1                                  00701250
      DWN2  = DWN2 + PREM1 * SUM2                                  00701260
      DWN3  = DWN3 + PREM1 * SUM3                                  00701270
C                                                                  00701280
   32 CONTINUE                                                     00701290
      SUM   = 0.0                                                  00701300
      DO 35 J=1,5                                                  00701310
      YJ    = DYP * U(J)  +  DYP                                   00701320
      CALL  PSI(  XI, YJ,  Z, SEIJ )                               00701330
      SUM   = SUM + C(J) * SEIJ                                    00701340
  35  CONTINUE                                                     00701350
      PREM  = SUM * DELTY * PREM                                   00701360
      POT2  = POT2 - PREM                                          00701370
 20   CONTINUE                                                     00701380
C                                                                  00701390
      CONST = (X1 - X3) * C1OPIV                                   00701400
      DO 40 J=1,3                                                  00701410
      FPOT(J) = (0.0,0.0)                                          00701420
      DO 41 I=1,3                                                  00701430
      FPOT(J) = FPOT(J)  +  POT(I) * T(I,J)                        00701440
   41 CONTINUE                                                     00701450
      FPOT(J) = CONST * FPOT(J)                                    00701460
   40 CONTINUE                                                     00701470
         IF( ISW .EQ. 2 )  GOTO 52                                 00701480
      CONST = -VEETA * DELZ * 2.0 * DXM / PI                       00701490
      DO 42 J=1,3                                                  00701500
      DWNX(J) = (0.0,0.0)                                          00701510
      DO 43 I=1,3                                                  00701520
      DWNX(J) = DWNX(J)  +  DWN(I) * T(I,J)                        00701530
   43 CONTINUE                                                     00701540
      DWNX(J) = CONST * DWNX(J)                                    00701550
   42 CONTINUE                                                     00701560
   52 RETURN                                                       00701570
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 00701580
102   FORMAT(7X,12HSWITCH  NOW   )                                 00701590
1000  FORMAT(//  16H ***SUB .TRINTX. / 3E18.6,I5/ 3(2E14.5,4X)/)   00701600
      END                                                          00701610
      SUBROUTINE  CWASHY                                                00800010
     1    (Y1,Y2,Y3,A12,A13,A23,B12,B13,B23,CASE,Z)                     00800020
      COMPLEX   ARG , ARGU, ARGL, SUMX, SUMY, PREM                      00800030
     *      , CMEGN, CWNY(3)                                            00800040
       COMPLEX    RCUM                                                  00800050
      INTEGER     CASE ,OUT                                             00800060
      REAL     MACH,KAPPA                                               00800070
      COMMON  /BASIC/ KCN(98),IERR                                      00800080
      COMMON  /TRINT/ RCUM(6),CWNY   ,U(5),C(5),X(3),Y(3),ZZ(3)         00800090
     *          , T(3,3),XR,YR,ZR,KAPPA,MACH,VEETA, PI                  00800100
      EQUIVALENCE                                                       00800110
     *          (KCN(58),OUT   )                                        00800120
     *        ,(KCN(66),KPRINT)                                         00800130
     *          ,(KCN(81),ISW   )                                       00800140
     *          ,(KCN(85),EPS   )                                       00800150
        IF(KPRINT .GT. 0 )  WRITE(OUT,1000)                             00800160
     *     Y1,Y2,Y3, CASE, A12,B12, A13,B13, A23,B23                    00800170
      NG = 5                                                            00800180
C                                                                       00800190
    5 CONTINUE                                                          00800200
      DELZ  =     (ZR - Z)                                              00800210
      EPS1 = 0.10*EPS                                                   00800220
      DVZ   = VEETA *   ABS(DELZ)                                       00800230
      OMEGN = (0.0,0.0)                                                 00800240
      CONST  = -VEETA * DELZ / PI                                       00800250
      DYM  = (Y3 - Y1) / 2.0                                            00800260
      DYP  = (Y3 + Y1) / 2.0                                            00800270
      IBR  = 1                                                          00800280
      IGO = 1                                                           00800290
      DO 20 I=1,NG                                                      00800300
      YI   = DYM * U(I) + DYP                                           00800310
         IF( CASE .EQ. 2 )  GOTO 22                                     00800320
      IF( YI .LE. Y2 )  GO TO 22                                        00800330
         IF( IBR .LT. 2  .AND.  KPRINT .GT.  0 )  WRITE(OUT,102)        00800340
      IBR   = 3                                                         00800350
      A12  = A23                                                        00800360
      B12  = B23                                                        00800370
   22 XL   = A12 * YI + B12                                             00800380
      XU   = A13 * YI + B13                                             00800390
        IF(XU - XL) 23,24,24                                            00800400
   23 TEMP = XU                                                         00800410
      XU   = XL                                                         00800420
      XL   = TEMP                                                       00800430
   24 CONTINUE                                                          00800440
      DXL   = XR - XL                                                   00800450
      DVY   = VEETA * (YR - YI)                                         00800460
      XMU   = DXL - DVZ                                                 00800470
         IF(XMU .LT. EPS)  GO TO 20                                     00800480
      XMU   = SQRT( DXL*DXL - DVZ*DVZ )                                 00800490
      RL    = XMU - DVY                                                 00800500
      SL    = XMU + DVY                                                 00800510
         IF( RL .LT. EPS1  .OR.   SL .LT. EPS1)  GOTO 20                00800520
      DXU   = XR - XU                                                   00800530
      XMU   = DXL - DVZ                                                 00800540
         IF(XMU .LT. EPS)  GO TO 33                                     00800550
      SU     = XML + DVY                                                00800580
         IF( RL .GT. 0.0   .AND.   SU .GT. 0.0 )  GOTO 34               00800590
   33 XU     = XR - SQRT( DVZ*DVZ + DVY*DVY )                           00800600
   34 CONTINUE                                                          00800610
      DXM    = (XL - XL) *0.50                                          00800620
      DXP    = (XL + XL) *0.50                                          00800630
      DXU    = (XR - XU)                                                00800640
      DXL    = (XR - XL)                                                00800650
      SUMY   = (0.0,0.0)                                                00800660
      ARGU   = CMPLX( 0.0, KAPPA * DXU )                                00800670
      ARGL   = CMPLX( 0.0, KAPPA * DXL )                                00800680
      CALL  PSI                                                         00800690
     *      ( XU, YI, Z, SEIU )                                         00800700
      SEIU   = SEIL / DXU                                               00800710
      CALL  PSI                                                         00800720
     *      ( XL, YI, Z, SEIL )                                         00800730
      SEIL   = SEIL / DXL                                               00800740
      SUMX   = SEIU * CEXP(-ARGU)  -  SEIL * CEXP(-ARGL)                00800750
      DO 25 J=1,NG                                                      00800760
      XJ     = DXM * U(J) + DXP                                         00800770
      DELX   = XR - XJ                                                  00800780
      DARG   = KAPPA * DELX                                             00800790
      ARG    = CMPLX( 0.0, DARG )                                       00800800
      PREM   = C(J) * CEXP(-ARG)                                        00800810
      ARG    = 1.C0 + ARG                                               00800820
      PREM   = PREM * ARG                                               00800830
      CALL  PSI                                                         00800840
     *      ( XJ, YI, Z, SEIJ )                                         00800850
      SEIJ   = SEIJ / (DELX * DELX)                                     00800860
      SUMY   = SUMY + PREM * SEIJ                                       00800870
   25 CONTINUE                                                          00800880
      SUMY   = SUMY * 2.0*DXM                                           00800890
      OMEGN  = OMEGN +(SUMX - SUMY)*C(I)*2.0*DYM                        00800900
   20 CONTINUE                                                          00800910
      SUMX     = OMEGN * CONST                                          00800920
      DWNY(1) = SUMX * T(2,1)                                           00800930
      DWNY(2) = SUMX*T(2,2)                                             00800940
      DWNY(3) = SUMX*T(2,3)                                             00800950
      RETURN                                                            00800960
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 00800970
 102  FORMAT(7X,12HSWITCH  NOW  )                                       00800980
1000  FORMAT(//   16H ***SUB .DWASHY. / 3E18.6,I5/ 3( 2E14.5,4X)/)      00800990
      END                                                               00801000
```

DECK 010

```
      SUBROUTINE  CRIENT                                                01000010
     *          ( X       ,Y        ,A12     ,A13     ,B12    ,B13      01000020
     *          ,CASE     ,X1       ,X2      ,X3      ,A23    ,B23      01000030
     *          ,Y1       ,Y2       ,Y3      )                          01000040
      INTEGER    CASE               ,OUT                                01000050
      DIMENSICN  X      (   3   ) ,Y     (   3   )                      01000060
      COMMON     /    BASIC     /  KONTRL(  100   )                     01000070
      EQUIVALENCE                                                       01000080
     1           (KONTRL(58),OUT    )                                   01000090
     2          ,(KONTRL(63),KPRINT)                                    01000100
      X1 = X(1)                                                         01000110
      X2 = X(2)                                                         01000120
      X3 = X(3)                                                         01000130
      Y1 = Y(1)                                                         01000140
      Y2 = Y(2)                                                         01000150
      Y3 = Y(3)                                                         01000160
      NORD1 = 3                                                         01000170
      NORD2 = 2                                                         01000180
      NORD3 = 1                                                         01000190
      K = 1                                                             01000200
      J = 0                                                             01000210
      XMIN = X3                                                         01000220
      XMAX = X1                                                         01000230
      YMAX = Y1                                                         01000240
      YMIN = Y3                                                         01000250
      IF(XMAX - X3) 19,18,20                                            01000260
18     IF(YMAX.GT.Y3) GO TO 20                                          01000270
19    CONTINUE                                                          01000280
      YMAX = Y3                                                         01000290
      YMIN = Y1                                                         01000300
      XMAX = X3                                                         01000310
      NORD1 = 1                                                         01000320
      NORD3 = 3                                                         01000330
      K = 0                                                             01000340
      XMIN = X1                                                         01000350
   20 IF(XMAX  -  X2) 22,21,30                                          01000360
21     IF(YMAX.GT.Y2) GO TO 30                                          01000370
22    CONTINUE                                                          01000380
      YMAX = Y2                                                         01000390
      NORD3 = 2                                                         01000400
      J = +1                                                            01000410
        IF(K .EC. 1) J = -1                                             01000420
      XMAX = X2                                                         01000430
30    IF(XMIN - X2) 39,37,38                                            01000440
37    IF(YMIN.LT.Y2) GO TO 39                                           01000450
38    YMIN = Y2                                                         01000460
      NORD1 = 2                                                         01000470
      XMIN = X2                                                         01000480
      J   = +1                                                          01000490
        IF(K .EC. 0) J=-1                                               01000500
39    CONTINUE                                                          01000510
      NORD2 =    2 + J                                                  01000520
40    CONTINUE                                                          01000530
         IF( KPRINT  .GT.  0)                                           01000540
     *  WRITE(OUT,1000)  NORD1,NORD2,NORD3                              01000550
```

```
      X3 = X(NCRD3  )                                                   01000570
      X2 = X(NCRD2  )                                                   01000580
      X1 = X(NCRD1  )                                                   01000590
      Y1 = Y(NCRD1  )                                                   01000600
      Y2 = Y(NCRD2  )                                                   01000610
      Y3 = Y(NCRD3  )                                                   01000620
C   SLOPE OF LINE 1 - 3                                                 01000630
      A13 = (Y3 - Y1) / (X3 - X1)                                       01000640
      B13 = Y3 - A13 * X3                                               01000650
      A23 = 0.0                                                         01000660
      B23 = 0.0                                                         01000670
      CASE = 1                                                          01000680
      IF(X1 - X2) 11,10,11                                              01000690
10    CASE = 2                                                          01000700
      A12 = (Y3 - Y2) / (X3 - X2)                                       01000710
      B12 = Y3 - A12 * X3                                               01000720
        GO TO 15                                                        01000730
11    A12 = (Y2 - Y1) / (X2 - X1)                                       01000740
      B12 = Y2 - A12 * X2                                               01000750
      DIVR = X3 - X2                                                    01000760
        IF(DIVR) 14,15,14                                               01000770
14    A23 = (Y3 - Y2) / DIVR                                            01000780
      B23 = Y3 - A23 * X3                                               01000790
15    CONTINUE                                                          01000800
99    CONTINUE                                                          01000810
      RETURN                                                            01000820
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 01000830
1000  FORMAT(//    16H ***SUB .ORIENT.  / 5X, 6HORDER , 3I3  )          01000840
      END                                                               01000850
```

DECK 011

```
      SUBROUTINE PSI                                                    01100010
     *    ( XJ, YJ, Z, SEI )                                            01100020
      COMPLEX    CDUM                                                   01100030
      INTEGER    OUT                                                    01100040
      REAL       RR     ( 101    ) ,JAPROX(     1.  ) ,J      ( 101   ) 01100050
     1           ,NU              ,KAPPA             ,MU                01100060
     2           ,MACH            ,L                 ,LAMBDA            01100070
      COMMON     /     BASIC     /  KON   (100      )                   01100080
      COMMON     /     TRINT     /  COUM  (    9     ) ,RDUM  (    28  ) 01100090
     1           ,XR              ,YR                ,ZR                01100100
     3           ,KAPPA           ,MACH              ,VEETA             01100110
     4           ,PI              ,XKOM              ,VSQRD             01100120
      EQUIVALENCE                                                       01100130
     1            (KON(58)   ,OUT    )                                  01100140
     2           ,(KON(65)   ,KPRINT )                                  01100150
     3           ,(KON(97)   ,DEPS   )                                  01100160
     4           ,(KON(99)   ,IERR   )                                  01100170
      DATA        D1, D12, D17, EPS                                     01100180
     *           /0.12899220E+02   ,0.64496099E+01   ,0.94907155E+01    01100190
     *           ,0.50000000E-05   /                                    01100200
C---------------------------------------------------------------------- 01100210
C                                                                       01100220
C         DATA  FOR  D1, D12, D17, EPS      BASED  ON ID = 5            01100230
C         D1   =  LOG-E(10)  * ID    +  2*LOG-E(2)                      01100240
C         D12 =  D1 / 2                                                 01100250
C         D17 =  2/E * D1                                               01100260
C         S       CONSTANT = E/2                                        01100270
C                                                                       01100280
C         ALGORITHM  SHOWS  2 PASSES  REQUIRED,  WITH  DELTA-KNU = 5    01100290
C         TEST  SHOW  THIS  IS  NOT  NEEDED                             01100300
C                                                                       01100310
C         ARCSIN  =  CONSTANT  =  PI / 2                                01100320
C                                                                       01100330
C---------------------------------------------------------------------- 01100340
      IER    = 0                                                        01100350
      SIGN   = +1.0                                                     01100360
      NU     = VEETA * (YR - YJ)                                        01100370
         IF( NU ) 2,3,3                                                 01100380
    2 SIGN   = -1.0                                                     01100390
      NU     = -NU                                                      01100400
    3 MU     = (XR - XJ)*(XR - XJ) - (VEETA * (ZR - Z))**2              01100410
         IF( MU ) 4,4,5                                                 01100420
    4 CONTINUE                                                          01100430
      SEI    = 0.0                                                      01100440
        GOTO 999                                                        01100450
    5 MU     = SQRT( MU )                                               01100460
         IF( NU .GE. MU )  GOTO 6                                       01100470
      ARCSIN= ASIN( NU / MU )                                           01100480
        GOTO 7                                                          01100490
    6 ARCSIN= 1.5707963268                                              01100500
    7 ARCSIN= SIGN * ARCSIN                                             01100510
C                                                                       01100520
        X    = XKOM * MU                                                01100530
C                                                                       01100540
      NMAX   = IFIX(X)*2 + 6                                            01100550
      WRITE(OUT,9001)  (KON(N),N=75,81), NMAX                           01100570
      IER    = 1                                                        01100580
      NMAX = 100                                                        01100590
    8    IF( X ) 9,10,11                                                01100600
    9 CALL  CRASH                                                       01100610
      WRITE(OUT,9002)  X, (KON(N),N=75,81)                              01100620
      CALL  CCREC                                                       01100630
C                                                                       01100640
   10 SEI    = ARCSIN                                                   01100650
        GOTO 999                                                        01100660
   11 CONTINUE                                                          01100670
      NMAX1 = NMAX + 1                                                  01100680
      KBR    = 0                                                        01100690
      Y      = D12 / NMAX                                               01100700
C-----------------------------------------------------------------------01100710
   15 CONTINUE                                                          01100720
         IF( Y .GT. 10.0 ) GOTO 16                                      01100730
      P      = 0.579541E-04 * Y    -  0.176148E-02                      01100740
      P      =   Y * P             +  0.208645E-01                      01100750
      P      =   Y * P             -  0.129013                          01100760
      P      =   Y * P             +  0.85777                           01100770
      T      =   Y * P             +  1.0125                            01100780
        GOTO 17                                                         01100790
   16 CONTINUE                                                          01100800
      W      =  ALOG(Y) - 0.775                                         01100810
      P      = (0.775 - ALOG(W)) / (1.0 + W)                            01100820
      P      = Y / (1.0 + P)                                            01100830
      T      = P / W                                                    01100840
C-----------------------------------------------------------------------01100850
   17 CONTINUE                                                          01100860
         IF( KBR .EQ. 1 )  GOTO18                                       01100870
      KBR    = 1                                                        01100880
      R      = NMAX * T                                                 01100890
      Y      = D17 / X                                                  01100900
        GOTO 15                                                         01100910
   18 CONTINUE                                                          01100920
      S      = 1.3591409 * X * T                                        01100930
      KNU    = IFIX( S )                                                01100940
         IF( R .GT. S )  KNU = IFIX( R )                                01100950
      KNU    = KNU + 6                                                  01100960
   20 CONTINUE                                                          01100970
      M      = KNU / 2                                                  01100980
      L      = 1.0 / FLOAT(M+1)                                         01100990
      N      = M + M                                                    01101000
      R      = 0.0                                                      01101010
      S      = 0.0                                                      01101020
   21 CONTINUE                                                          01101030
      XN     = FLOAT( N )                                               01101040
      YDIV   = (XN + XN) / X    -    R                                  01101050
         IF( YDIV .EQ. 0.0 )  YDIV = 1.0E-06                            01101060
      R      = 1.0 / YDIV                                               01101070
      LAMBDA= 0.0                                                       01101080
         IF(( (N/2)*2 - N )  .NE. 0 )  GOTO 22                          01101090
      L      =  FLOAT( N+2 ) * L / XN                                   01101100
      LAMBDA=  XN * L                                                   01101110
   22 S      = R * (LAMBDA + S)                                         01101120
         IF( N      .LE.   NMAX )  RR( N )  =  R                        01101130
      N      = N - 1                                                    01101140
         IF( N      .GE.   1    )  GOTO 21                              01101150
```

```
C       CALCULATIONS   COMPLETE  (RATIO  J,N-1 / J,N+1)                   01101170
C******                                                                   01101180
      J(1)  = 1.CC /(1.0 + S)                                             01101190
      DO 24 N=1,NMAX                                                      01101200
   24 J(N+1) = RR(N) * J(N)                                               01101210
C-------------------------------------------------------------------------01101220
      SIGN  = +1.CCC                                                      01101230
      SEI   = J(1) * ARCSIN                                               01101240
      TWOR  = 0.CO                                                        01101250
      RSUB  = 0.CC                                                        01101260
      DO 2020 N=3,NMAX1,2                                                 01101270
      RSUB  = RSLB + 1.00                                                 01101280
      TWOR  = TWCR + 2.00                                                 01101290
      SIGN  = -SIGN                                                       01101300
      SUM   =  SIGN*  J(N) * SIN( TWOR * ARCSIN ) / RSUB                  01101310
      SEI   = SEI + SUM                                                   01101320
 2020 CONTINUE                                                            01101330
         IF( ABS( SUM )  .LE.  CEPS )  GOTO 999                           01101340
      IER   = 2                                                           01101350
      WRITE(OLT,1000)                                                     01101360
      WRITE(OLT,9003)  DEPS, SUM                                          01101370
  999 CONTINUE                                                            01101380
         IF( KPRINT  .GT.  0)                                             01101390
     *WRITE (CUT,1002) XJ,YJ,Z,NU,MU,X,SEI                                01101400
      IF( IER  .GT. 0 ) IERR=IERR+1                                       01101410
      RETURN                                                              01101420
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 01101430
 1000 FORMAT(// 12H***SUB .PSI. )                                         01101440
 1002 FORMAT(1X, 7E14.6 )                                                 01101450
 9001 CFORMAT(// 10X,40H** ** WARNING ** **    ACCURACY  LOST    , 7I5 /  01101460
     1          16X,16HNMAX  REQUEST    , I6 ,20H    RESET  TO  100     ) 01101470
 9002 FORMAT(1H1////2CX,70HFROGRAM  LOGIC  HAS  FAILED,   SUBROUTINE  PS01101480
     1I.     BESSEL  FNC  ARG = ,E18.6  /10X, 7I5    //                   01101490
     2    30X,   1CHH E L P ./     )                                      01101500
 9003 OFORMAT( 10X,21HCCNVERGENCE  NOT  MET , E14.6 ,  /                  01101510
     1         10X,20HLAST  TERM  ADDED      , E14.7  )                   01101520
      END                                                                 01101530
```

DECK 012

```
      SUBROUTINE  ASSPOT
     *          ( PHI      ,PHIXX    ,PHI22    ,T        ,NP1     ,NP2
     *           ,ND1      ,ND2      ,NTAPE    ,KTAPE   ,MTAPE   )
      COMPLEX  PHI(1), PHIXX(1), PHI22(1), T(1)
      INTEGER   OUT
      COMMON    /BASIC/ KON(98),IERR
C
      OUT = KON(58)
      KPRINT = KON(67)
      REWIND  KTAPE
      REWIND  NTAPE
      ND    = ND1 + ND2
      NP    = NP1 + NP2
      NPND  = NP * ND
      NPSQ  = NP * NP
      NDSQ  = ND * ND
         IF( ND .EQ. 0 )  GO TO 20
C
C.......ASSEMBLE  PHI-WD....ORDER NP*ND.....STORE KTAPE
C
      NSET  = 2
      DO 1 J=1,NPND
    1 PHIXX(J) = (0.0,0.0)
      CALL  ASSY(PHIXX,NP,NP1,NP2,ND1,NTAPE,NSET )
         IF(KPRINT .GT. 0 )  CALL  CMATPR(PHIXX ,NP,ND,4HP-WD)
      ISW    = 1
      CALL  TINCLT( PHIXX, NP,ND, 4HPHWD, ISW, KTAPE )
C
C.......ASSEMBLE  PHI-WW....ORDER NP*NP.....STORE KTAPE
C
      REWIND  NTAPE
   20 CONTINUE
      DO 2 J=1,NPSQ
    2 PHI  (J) = (0.0,0.0)
      NSET  = 1
      CALL  ASSY(PHI,NP,NP1,NP2,ND1,NTAPE,NSET )
         IF(KPRINT .GT. 0 )  CALL  CMATPR(PHI    ,NP,NP,4HP-WW)
      ISW    = 1
      CALL  TINCLT( PHI   , NP,NP, 4HPHWW, ISW, KTAPE )
         IF( ND .EQ. 0 )  GO TO 50
C
C.......ASSEMBLE  PHI-DD....ORDER ND*ND.....INVERT..STORE IN CORE
C
      REWIND  NTAPE
      DO 3 J=1,NDSQ
    3 PHI22(J) = (0.0,0.0)
      NSET  = 4
      CALL  ASSY(PHI22,ND,NP1,NP2,ND1,NTAPE,NSET )
         IF(KPRINT .GT. 0 )  CALL  CMATPR(PHI22 ,ND,ND,4HP-DD)
      CALL  CMINV(PHI22, ND, PHIXX(1), PHIXX(ND+1) )
         IF(KPRINT .GT. 0 )  CALL  CMATPR(PHI22, ND,ND,4HPINV)
C
C.......ASSEMBLE  PHI-DW....ORDER ND*NP.....STORE IN CORE
C
      REWIND NTAPE
      DO 4 J=1,NFND                                                     01200570
    4 PHIXX(J) = (0.0,0.0)                                              01200580
      CALL  ASSY(PHIXX,ND,NF1,NP2,ND1,NTAPE,NSET )                      01200590
         IF(KPRINT .GT. 0 )  CALL  CMATPR(PHIXX, ND,NP,4HP-DW)          01200600
C                                                                       01200610
C.......EVALUATE T(ND*NP) = PHI22-INV * PHI21                           01200620
C                                                                       01200630
      CALL  CMPRD(PHI22,PHIXX,T,ND,ND,NP)                               01200640
         IF(KPRINT .GT. 0 )  CALL  CMATPR(T      ,ND,NP,4HTMAT)         01200650
      ISW    = 1                                                        01200660
      CALL  TINOUT(  T    , ND,NP, 4HTMAT, ISW, KTAPE )                 01200670
      REWIND  KTAPE                                                     01200680
C                                                                       01200690
C.......RECALL  PHIWD  FROM  KTAPE....FIRST  SET  OF  RECORDS           01200700
C                                                                       01200710
      ISW    = 2                                                        01200720
      CALL  TINOUT( PHIXX, NP,ND, 4HPHWD, ISW, KTAPE )                  01200730
      CALL  CMPRD(PHIXX,T,PHI,NP,ND,NP)                                 01200740
C                                                                       01200750
C.......THIS  PHI  IS  REALLY  PHI-WD * T                               01200760
C                                                                       01200770
         IF(KPRINT .GT. 0 )  CALL  CMATPR(PHIXX ,NP,ND,4HPHWD)          01200780
         IF(KPRINT .GT. 0 )  CALL  CMATPR(PHI   ,NP,NP,4HPHI )          01200790
C                                                                       01200800
C.......NOW  RECALL  PHI-WW  ONE  COLUMN  AT  A  TIME                   01200810
C        (WE  HAVE  A  SIZE  PROBLEM  OTHERWISE)                        01200820
C                                                                       01200830
      ISW    = -2                                                       01200840
      CALL  TINOUT(  PHI  , NP,NF, 4HPHWW, ISW, KTAPE )                 01200850
      K2   = 0                                                          01200860
      DO 10 K10=1,NP                                                    01200870
      K1   = K2 + 1                                                     01200880
      K2   = K2 + NP                                                    01200890
      READ(KTAPE)  (PHIXX(L),L=1,NP)                                    01200900
      LL   = 0                                                          01200910
      DO 11 K11=K1,K2                                                   01200920
      LL   = LL + 1                                                     01200930
11    PHI(K11) = PHIXX(LL) - PHI(K11)                                   01200940
   10 CONTINUE                                                          01200950
C                                                                       01200960
C.......POSITION  TAPE  FOR  NEXT  READ  (DONE  AUTO.  IN  TINOUT       01200970
C                                                                       01200980
      READ(KTAPE)  KODE                                                 01200990
         IF( KODE  .EQ. -20 )  GOTO 50                                  01201000
      WRITE(OUT,9001)        KTAPE                                      01201010
      CALL  C R A S H                                                   01201020
      CALL  CCRED                                                       01201030
   50 CONTINUE                                                          01201040
      ISW    = 1                                                        01201050
      CALL  TINOUT( PHI   , NP,NF, 4HPHI , ISW, MTAPE )                 01201060
         IF(KPRINT .GT. 0 )  CALL  CMATPR(PHI   ,NP,NP,4HPHI )          01201070
      RETURN                                                            01201080
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  01201090
 9001 FORMAT(1H1 /// 10X, 42H*** ERROR  IN  READING  TRAILER  OF  ARRAY 01201100
     *                  , 20H  'PHWW' .,  TAPE   ,I4 )                  01201110
      END                                                               01201120
```

DECK 013

```
      SUBROUTINE  ASSY                                                  01300010
     *    ( PHI, N, NP1, NP2, ND1, NTAPE, NSET )                        01300020
      COMPLEX   PHI( N,1 ), P1, P2, P3                                  01300030
      INTEGER   BUR(2,2), BUC(2,2), TESTR(2,2), TESTC(2,2)              01300040
     1          ,BR, BC, TR, TC, ROW, COL1, COL2, COL3                  01300050
     2          ,RW, IW, RELM  , IELM                                   01300060
      REAL      RECORD(14)                                              01300070
      EQUIVALENCE (RECORD( 1) , KODE , KPT )                            01300080
     1            ,(RECORD( 2) , IB1        )                           01300090
     2            ,(RECORD( 3) , IB2        )                           01300100
     3            ,(RECORD( 4) , IB3        )                           01300110
     4            ,(RECORD( 5) , RW         )                           01300120
     5            ,(RECORD( 6) , IW         )                           01300130
     6            ,(RECORD( 7) , RELM       )                           01300140
     7            ,(RECORD( 8) , IELM       )                           01300150
     8            ,(RECORD( 9) ,  P1        )                           01300160
     9            ,(RECORD(11) ,  P2        )                           01300170
     A            ,(RECORD(13) ,  P3        )                           01300180
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 01300190
C         THIS  SUB-PROGRAM  ASSEMBLES  ARRAYS  OF  THE  FORM  'PHI-NN'.01300200
C         FOR THE  AIC / INT  PROGRAM,                                  01300210
C           NN     NSET                                                 01300220
C           WW      1                                                   01300230
C           WC       2                                                  01300240
C           DW       3                                                  01300250
C           DC       4                                                  01300260
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 01300270
C                                                                       01300280
      TESTR(1,1) = NP1                                                  01300290
      TESTR(1,2) = NP1                                                  01300300
      TESTR(2,1) = NP2                                                  01300310
      TESTR(2,2) = NP2                                                  01300320
      TESTC(1,1) = NP1                                                  01300330
      TESTC(1,2) = NP2                                                  01300340
      TESTC(2,1) = NP1                                                  01300350
      TESTC(2,2) = NP2                                                  01300360
        GO TO ( 10,20,30,40 ), NSET                                     01300370
10    CONTINUE                                                          01300380
      BUR  (1,1) = 0                                                    01300390
      BUR  (1,2) = 0                                                    01300400
      BUR  (2,1) = NP1                                                  01300410
      BUR  (2,2) = NP1                                                  01300420
      BUC  (1,1) = 0                                                    01300430
      BUC  (1,2) = NP1                                                  01300440
      BUC  (2,1) = 0                                                    01300450
      BUC  (2,2) = NP1                                                  01300460
      GO TC 5                                                           01300470
20    BUR  (1,1) = 0                                                    01300480
      BUR  (1,2) = 0                                                    01300490
      BUR  (2,1) = NP1                                                  01300500
      BUR  (2,2) = NP1                                                  01300510
      BUC  (1,1) = -NP1                                                 01300520
      BUC  (1,2) = -NP2 + ND1                                           01300530
      BUC  (2,1) = BUC(1,1)                                             01300540
      BUC  (2,2) = BUC(1,2)                                             01300550
```

```
30    BUR  (1,1) = -NP1                                                  01300570
      BUR  (1,2) = BUR(1,1)                                               01300580
      BUR  (2,1) = -NF2 + ND1                                             01300590
      BUR  (2,2) = BUR(2,1)                                               01300600
      BUC  (1,1) = 0                                                      01300610
      BUC  (1,2) = NP1                                                    01300620
      BUC  (2,1) = 0                                                      01300630
      BUC  (2,2) = NP1                                                    01300640
      GO TC 5                                                             01300650
40    BUR  (1,1) = -NP1                                                   01300660
      BUR  (1,2) = BUR(1,1)                                               01300670
      BUR  (2,1) = -NF2 + ND1                                             01300680
      BUR  (2,2) = BUR(2,1)                                               01300690
      BUC  (1,1) = BUR(1,1)                                               01300700
      BUC  (1,2) = BUR(2,1)                                               01300710
      BUC  (2,1) = BUR(1,1)                                               01300720
      BUC  (2,2) = BUR(2,1)                                               01300730
 5    CONTINUE                                                            01300740
    1 READ(NTAPE       ) RECCRD                                           01300750
      IF(ENDFILE NTAPE) 99,22                                             01300760
22    CONTINUE                                                            01300770
      IF( KCDE .EC. -20 )  GO TC 2                                        01300780
      TR  = TESTR(RW,IW)                                                  01300790
      TC  = TESTC(RW,IW)                                                  01300800
      BR  = BLR  (RW,IW)                                                  01300810
      BC  = BLC  (RW,IW)                                                  01300820
        GC TO ( 11,21,31,41 ), NSET                                       01300830
11    CONTINUE                                                            01300840
      IF( KPT .GT. TR )  GO TC 1                                          01300850
      ROW  = KPT + BR                                                     01300860
      COL1 = IB1 + BC                                                     01300870
      COL2 = IB2 + BC                                                     01300880
      COL3 = IB3 + BC                                                     01300890
      IF(IB1 .LE. TC) PHI(RCW,CCL1) = PHI(ROW,COL1) + P1                  01300900
      IF(IB2 .LE. TC) PHI(RCW,CCL2) = PHI(ROW,COL2) + P2                  01300910
      IF(IB3 .LE. TC) PHI(RCW,CCL3) = PHI(ROW,COL3) + P3                  01300920
        GC TO 1                                                           01300930
21    CONTINUE                                                            01300940
      IF( KPT .GT. TR )  GO TO 1                                          01300950
      ROW  = KPT + BR                                                     01300960
      COL1 = IB1 + BC                                                     01300970
      COL2 = IB2 + BC                                                     01300980
      COL3 = IB3 + BC                                                     01300990
      IF(IB1 .GT. TC) PHI(RCW,CCL1) = PHI(ROW,COL1) + P1                  01301000
      IF(IB2 .GT. TC) PHI(RCW,CCL2) = PHI(ROW,COL2) + P2                  01301010
      IF(IB3 .GT. TC) PHI(RCW,CCL3) = PHI(ROW,COL3) + P3                  01301020
        GC TO 1                                                           01301030
31    CONTINUE                                                            01301040
      IF( KPT .LE. TR )  GO TO 1                                          01301050
      ROW  = KPT + BR                                                     01301060
      COL1 = IB1 + BC                                                     01301070
      COL2 = IB2 + BC                                                     01301080
      COL3 = IB3 + BC                                                     01301090
      IF(IB1 .LE. TC) PHI(RCW,CCL1) = PHI(ROW,COL1) + P1                  01301100
      IF(IB2 .LE. TC) PHI(RCW,CCL2) = PHI(ROW,COL2) + P2                  01301110
      IF(IB3 .LE. TC) PHI(RCW,CCL3) = PHI(ROW,COL3) + P3                  01301120
        GO TO 1                                                           01301130
41    CONTINUE                                                            01301140
      IF( KPT .LE. TR )  GO TC 1                                          01301150
      COL1 = IB1 + BC                                          01301170
      COL2 = IB2 + BC                                          01301180
      CCL3 = IB3 + BC                                          01301190
      IF(IB1 .GT. TC) PHI(RCW,CCL1) = PHI(ROW,COL1) + P1       01301200
      IF(IB2 .GT. TC) PHI(RCW,CCL2) = PHI(RUW,COL2) + P2       01301210
      IF(IB3 .GT. TC) PHI(RCW,CCL3) = PHI(ROW,COL3) + P3       01301220
      GO TO 1                                                  01301230
    2 CONTINUE                                                 01301240
      RETURN                                                   01301250
   99 CONTINUE                                                 01301260
      STOP   7                                                 01301270
      END                                                      01301280
```

DECK 014

```
      SUBROUTINE  ASSChN                                                01400010
     *          ( h12      ,W21      ,A        ,WT       ,WORK   ,T      01400020
     *           ,INChN    ,NP       ,NP1      ,NP2      ,ND1    ,ND2    01400030
     *           ,ND       ,MTAPE    ,KTAPE    ,NMOD     ,NP2ND2 ,NP1ND1 01400040
     *           ,C        )                                            01400050
      COMPLEX     h12    ( NP1,1   ) ,W21    (  NP2,1 ) ,A     (  NP,1  ) 01400060
     1           ,hT     (  NF,1   ) ,WORK   (     1  ) ,T     (  ND,1  ) 01400070
     2           ,INChN  ( 100,1   ) ,C      (  NP,1  ) ,SUM             01400080
     3           ,ChN1               ,DWN2              ,DWN3            01400090
      REAL        FECCRD(    14    )                                    01400100
      INTEGER     CLT                                                   01400110
      COMMCN     /BASIC/ KCN(98),IERR                                   01400120
      EQUIVALENCE     (RECCRD(1), KODE, KPT )                           01400130
     1               ,(RECCRD(2), IB1       )                           01400140
     2               ,(RECCRC(3), IB2       )                           01400150
     3               ,(RECCRC(4), IB3       )                           01400160
     4               ,(RECCRD( 9), DWN1     )                           01400170
     5               ,(RECORD(11), DWN2     )                           01400180
     6               ,(RECCRD(13), DWN3     )                           01400190
C                                                                       01400200
      OUT = KCN(58)                                                     01400210
      KPRINT = KCN(68)                                                  01400220
      DO 2 K=1,NF2ND2                                                   01400230
      DO 2 J=1,NF1                                                      01400240
    2 W12(J,K) = (0.0,0.0)                                              01400250
      DO 4 K=1,NF1NC1                                                   01400260
      DC 4 J=1,NF2                                                      01400270
    4 W21(J,K) = (C.0,0.0)                                              01400280
C                                                                       01400290
      REWIND  MTAFE                                                     01400300
   10 READ(MTAPE         ) RECORD                                       01400310
      IF(KCDE .EC. -20)  GC TC 20                                       01400320
      W12(KPT,IB1) = h12(KPT,IB1) + DWN1                                01400330
      W12(KPT,IB2) = h12(KPT,IB2) + DWN2                                01400340
      W12(KPT,IB3) = h12(KPT,IB3) + DWN3                                01400350
        GC TO 10                                                        01400360
C-----------------------------------                                    01400370
   20 READ(MTAPE         ) RECORC                                       01400380
      IF(KCDE .EC. -20)  GO TO 30                                       01400390
      W21(KPT,IB1) = W21(KPT,IB1) + DWN1                                01400400
      W21(KPT,IB2) = h21(KPT,IB2) + DWN2                                01400410
      W21(KPT,IB3) = h21(KPT,IB3) + DWN3                                01400420
        GC TC 20                                                        01400430
C                                                                       01400440
C........FULLY  ASSEMBLED  COWNhASH  ARRAYS  W12,W21                    01400450
C                                                                       01400460
C-----------------------------------------------------------------------01400470
C                                                                       01400480
C........ASSEMBLE  WT/11   CRDER(NP1,ND1)                               01400490
C                                                                       01400500
   30 CONTINUE                                                          01400510
         IF(KPRINT .GT. 0 )  CALL  CMATPR(W12   ,NP1,NP2ND2,4HW-12)     01400520
         IF(KPRINT .GT. 0 )  CALL  CMATPR(W21   ,NP2,NP1ND1,4HW-21)     01400530
         IF( ND .EC. 0 )  GC TO 32                                      01400540
C                                                                       01400550
```

```
C                                                                       01400580
      ISW  = 2                                                          01400590
      CALL  TINCLT(  T  , ND,NP, 4HTMAT, ISW, KTAPE )                   01400600
      DO 2002 K=1,ND                                                    01400610
      DO 2002 J=1,NP                                                    01400620
      T(K,J) = 0.50*T(K,J)                                              01400630
 2002 WT(J,K) = (0.0,0.0)                                               01400640
         IF( ND1.EQ. 0 )  GO TO 321                                     01400650
C                                                                       01400660
C........NOW  STORE  WT/21  ORDER(NP2,ND1)                              01400670
C                                                                       01400680
      CALL  STORE(WT,W21(1,NP1+1),NP, NP2, NP1+1, 1, NP, ND1,  1)       01400690
C                                                                       01400700
  321 CONTINUE                                                          01400710
         IF( ND2.EQ. 0 )  GO TO 322                                     01400720
C                                                                       01400730
C........NOW  STORE  WT/12  ORDER(NP1,ND2)                              01400740
C                                                                       01400750
      CALL STORE(WT,W12(1,NP2+1),NP,NP1,1, ND1+1, NP1, ND,  4)          01400760
C                                                                       01400770
  322 CONTINUE                                                          01400780
C                                                                       01400790
      DO 2003 K=1,NP1                                                   01400800
      DO 2003 J=1,ND1                                                   01400810
 2003 T(J,K) = -1.00 * T(J,K)                                           01400820
      NEW1 = ND1+1                                                      01400830
      NEW2 = NP1+1                                                      01400840
      DO 2009 K=NEW2,NP                                                 01400850
      DO 2009 J=NEW1,ND                                                 01400860
 2009 T(J,K) = -1.00 * T(J,K)                                           01400870
      CALL  CMPRD(WT,T,A,NP,ND,NP)                                      01400880
         IF(KPRINT .GT. 0 )  CALL  CMATPR(WT  ,NP,ND,4HWTAS)            01400890
         IF(KPRINT .GT. 0 )  CALL  CMATPR(A   ,NP,NP,4HAWTT)            01400900
         IF(KPRINT .GT. 0 )  CALL  CMATPR(T   ,ND,NP,4HGAMM)            01400910
      GO TO 38                                                          01400920
C-----------------------------------------------------------------------01400930
C........WHEN  NO  DIAPHRAGM  EXISTS,  MUST  ZERO  'A-ARRAY'             01400940
C        OTHERWISE  IT IS POPULATED  THRU MATRIX  PRODUCT               01400950
   32 CONTINUE                                                          01400960
      DO 34 K=1,NP                                                      01400970
      DO 34 J=1,NP                                                      01400980
      A(J,K) = (0.0,0.0)                                                01400990
   34 CONTINUE                                                          01401000
C-----------------------------------------------------------------------01401010
   38 CONTINUE                                                          01401020
C                                                                       01401030
C........MUST STORE 'INCWN' HERE SO IT WILL  BE PROPER FOR MATRIX-MULT. 01401040
C                                                                       01401050
      DO 2006 K=1,NMOD                                                  01401060
      DO 2006 J=1,NP1                                                   01401070
 2006 C(J,K) = INCWN(J,K)                                               01401080
      NEW1 = NP1+1                                                      01401090
      DO 2008 K=1,NMOD                                                  01401100
      DO 2008 J=NEW1,NP                                                 01401110
 2008 C(J,K) =-INCWN(J,K)                                               01401120
      CALL  CMPRD(A,C,WT,NP,NP,NMOD)                                    01401130
C                                                                       01401140
         IF(KPRINT .GT. 0 )  CALL  CMATPR(WT    ,NP,NMOD,4HSIG )        01401150
         IF( KPRINT .GT. 0 )  CALL  CMATPR( WT , NP,NMOD, 4HDSIG )
      DO 2014 K=1,NP
      DC 2014 J=1,NP
2014  A(J,K) =  -1.C * A(J,K)
      CALL  STORE(A,W12,NP,NP1, 1    ,NP1+1, NP1, NP, 3 )
      CALL  STORE(A,W21,NP,NP2,NP1+1,  1  , NP ,NP1, 4 )
C
C........COMPLETE  BY  ADDING  +1.0  TO  DIAGONAL
C
      DO 40 J=1,NP
   40 A(J,J) = A(J,J) +(1.0,0.0)
         IF(KPRINT .GT. 0 )  CALL  CMATPR(A      ,NP,NP,4HAMAT)
C
      CALL  CMINV( A, NP, C(1,1), C(NP,1) )
C
      CALL  CMPRD ( A, WT, C, NP,NP,NMOD  )
      DO 2010 K=1,NMOD
      DO 2010 J=1,NP1
2010  WT(J,K) = (  C(J,K) + INDWN(J,K) )  * 0.50
      DO 2012 K=1,NMOD
      DC 2012 J=NEW1,NP
2012  WT(J,K) = ( INDWN(J,K) -  C(J,K) )  *  0.50
         IF( KPRINT .GT. 0 )  CALL  CMATPR( WT , NP,NMOD, 4HDSIG )
C
C........NOW  RETRIEVE 'PHI'  MATRIX  FROM  TAPE
C        STORE  SAME IN '' A ''.
C          VEL-PCT = PHI * SIGMA
C
C        THEN SEPARATE THE POTENTIALS INTO PARTS
C........I.E.  BY  WING
C
      ISW  = 2
      CALL  TINCLT(   A   , NP,NP, 4HPHI , ISW, MTAPE )
      CALL  CMPRD ( A, WT, C, NP,NP,NMOD  )
         IF( KPRINT .GT. 0 )  CALL  CMATPR(C      ,NP,NMOD,4HCMAT)
C
C........WING  1
      DO 50 J=1,NMOD
      DO 50 K=1,NP1
      W12(K,J) = C(K,J)
   50 CONTINUE
C
C........WING  2
      NPP = NP1+1
      DO 52 J=1,NMOD
      KK = 0
      DO 52 K=NPP,NP
      KK = KK + 1
      W21(KK,J) = C(K,J)
   52 CONTINUE
         IF(KPRINT .GT. 0 )  CALL  CMATPR(W12    ,NP1,NMOD,4HPH-1)
         IF(KPRINT .GT. 0 )  CALL  CMATPR(W21    ,NP2,NMOD,4HPH-2)
      RETURN
      END
```

DECK 015

```
      SUBROUTINE  OUTPUT                                                01500010
     *            ( PHI      ,PM      ,XMOD     ,PRES     ,GENF    ,NP    01500020
     *            ,NMOD     ,LTAPE   ,KEY      )                         01500030
      COMPLEX                                                           01500040
     1            PHI    (     1    ) ,PM    (     1    ) ,PRES  (  NP,1  )  01500050
     2            ,GENF  ( NMOD,1 ) ,SUM                                01500060
      INTEGER     KARD   (    20    ) ,OUT                ,TYSIN        01500070
      REAL        XMOD   (50,10,2 )                                     01500080
      COMMON   /     BASIC     /  KON    (  100    )                    01500090
      OUT    = KON(58)                                                  01500100
         IF( KEY  .GT. 1 )  GO TO 3                                     01500110
      NTIT   = KON(89)                                                  01500120
         IF( NTIT .LE. 0 )  GO TO 3                                     01500130
      TYSIN = KON(60)                                                   01500140
      REWIND  TYSIN                                                     01500150
      WRITE(OUT,1000)                                                   01500160
      DO  2 J=1,NTIT                                                    01500170
      READ (TYSIN,2000)  KARD                                           01500180
      WRITE(OUT   ,2002)  KARD                                          01500190
    2 CONTINUE                                                          01500200
      REWIND  TYSIN                                                     01500210
    3 CONTINUE                                                          01500220
C                                                                       01500230
      GO TO( 4,5 ), KEY                                                 01500240
    4 WRITE(OUT,1004)                                                   01500250
      LOC  =  2                                                         01500260
      GO TO 6                                                           01500270
    5 WRITE(OUT,1006)                                                   01500280
      LOC  =  17                                                        01500290
    6 WRITE(OUT,1008)   KON(87) , KON(LOC)                              01500300
C                                                                       01500310
      ISW = 2                                                           01500320
      CALL  TINOUT(  PM   , NP,NF, 4HPMAT, ISW, LTAPE )                 01500330
      CALL  CMPRD(  PM,PHI,PRES, NP,NP,NMOD )                           01500340
      DO 12 J=1,NMOD                                                    01500350
      DO 12 I=1,NMOD                                                    01500360
      SUM    = (0.0,0.0)                                                01500370
      DO 10 K=1,NF                                                      01500380
      SUM    = SUM +  XMOD(K,I,KEY) * PRES(K,J)                         01500390
   10 CONTINUE                                                          01500400
      GENF(I,J) = SUM                                                   01500410
   12 CONTINUE                                                          01500420
C                                                                       01500430
      WRITE(OUT,1010)                                                   01500440
      CALL  CMATPR( PHI, NP, NMOD, 4HNONE )                             01500450
      WRITE(OUT,1012)                                                   01500460
      CALL  CMATPR(PRES, NP, NMOD, 4HNONE )                             01500470
      WRITE(OUT,1014)                                                   01500480
      CALL  CMATPR(GENF, NMOD,NMOD,4HNONE )                             01500490
C                                                                       01500500
         IF( KON(83) .EQ. 0 )  GOTO 99                                  01500510
C                                                                       01500520
C     PUNCHED  OUTPUT  OPTION  IS   ON.                                 01500530
C                                                                       01500540
      IPUN = KON(84)                                                    01500550
      DO 14 J=1,NFCC                                                    01500570
      WRITE(IFLN,1016) (GENF(J,I),I=1,NMOD)                             01500580
   14 CONTINUE                                                          01500590
         IF( IABS(KON(83)) .LT. 2 ) GOTO 99                             01500600
      WRITE(IFLN,1020) KEY, KON(87), KON(2)                             01500610
      DO 16 J=1,NP                                                      01500620
      WRITE(IFUN,1016)  (PRES(J,I),I=1,NMOD)                            01500630
   16 CONTINUE                                                          01500640
   99 RETURN                                                            01500650
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 01500660
 1000 OFORMAT(1H1///54X,22H* * * * * * * * * * * / 54X,1H*,19X,1H* /   01500670
     1             54X,22H*   T  I  T  L  E   * / 54X,1H*,19X,1H* /    01500680
     2             54X,22H* * * * * * * * * * * //)                     01500690
 1004  FORMAT(1H1 // 54X,22H...W I N G....O N E...)                     01500700
 1006  FORMAT(1H1 // 54X,22H...W I N G....T W O...)                     01500710
 1008  FORMAT(     // 54X,11HMACH  NBR.. , E11.4   /                    01500720
     1               54X,11HFREQUENCY.. , E11.4   )                     01500730
 1010  FORMAT(1H1 // 46X,42HV E L O C I T Y ...... P O T E N T I A L S /)01500740
 1012  FORMAT(1H1 // 46X,42HR E S U L T A N T ...... P R E S S U R E S /)01500750
 1014  FORMAT(1H1 // 46X,42HG E N E R A L I Z E D ....... F O R C E S /)01500760
 1016  FORMAT(   4E15.6  )                                              01500770
 1018 OFORMAT(26HGENERALIZED..FORCES...WING,I2,8H  MACH =,F8.4,         01500780
     1                                        8H  FREQ =,E14.6 )        01500790
 1020 OFORMAT(26HRESULTANT..PRESSURES..WING,I2,8H  MACH =,F8.2,         01500800
     1                                        8H  FREQ =,E15.6 )        01500810
 2000  FORMAT(    20A4)                                                 01500820
 2002  FORMAT(26X,20A4)                                                 01500830
       END                                                              01500840
```

DECK 024

```
      SUBROUTINE  CORED                                                 02400010
      INTEGER    OUT                                                    02400020
      DIMENSION  I(2)                                                   02400030
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 02400040
C         PURPOSE  IS  TO  FORCE  A  CORE  DUMP  WHEN  CALLED.          02400050
C            LOGIC  SHOWN  WILL  SUCCEDE  ON  IBM.360  EQUIPPED         02400060
C            WITH  STORE  PROTECTION.                                   02400070
C            OTHER  MACHINES  MAY  REQUIRE  OTHER  METHODS, AS          02400080
C            A  DIVIDE  BY  ZERO.                                       02400090
C******                                                                 02400100
CDC         IF  NO  PARAMETERS  GIVEN,   'CALL DUMP'  YIELDS            02400110
C              OCTAL  DUMP  OF  ALL  STORAGE.   (MAY NOT BE DESIRABLE)  02400120
C******                                                                 02400130
C                                                                       02400140
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 02400150
CIBM  IDUMP = 999999                                                    02400160
CIBM  I(IDUMP) = 0                                                      02400170
C                                                                       02400180
      CALL  DUMP                                                        02400190
      CALL  EXIT                                                        02400200
      END                                                               02400210
      SUBROUTINE  TINOUT                                                01700010
     *     ( X, L, M, NAME, IC, KTAPE )                                 01700020
      COMPLEX  X(L,M)                                                   01700030
      INTEGER OUT                                                       01700040
      COMMON     /BASIC /    KCNTRL(98),  IER                           01700050
      EQUIVALENCE                                                       01700060
     *                   (KCNTRL(58),OUT    )                           01700070
     *                  ,(KCNTRL(71),KPRINT)                            01700080
      DATA      IM10, IM20, IZR / -10, -20, 0 /                         01700090
C-----------------------------------------------------------------------01700100
C                                                                       01700110
C........SUBROUTINE  TO  READ  OR  WRITE  FROM  TAPE                    01700120
C        STORAGE  MODE  IS  BY  'COLUMNS'                               01700130
C     IO  = 1     WRITE  ARRAY  TO  KTAPE                               01700140
C     IO  = 2     READ   ARRAY  FROM  KTAPE                             01700150
C     IO  = -2  READ  AND  CHECK  HEADER  ONLY                          01700160
C                                                                       01700170
C-----------------------------------------------------------------------01700180
         IF( KPRINT .GT. 0 )                                            01700190
     *   WRITE(OUT,1000) NAME,L,M,KTAPE,IO                              01700200
      ISW = 1                                                           01700210
      IF(IC .LE. 0)  ISW = 2                                            01700220
      IO  = IABS(IC)                                                    01700230
        GO TO ( 1,2 ), IO                                               01700240
1     CONTINUE                                                          01700250
      WRITE(KTAPE) IM10, L, M, NAME, (IZR, J=1,10)                      01700260
      DO 10 J=1,M                                                       01700270
      WRITE(KTAPE)  (X(K,J),K=1,L)                                      01700280
10    CONTINUE                                                          01700290
      WRITE(KTAPE) IM20, L, M, NAME, (IZR, J=1,10)                      01700300
      GO TO  200                                                        01700310
C                                                                       01700320
2     READ(KTAPE) KODE, LL, MM, NNAME                                   01700330
      IF( KODE .EQ. IM10 )  GO TO 4                                     01700340
      IER  = 1                                                          01700350
      GO TO 99                                                          01700360
4     IF( NAME .EQ. NNAME )  GO TO 6                                    01700370
      IER  = 2                                                          01700380
      GO TO 99                                                          01700390
6     IF( LL .EQ. L   .AND.   MM .EQ. M )  GO TO 8                      01700400
      IER  = 3                                                          01700410
      GO TO 99                                                          01700420
8     CONTINUE                                                          01700430
        GO TO ( 9,200 ), ISW                                            01700440
9     CONTINUE                                                          01700450
      DO 20 J=1,M                                                       01700460
      READ(KTAPE)  (X(K,J),K=1,L)                                       01700470
20    CONTINUE                                                          01700480
      READ(KTAPE)  KODE                                                 01700490
      IF( KODE .EQ. IM20 ) GO TO 200                                    01700500
      IER  = 4                                                          01700510
   99 WRITE(OUT,9000) IER,KTAPE,NAME,L,M,NNAME,LL,MM                    01700520
      CALL  CORED                                                       01700530
C                                                                       01700540
200   RETURN                                                            01700550
```

```
     1           5X,14HARRAY  NAME...., A4, 10H  ORDER...., 2I6 ,            01700560
     2              10H  TAPE...., I2 ,10H  KODE....,I3  )                   01700590
9000  FORMAT(1H1 //////                                                      01700600
     1         30X,43H ***E R R O R***        SUB.TINCUT.   NBR =  ,I3/      01700610
     2         35X,I4,4X,A4,2I6 / 43X,A4,2I6/)                               01700620
      END                                                                    01700630
```

DECK 016

```
      SUBROUTINE  DISPLA                                                     01600010
     *     ( INDWN , WORK, NP1, NP2, NP, NMOD )                              01600020
      COMPLEX    INCWN( 100 , 6 ) ,WORK(NP  , 1  )                           01600030
      INTEGER    CLT                                                         01600040
      COMMCN    /   BASIC       /  KON(57), OUT                              01600050
      DO 2 J=1,NMCD                                                          01600060
      DO 2 K=1,NF                                                            01600070
    2 WORK(K,J) = INDWN(K,J)                                                 01600080
      WRITE(CLT,1000)                                                        01600090
      CALL  CMATFR( WORK, NF, NMOD,4HNONE )                                  01600100
      RETURN                                                                 01600110
1000  FCRMAT(1H1 //  30X,30HI N P U T......D O W N W A S H   / )             01600120
      END                                                                    01600130
```

DECK 018

```
      SUBROUTINE  STORE
     *      ( WT, W, NP, NN, M1, L1, M2, L2, KODE )
      COMPLEX  WT( NP,1 ) ,  W( NN,1 )
      INTEGER            OUT
      COMMON     /BASIC /    KCNTRL(98),  IERR
      EQUIVALENCE
     *                   (KCNTRL(58),OUT   )
     *                   ,(KCNTRL(71),KPRINT)
C-----------------------------------------------------------------------
C
C       M1 = START ROWS
C       L1 = START COLS
C       M2 = STOP  ROWS
C       L2 = STOP  COLS
C       NOTE  THAT  INPUT  ARRAY  IS  STORED  FROM  (1,1)
C       AND  IS  OF  ORDER M2,L2
C
C******
C*********NOTE...DO  NOT  ALLOW   'KODE'   TO APPEAR  ON  LEFT  SIDE
C         OF  AN  EQUALITY./
C******
C-----------------------------------------------------------------------
         IF( KPRINT .GT. 0 )
     *   WRITE(OUT,1000) NP,NN,M1,L1,M2,L2,KODE
      SIGN = +1.00
      IF( KODE .LT. 4) GO TO 1
      SIGN = -1.00
       GO TO 13
    1 CONTINUE
      GO TO (10,12,13), KODE
10    LL = 0
      DO 2 L=L1,L2
      LL = LL + 1
      MM = 0
      DO 2 M=M1,M2
      MM = MM + 1
    2 WT(M,L) = W(MM,LL)
        GO TO 9
12    LL = 0
      DO 4 L=L1,L2
      LL = LL + 1
      MM = 0
      DO 4 M=M1,M2
      MM = MM + 1
    4 WT(M,L) = -W(MM,LL)
        GO TO 9
13    LL = 0
      DO 6 L=L1,L2
      LL = LL + 1
      MM = 0
      DO 6 M=M1,M2
      MM = MM + 1
    6 WT(M,L) = WT(M,L) + W(MM,LL)  * SIGN
C
    9 RETURN
```

```
      END                                                               01800570
```

DECK 019

```
      SUBROUTINE  TAPES(C)                                              01900010
      COMPLEX C(1)                                                      01900020
      REAL    RECORD(14)                                                01900030
      EQUIVALENCE  (RECORD(1),KODE)                                     01900040
      COMMON   /BASIC/ KON( 57)  ,K                                     01900050
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 01900060
C        THIS  SUB-PROGRAM  WAS  DESIGNED  ONLY  FOR  'DEBUGGING'.      01900070
C        HOWEVER,  THE  USER  MAY  BE  INTERESTED  IN  THE  INDIVIDUAL  01900080
C        POTENTIALS  DERIVED  FOR  EACH  NODE-POINT.  HENCE  IT  WAS    01900090
C        INCLUDED  IN  THE  TOTAL  PACKAGE.                             01900100
C        **NOTE**   CALL  THIS  SUB-PROGRAM  ONLY  AT  END-OF-JOB        01900110
C        DUE  TO  TAPE  'REWINDS'.                                      01900120
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 01900130
      N    = 1                                                          01900140
      ISW = 1                                                           01900150
1     REWIND N                                                          01900160
      WRITE(K,1000) N                                                   01900170
10    READ(N)  RECORD                                                   01900180
      WRITE(K,1001)  RECORD                                             01900190
      IF(KODE .NE. -20)  GO TO 10                                       01900200
      GO TO( 20,20,30),N                                                01900210
20    REWIND  N                                                         01900220
      N    = 3                                                          01900230
      GO TO 1                                                           01900240
30    GO TO( 40,50 ), ISW                                               01900250
40    ISW = 2                                                           01900260
      GO TO 10                                                          01900270
50    CONTINUE                                                          01900280
      REWIND  N                                                         01900290
1001  FORMAT(//8I5,3(4X,2E12.4))                                        01900300
      N = 2                                                             01900310
      REWIND  N                                                         01900320
      NP = KON(5)                                                       01900330
      CALL TINCLT(C,NP,NP,4HPMAT,2,2)                                   01900340
      CALL CMATPR(C,NP,NP,8HPRES...1)                                   01900350
      NP = KON(20)                                                      01900360
      CALL TINCLT(C,NP,NP,4HPMAT,2,2)                                   01900370
      CALL CMATPR(C,NP,NP,8HPRES...2)                                   01900380
      REWIND  N                                                         01900390
      RETURN                                                            01900400
1000  FORMAT(1H1//10X,8HT A P E  , I4//)                                01900410
      END                                                               01900420
```

DECK 021

```
      SUBROUTINE  CMATPR( A,N,M,NAME )                                  02100010
C                                                                       02100020
C         DISPLAY  COMPLEX  ARRAYS.      ORDER = N * M                  02100030
C                                                                       02100040
      COMPLEX  A(N,M)                                                   02100050
      INTEGER  REPEAT, REMAIN                                           02100060
      INTEGER  OUT                                                      02100070
      DIMENSION   FMT(3) ,FMS(4)                                        02100080
      COMMON     /BASIC/  KCNTRL(57),OUT                                02100090
      DATA  FMT /10H(1X,4H**** ,10H,32(4H**** ,10H))         /          02100100
      DATA  FMS /10H, 8(4H**** ,10H,16(4H**** ,                         02100110
     *           10H,24(4H**** ,10H,32(4H**** /                         02100120
     CDATA      NONE, COL, ROW, REAL, IMAG / 4HNONE, 4H COL, 4H ROW,    02100130
     1          4HREAL, 4HIMAG /                                        02100140
C........IF NAME = 'NONE'  THEN IT IS ASSUMED THAT PREVIOUS             02100150
C     HEADER INFORMATION HAS BEEN SUPPLIED.  THEREFORE SKIP.            02100160
      OUT = 6                                                           02100170
        IF( NAME .EQ. NONE )  GO TO 5                                   02100180
      WRITE(OUT,100) NAME, N, M                                         02100190
    5 CONTINUE                                                          02100200
      REPEAT = M/4                                                      02100210
      REMAIN = M - 4*REPEAT                                             02100220
      IS     = 1                                                        02100230
      IE     = 4                                                        02100240
      FMT(2) = FMS(4)                                                   02100250
         IF( REPEAT .GT. 0 )  GO TO 1                                   02100260
C......HERE  FOR  LESS THAN  3-COLUMNS TO BE PRINTED.                   02100270
      IE     = REMAIN                                                   02100280
      FMT(2) = FMS(REMAIN)                                              02100290
      REMAIN = 0                                                        02100300
      REPEAT = 1                                                        02100310
    1 DO 2 J=1,REPEAT                                                   02100320
      WRITE(OUT,101)  ((COL,I),I=IS,IE)                                 02100330
      WRITE(OUT,102)  ROW,((REAL,IMAG),I=IS,IE)                         02100340
      WRITE(OUT,FMT)                                                    02100350
      DO 3 K=1,N                                                        02100360
    3 WRITE(OUT,104) K, (A(K,L),L=IS,IE)                                02100370
      IS = IE + 1                                                       02100380
      IE = IS + 3                                                       02100390
    2 CONTINUE                                                          02100400
      IF( REMAIN .EQ. 0 )  GO TO 90                                     02100410
      IE = IS + REMAIN - 1                                              02100420
      FMT(2) = FMS(REMAIN)                                              02100430
      REMAIN = 0                                                        02100440
      REPEAT = 1                                                        02100450
        GO TO 1                                                         02100460
   90 RETURN                                                            02100470
  100 FORMAT(    // 31X, 5HARRAY, 3X,A4, I6,4H   *,I4/)                 02100480
  101 FORMAT( / 16X,A4,I3,3(25X,A4,I3))                                 02100490
  102 FORMAT(       A4,6X,A4,12X,A4,6(12X,A4))                          02100500
  104 FORMAT(    I4, 8E16.6)                                            02100510
      END                                                               02100520
      SUBROUTINE   CMINV                                                02200010
     *    ( A, N, L, M )                                                02200020
      COMPLEX    A     (    1     ) ,D                  ,BIGA           02200030
     *          ,HOLD                                                   02200040
      INTEGER    OUT                                                    02200050
      DIMENSION  L      (    1     ) ,M      (    1    )                02200060
      COMMON     /     BASIC      /  KONTRL(  100    )                  02200070
      EQUIVALENCE                                                       02200080
     *            (KONTRL(58),OUT    )                                  02200090
     *           ,(KONTRL(69),KPRINT)                                   02200100
C                                                                       02200110
C     SEARCH FOR LARGEST ELEMENT                                        02200120
C                                                                       02200130
      D=(1., 0.)                                                        02200140
      NK = -N                                                           02200150
      DO  80  K =1,N                                                    02200160
      NK =NK+ N                                                         02200170
      L(K) =K                                                           02200180
      M(K) =K                                                           02200190
      KK = NK+K                                                         02200200
      BIGA= A(KK)                                                       02200210
      DO 20 J=K,N                                                       02200220
      IZ = N*(J-1)                                                      02200230
      DO 20 I=K,N                                                       02200240
      IJ =IZ+ I                                                         02200250
   10 IF ( CABS(BIGA) - CABS(A(IJ))) 15,20,20                           02200260
   15 BIGA = A(IJ)                                                      02200270
      L(K)=I                                                            02200280
      M(K)=J                                                            02200290
   20 CONTINUE                                                          02200300
         IF( KPRINT  .GT.   0 )  WRITE(OUT,1001)  BIGA                  02200310
C                                                                       02200320
C     INTERCHANGE ROWS                                                  02200330
C                                                                       02200340
      J=L(K)                                                            02200350
      IF(J-K) 35,35,25                                                  02200360
   25 KI =K-N                                                           02200370
      DO 30 I=1,N                                                       02200380
      KI =KI +N                                                         02200390
      HOLD=-A(KI)                                                       02200400
      JI = KI -K+J                                                      02200410
      A(KI) = A(JI)                                                     02200420
   30 A(JI) = HOLD                                                      02200430
C                                                                       02200440
C     INTER CHANGE  COLUMNS                                             02200450
C                                                                       02200460
   35 I=M(K)                                                            02200470
      IF(I-K) 45,45,38                                                  02200480
   38 JP =N* (I-1)                                                      02200490
      DO 40 J=1,N                                                       02200500
      JK=NK+ J                                                          02200510
      JI=JP+J                                                           02200520
      HOLD= -A(JK)                                                      02200530
      A(JK)= A(JI)                                                      02200540
   40 A(JI) = HOLD                                                      02200550
```

```
C     DIVIDE CCLUPN BY EIGA                                       02200570
C                                                                 02200580
 45   IF(CABS(BIGA)) 48,46,48                                     02200590
   46 D= (0.0,C.C)                                                02200600
      CALL  CRASH                                                 02200610
      WRITE(CLT,9C00)                                             02200620
9000  FCRMAT(1H1 /// 20X   24HDET. = 0.0   SUBR CMINV. )          02200630
      CALL  CCREC                                                 02200640
      RETURN                                                      02200650
   48 DO 55 I=1,N                                                 02200660
      IF(I-K) 50,55,50                                            02200670
   5C IK=NK +I                                                    02200680
      A(IK) =A(IK)/(-BIGA)                                        02200690
   55 CONTINUE                                                    02200700
C                                                                 02200710
C     REDUCE  MATRIX                                              02200720
C                                                                 02200730
      DO 65 I=1,N                                                 02200740
      IK= NK+I                                                    02200750
      HOLD=A(IK)                                                  02200760
      IJ= I-N                                                     02200770
      DO 65 J=1,N                                                 02200780
      IJ= IJ+N                                                    02200790
      IF(I-K) 60,65,60                                            02200800
   6C IF(J-K) 62,65,62                                            02200810
   62 KJ =IJ- I+ K                                                02200820
      A(IJ)=HCLD*A(KJ)+A(IJ)                                      02200830
   65 CONTINUE                                                    02200840
C                                                                 02200850
C     DIVIDE RCW BY PIVOT                                         02200860
C                                                                 02200870
      KJ=K-N                                                      02200880
      DO 75 J=1,N                                                 02200890
      KJ= KJ+N                                                    02200900
      IF(J-K) 7C,75,7C                                            02200910
   7C A(KJ)= A(KJ)/BIGA                                           02200920
   75 CONTINUE                                                    02200930
C                                                                 02200940
C     REPLACE  PIVCT  BY RECIPRCCAL                               02200950
C                                                                 02200960
      A(KK)=  1.C /BIGA                                           02200970
   8C CONTINUE                                                    02200980
C                                                                 02200990
C     FINAL CCLUMN ANC ROW INTERCHANGE                            02201000
C                                                                 02201010
      K=N                                                         02201020
  1CC K=(K-1)                                                     02201030
      IF(K) 150, 150, 105                                         02201040
  1C5 I=L(K)                                                      02201050
      IF(I-K) 12C,120,108                                         02201060
  108 JQ =N*(K-1)                                                 02201070
      JR= N*(I-1)                                                 02201080
      DO 110  J=1,N                                               02201090
      JK =JQ + J                                                  02201100
      HOLD =A(JK)                                                 02201110
      JI =JR+J                                                    02201120
      A(JK) = -A(JI)                                              02201130
  11C A(JI)=HCLD                                                  02201140
  120 J=M(K)                                                      02201150
 125 KI=K-N                                                              02201170
     DC 130 I=1,N                                                        02201180
     KI=KI +N                                                            02201190
     HOLD = A(KI)                                                        02201200
     JI= KI- K+J                                                         02201210
     A(KI)=-A(JI)                                                        02201220
 13C A(JI) = HCLC                                                        02201230
     GO TC 1C0                                                           02201240
 15C RETURN                                                              02201250
1001 FORMAT(// 5X, 8HPIVOT = , 2E20.8   )                                02201260
     END                                                                 02201270
```

DECK 020

```
      SUBROUTINE  CMPRC(A,B,R,N,M,L)                                     02000010
      COMPLEX     A(1),B(1),R(1)                                         02000020
C                                                                        02000030
C       CCMPLEX   MATRIX   MULTIPLY      ** SPECIAL  PURPOSE **          02000040
C          R(N,L)  =  A(N,M) * B(M,L)                                    02000050
C       ALL  MATRICES  MUST  BE  STORED  COLUMNWISE                      02000060
C       AND  AS  A  CCNTINOUS  VECTOR.   I.E.  'IBM-GENERAL-MATRIX'      02000070
C       ALL  ARRAYS  MUST  CCCUPY  DIFFERENT  STORAGE.                   02000080
C                                                                        02000090
      IR = 0                                                             02000100
      IK = -M                                                            02000110
      DO 10 K=1,L                                                        02000120
      IK = IK+M                                                          02000130
      DO 10 J=1,N                                                        02000140
      IR = IR+1                                                          02000150
      JI = J-N                                                           02000160
      IB = IK                                                            02000170
      R(IR) =(0.C,0.0)                                                   02000180
      DO 10 I=1,M                                                        02000190
      JI = JI+N                                                          02000200
      IB = IB+1                                                          02000210
   1C R(IR) = R(IR) + A(JI) * B(IB)                                      02000220
      RETURN                                                             02000230
      END                                                                02000240
```

DECK 023

```
      SUBRCUTINE  CRASH                                                  02300010
      INTEGER     CLT                                                    02300020
      COMMCN     /EASIC/  ICUMMY(57),CUT                                 02300030
      WRITE(CLT,9C0)                                                     02300040
C                                                                        02300050
  900 FORMAT( 1H1 ///           8(133(1H*)/) /////          43X,         02300060
     1    47HEEEEEEEE    RRRRRR    RRRRRR     OOOOO    RRRRRR   / 43X ,  02300070
     2    47HE           R     R   R     R   O     O  R     R  / 43X ,  02300080
     3    47HEEEEE       RRRRRR    RRRRRR    O     O  RRRRRR   / 43X ,  02300090
     4    47HE           R   R     R   R     O     O  R   R    / 43X ,  02300100
     5    47HE           R    R    R    R    O     O  R    R   / 43X ,  02300110
     6    47HEEEEEEEE    R     R   R     R    OOOOO   R     R  /// 48X , 02300120
     7    37HH      H     AAAAA    L          TTTTTTT   / 48X ,           02300130
     8    37HH      H    A     A   L             T      / 48X ,           02300140
     9    37HHHHHHHH    AAAAAAA   L             T      / 48X ,           02300150
     A    37HH      H    A     A   L             T      / 48X ,           02300160
     B    37HH      H    A     A   L             T      / 48X ,           02300170
     C    37HH      H    A     A   LLLLLLL       T      /////             02300180
     D     8(133(1H*) / )  )                                             02300190
C                                                                        02300200
      RETURN                                                             02300210
      END                                                                02300220
```

DECK 025

```
      SUBROUTINE   ENCJCB                                                02500010
      INTEGER     CUT                                                    02500020
      COMMON     /EASIC/  ICUMMY(57),CUT                                 02500030
      WRITE(OLT,1132)                                                    02500040
C                                                                        02500050
 1132 FORMAT( 1H1,  7(1H /) / 133(1H+)  ///                              02500060
     1     53X, 28HEEEEEEEE   N     N   DDDDDD    /                      02500070
     2     53X, 28HE          NN    N   D     D   /                      02500080
     3     53X, 28HEEEEE      N N   N   D     D   /                      02500090
     4     53X, 28HE          N  N  N   D     D   /                      02500100
     5     53X, 28HE          N   N N   D     D   /                      02500110
     6     53X, 28HEEEEEEEE   N     N   DDDDDD    ///                    02500120
     7     58X, 15H CCCCC     FFFFFFF  /                                 02500130
     8     58X, 12HC      O   F        /                                 02500140
     9     58X, 15HC      O   FFFF     /                                 02500150
     A     58X, 12HC      O   F        /                                 02500160
     B     58X, 12HC      C   F        /                                 02500170
     C     58X, 12H CCCCC     F        ///                               02500180
     D     53X, 28H       J    OCCOO    BBBBBB   /                       02500190
     F     53X, 28H       J   O     O   B     B  /                       02500200
     F     53X, 28H       J   O     O   BBBBBB   /                       02500210
     G     53X, 28H       J   O     O   B     B  /                       02500220
     H     53X, 28HJ      J   O     O   B     B  /                       02500230
     I     53X, 28H JJJJJ      OCCOO    BBBBBB   ////, 133( 1H+ )  )     02500240
      RETURN                                                             02500250
      END                                                                02500260
```

APPENDIX A.

As a debugging aid, intended for the programmer only, the input package will accept an LABEL data section not previously described in the users manual. This section is called "PRINT" and requires 2 (two) cards. Card 2 tells the program which areas are to provide printout. Any or all of card 2 may be blank or zero. (A blank/zero inplies print off.)

Card 1: Format:Type 2

1	2	3	4	5
P	R	I	N	T

Card 2: Format:Type 3

MAIN	1
INPUT,DATA, DIAPH3	1
MASTER, ORIENT	1
TRINTX	2
PSI	1
DWASHY	2
ASSPOT, ASSY	1
ASSDWN	1
CMINV	1
TAPES	3
STORE, TINOUT	1
SIZE	1

This section is optional. It may appear anywhere before the (first) end card. (A logical position is between TITLE and SYSTEM)

** NOTE **

Judicious use of this option is suggested, as horrendous amounts of print can be generated. The potential of this option to the programmer can be shown by exercising the full print on the very small test case supplied with the users manual (Ref. 2.).

APPENDIX B.

PROBLEM SIZE ALTERATIONS

This program utilizes both fixed and variable length arrays. If a change in program size (data) is required, only the fixed length arrays need be changed. Table B.1 will be the basis for further discussion.

If data storage must be altered, the user must first determine his upper limits and then alter the arrays according to the following definitions.

1. XYZ (i, j, k) (grid point coordinates)

   i = coordinate value (x, y, or z)

   j = grid point number (1 -- N)

   k = planform (1 or 2)

2. IBLN (i, j, k) (element boolean)

   i = grid point number

   j = element number

   k = planform (1 or 2)

3. XMOD (i, j, k) (mode shapes)

   i = mode shape for grid point

   j = mode shape set

   k = planform (1 or 2)

4. LND (i, j) (leading edge grid point #)

i = grid point number (increasing order)

j = planform (1 or 2)

5. MLINE (i, j) (mach line grid point #)

i = grid point number (increasing order)

j = planform (1 or 2)

6. TITLE (i, j) (work storage)

i = 4-literal characters

j = card number

7. INDWN (i, j) (input downwash - calculated)

i = related grid point value (planform 1 and 2)

j = related mode shape set

Note: planform 1 data is stored in "i" col, position 1 through NP1; planform 2 data is stored in "i" col. (NP1 + 1) through NP = (NP1 + NP2).

Once the new problem size is determined, program alteration takes place in the following manner:

Step 1. Subroutine SIZE - alter data stament (card no. 00400050) as required.

Step 2. Main program - change cards as required.
dimension of WORC (card no. 00100010),
dimension of WORK (card no. 00100040),
set new maximums on cards no. 00100500 through 00100580

Step 3. Check Table B.1 for array/subroutine and alter dimension statements as required.

Step 4. Exercise new problem data (no run card) with PRINT option on for sub SIZE. Compare with hand calculations.

Array	Type	Delivery Size	Subroutine					
			Input	Data	DIAPH3	Master	Output	DISPLA
XYZ	R	(3,80,2)	Y	Y	Y	Y	-	-
IBLN	I	(3,100,2)	Y	Y	Y	Y	-	-
XMOD	R	(50,10,2)	Y	Y	-	Y	Y	-
LND	I	(20,2)	Y	Y	Y	Y	-	-
MLINE	I	(20,2)	Y	-	Y	Y	-	-
TITLE	R	(20,20)	Y	Y	-	-	-	-
INDWN	C	(100,10)	-	-	-	Y	-	Y

Table B.1

Legend:

Y	=	array present in subroutine
R	=	Real
I	=	Integer
C	=	Complex

APPENDIX C

SAMPLE PROBLEM/DATA/RESULTS

The purpose of this section is to demonstrate graphically how the user describes a planform, the actual data required, and the results generated by the computer. Figure C.1 shows a simple DELTA-WING planform consisting of 4-gridpoints and 2-elements. The broken line elements and gridpoints are generated interally, but the user should always have an idea of what this "DIAPHRAGM" region looks like before data preparation begins. The angle θ ($\sphericalangle$ 2,1,6) is computed thru SINE θ = 1 / (MACH NBR). Gridpoints are indicated by numbers. Elements are the circled numbers. Note that the coordinate system is a normal right-handed system if this page is turned 90-degrees. The CHORD of this planform lies on the X-axis, with gridpoint 1 at the origin.

Should questions arise during the study of this example, re-read the LABEL section covering the question.

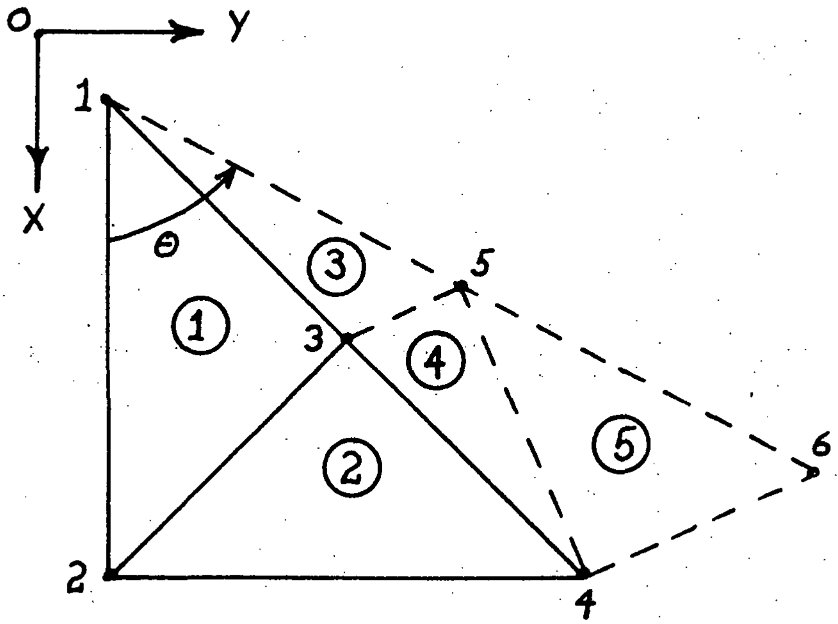


Figure C.1 Sample Planform Showing Basic Structure and Program Generated Elements

A I C / I N T......D A T A   D E C K

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
RUN                                                                     TEST0001
TITLE    9                                                              TEST0002
                                                                        TEST0003
             .. A I C   /    I N T ..  .. T E S T     C A S E ..       TEST0004
         THIS  EXAMPLE  IS  USED  TC  DEMONSTRATE  THE  INPUT  DATA     TEST0005
         REQUIRED  TO  EXECUTE  THE   'AIC / INT'   PROGRAM.            TEST0006
         BOTH  WINGS  ARE  IDENTICAL,  AND  THE  UPPER  WING  LIES      TEST0007
         DIRECTLY  ABOVE  THE  LOWER  WING (I.E.  STAGGER  DISTANCE    TEST0008
         IS  ZERO).   BOTH  WINGS  ARE  SUB-SONIC.   EXECUTION  TIME    TEST0009
         IS  TRIVIAL.   FULL  LOGIC  IS  EMPLOYED  DURING  ANALYSIS.    TEST0010
                                                                        TEST0011
SYSTEM                                                                  TEST0012
         4    2    2    3    1    1                                     TEST0013
              1.15                           2.0            .005        TEST0014
LEDGE                                                                   TEST0015
         1    3    4                                                    TEST0016
ELEM                                                                    TEST0017
         1    1    2    3                                               TEST0018
         2    2    4    3                                               TEST0019
COORD                                                                   TEST0020
         1    0.0            0.0                                        TEST0021
         2    2.0            0.0                                        TEST0022
         3    1.0            1.0                                        TEST0023
         4    2.0            2.0                                        TEST0024
MODE                                                                    TEST0025
         1    1.0           -1.0                                        TEST0026
         2    1.0            1.0                                        TEST0027
         3    1.0            0.0                                        TEST0028
         4    1.0            1.0                                        TEST0029
END                                                                     TEST0030
TITLE    5                                                              TEST0031
                             * * * N O T E * * *                        TEST0032
         THESE  CARDS  ARE  BEING  PRINTED  FROM   'WING - 2'           TEST0033
         DATA.    THEREFORE  USER  MAY  KEEP  DATA  LABELED             TEST0034
         WITH  A  TITLE  SECTION  AND  IT  WILL  NOT  INTEFER           TEST0035
         WITH  PROGRAM  INPUT.                                          TEST0036
SYSTEM                                                                  TEST0037
         4    2    2    3    1    1                                     TEST0038
              1.15                           2.0           0.00         TEST0039
LEDGE                                                                   TEST0040
         1    3    4                                                    TEST0041
ELEM                                                                    TEST0042
         1    1    2    3                                               TEST0043
         2    2    3    4                                               TEST0044
COORD                                                                   TEST0045
         1    0.0            0.0            0.5                         TEST0046
         2    2.0            0.0            0.5                         TEST0047
         3    1.0            1.0            0.5                         TEST0048
         4    2.0            2.0            0.5                         TEST0049
MODE                                                                    TEST0050
         1    1.0           -1.0                                        TEST0051
         2    1.0            1.0                                        TEST0052
         3    1.0            0.0                                        TEST0053
         4    1.0            1.0                                        TEST0054
END                                                                     TEST0055
ENDATA                                                                  TEST0056
12345678901234567890123456789012345678901234567890123456789012345678901234567890
```

```
...W I N G..1.....C O N T R L S...

NBR  GRID  POINTS  ON  WING...    4
NBR  ELEMENTS ON  PLANFORM....    2
NBR  DEGREES  OF  FREEDOM.....    2
NBR  LEADING  EDGE  POINTS....    3
NBR  DIVISIONS  IN  DIAPHRAGM.    1
SYMMETRY   FACTOR.............    1
REFERENCE  LENGTH..... 0.20000E 01


..G E N E R A L....C O N T R L S..


M A C H  NUMBER....... 0.11500E 01
NBR  MODE  SHAPES.............    2
TRUNCATION  (EPS)..... 0.50000E-02


LEADING  EDGE  GRID  POINTS.
1    3    4
```

```
                    ELEMENT  GRID  POINTS

                   ELEM        A        B        C

                     1         1        2        3
                     2         2        4        3



                    GRID  POINT  COORDINATES

GRID  POINT       X                   Y                   Z
        1       0.0                 0.0                 0.0
        2       0.100000E 01        0.0                 0.0
        3       0.500000E 00        0.500000E 00        0.0
        4       0.100000E 01        0.100000E 01        0.0



                      MODE  SHAPE  DATA

                 1   0.100000E 01  -0.100000E 01
                 2   0.100000E 01   0.100000E 01
                 3   0.100000E 01   0.0
                 4   0.100000E 01   0.100000E 01
```

```
...W I N G..2.....C O N T R L S...

NBR  GRID  POINTS  ON  WING...     4
NBR  ELEMENTS ON  PLANFORM....     2
NBR  DEGREES  OF  FREEDOM.....     2
NBR  LEADING  EDGE  POINTS....     3
NBR  DIVISIONS  IN  DIAPHRAGM.     1
SYMMETRY  FACTOR.............     1
REFERENCE  LENGTH..... 0.20000E 01


..G E N E R A L....C O N T R L S..


EXTRAPOLATION REQUIRED(1=YES).    0
STAGGER       (LE.TO.LE) 0.0


LEADING -EDGE  GRID  POINTS.
1    3    4
```

```
                  ELEMENT  GRID  POINTS

                ELEM       A       B       C

                  1        1       2       3
                  2        2       3       4



                  GRID  POINT  COORDINATES

GRID  POINT       X                    Y                    Z
        1       0.0                  0.0                  0.250000E 00
        2       0.100000E 01         0.0                  0.250000E 00
        3       0.500000E 00         0.500000E 00         0.250000E 00
        4       0.100000E 01         0.100000E 01         0.250000E 00



                       MODE  SHAPE  DATA

                1   0.100000E 01  -0.100000E 01
                2   0.100000E 01   0.100000E 01
                3   0.100000E 01   0.0
                4   0.100000E 01   0.100000E 01

                ***** END  CARD  ENCOUNTERED *****
```

```
                    'W I N G....O N E


                  DIAPHRAGM  ELEMENTS  WERE
                  ADDED  AS  FOLLOWS.



                  GRID  POINT  COORDINATES
GRID  POINT           X                   Y                    Z
       5          0.391973E 00        0.690226E 00          0.0
       6          0.783945E 00        0.138045E 01          0.0


                     ELEMENT  GRID  POINTS
                  ELEM        A         B         C

                     3        3         5         1
                     4        4         5         3
                     5        4         6         5

                   2  MACH LINE GRID POINTS.
  5    6
```

```
                   W I N G....T W O


                 DIAPHRAGM  ELEMENTS  WERE
                 ADDED  AS  FOLLOWS.



                 GRID  POINT  COORDINATES
             GRID  POINT       X                    Y                    Z
                   5         0.391973E 00         0.690226E 00         0.250000E 00
                   6         0.783945E 00         0.138045E 01         0.250000E 00


                    ELEMENT  GRID  POINTS
                 ELEM        A        B        C

                    3        3        5        1
                    4        4        5        3
                    5        4        6        5

                  2  MACH LINE GRID POINTS.
             5     6
```

```
                              I N P U T......D O W N W A S H

                    COL  1                                 COL  2
 ROW       REAL                  IMAG               REAL              IMAG
*********************************************************************************
   1     0.0               0.400000E 00       0.200000E 01    -0.400000E 00
   2     0.0               0.400000E 00       0.200000E 01     0.400000E 00
   3     0.0               0.400000E 00       0.200000E 01     0.0
   4     0.0               0.400000E 00       0.200000E 01     0.400000E 00
   5     0.0               0.400000E 00       0.200000E 01    -0.400000E 00
   6     0.0               0.400000E 00       0.200000E 01     0.400000E 00
   7     0.0               0.400000E 00       0.200000E 01     0.0
   8     0.0               0.400000E 00       0.200000E 01     0.400000E 00
```

```
                    * * * * * * * * * * *
                    *                   *
                    *   T  I  T  L  E   *
                    *                   *
                    * * * * * * * * * * *

                                                            TEST0003
   .. A I C   /   I N T ..    .. T E S T   C A S E ..       TEST0004
THIS  EXAMPLE  IS  USED  TO  DEMONSTRATE  THE  INPUT  DATA  TEST0005
REQUIRED  TO  EXECUTE  THE   'AIC / INT'   PROGRAM.         TEST0006
BOTH  WINGS  ARE  IDENTICAL,  AND  THE  UPPER  WING  LIES   TEST0007
DIRECTLY  ABOVE  THE  LOWER  WING (I.E.  STAGGER  DISTANCE  TEST0008
IS  ZERO).   BOTH  WINGS  ARE  SUB-SONIC.   EXECUTION  TIME TEST0009
IS  TRIVIAL.   FULL  LOGIC  IS  EMPLOYED  DURING  ANALYSIS. TEST0010
                                                            TEST0011
                    * * * N O T E * * *                     TEST0032
THESE  CARDS  ARE  BEING  PRINTED  FROM   'WING - 2'        TEST0033
DATA.     THEREFORE  USER  MAY  KEEP  DATA  LABELED         TEST0034
WITH  A  TITLE  SECTION  AND  IT  WILL  NOT  INTEFER        TEST0035
WITH  PROGRAM  INPUT.                                       TEST0036
```

```
                       ...W I N G....) N E...

                       MACH  NBR.. 0.1150E 01
                       FREQUENCY.. 0.4000E 00



                 V E L O C I T Y ...... P O T E N T I A L S


                   COL  1                           COL  2
ROW        REAL             IMAG             REAL             IMAG
*****************************************************************************
  1    0.0              0.0              0.0              0.0
  2    0.271526E 00     0.506305E 00     0.268412E 01    -0.108701E 01
  3    0.393869E-01     0.101497E 00     0.491935E 00    -0.234013E 00
  4    0.184515E 00     0.249123E 00     0.126373E 01    -0.838053E 00



                 R E S U L T A N T ...... P R E S S U R E S


                   COL  1                           COL  2
ROW        REAL             IMAG             REAL             IMAG
*****************************************************************************
  1    0.175622E-01     0.447830E-01     0.234685E 00    -0.641168E-01
  2    0.334217E-01     0.994732E-01     0.517793E 00    -0.103803E 00
  3    0.401685E-01     0.956043E-01     0.503576E 00    -0.140339E 00
  4    0.222219E-01     0.517025E-01     0.271975E 00    -0.738901E-01



                 G E N E R A L I Z E D ....... F O R C E S


                   COL  1                           COL  2
ROW        REAL             IMAG             REAL             IMAG
*****************************************************************************
  1    0.113374E 00     0.291563E 00     0.152803E 01    -0.382149E 00
  2    0.380814E-01     0.106393E 00     0.555082E 00    -0.113576E 00
```

```
                    ....W I N G....T W O...

                    MACH  NBR.. 0.1150E 01
                    FREQUENCY.. 0.4000E 00



           V E L O C I T Y ...... P O T E N T I A L S

              COL  1                         COL  2
ROW      REAL            IMAG           REAL            IMAG
***********************************************************************
  1    0.0             0.0            0.0             0.0
  2    0.271527E 00    0.506306E 00   0.268412E 01   -0.108701E 01
  3    0.393868E-01    0.101497E 00   0.491935E 00   -0.234013E 00
  4    0.184515E 00    0.249124E 00   0.126373E 01   -0.838053E 00



           R E S U L T A N T ...... P R E S S U R E S

              COL  1                         COL  2
ROW      REAL            IMAG           REAL            IMAG
***********************************************************************
  1    0.175622E-01    0.447831E-01   0.234685E 00   -0.641168E-01
  2    0.334217E-01    0.994732E-01   0.517793E 00   -0.103803E 00
  3    0.401685E-01    0.956043E-01   0.503577E 00   -0.140339E 00
  4    0.222219E-01    0.517025E-01   0.271975E 00   -0.738900E-01



           G E N E R A L I Z E D ........ F O R C E S

              COL  1                         COL  2
ROW      REAL            IMAG           REAL            IMAG
***********************************************************************
  1    0.113374E 00    0.291563E 00   0.152803E 01   -0.382149E 00
  2    0.380814E-01    0.106393E 00   0.555083E 00   -0.113576E 00
```

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

EEEEEEE   N     N   DDDDDD
E         NN    N   D     D
EEEE      N N   N   D     D
E         N  N  N   D     D
E         N   N N   D     D
EEEEEEE   N     N   DDDDDD

      OOOOO   FFFFFFF
     O     O  F
     O     O  FFFF
     O     O  F
     O     O  F
      OOOOO   F

      J   OOOOO   BBBBBB
      J  O     O  B     B
      J  O     O  BBBBBB
      J  O     O  B     B
J     J  O     O  B     B
 JJJJJ    OOOOO   BBBBBB

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

REFERENCES

1. Appa, K. and Smith, G. C. C., "Development and Applications of Supersonic Unsteady Consistent Aerodynamics for Interfering Parallel Wings", NASA CR-2168

2. Paine, A. A., "Development and Applications of Supersonic Unsteady Consistent Aerodynamics for Interfering Parallel Wings - User's Manual", NASA CR-112184