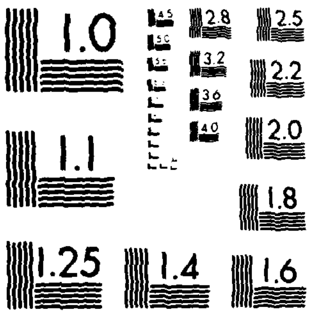
AD-A144 369   UPDATE TO THE FREE FLOATING TWO-DIMENSIONAL EXTENSIBLE   1/1
CABLE SYSTEM MODEL (FF2E)(U) NAVAL AVIONICS CENTER
INDIANAPOLIS IN K L HOUSER 18 MAY 84 NAC-TR-2359

UNCLASSIFIED   F/G 20/4   NL

END
DATE
FILMED
9-84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

TR-2359

12

AD-A144 369

UPDATE TO THE FREE FLOATING TWO-DIMENSIONAL EXTENSIBLE CABLE SYSTEM MODEL (FF2E)

KEVIN L. HOUSER
ANTISUBMARINE WARFARE
ENGINEERING BRANCH
NAVAL AVIONICS CENTER
INDIANAPOLIS, IN 46218

18 MAY 1984

Approved for public release; distribution unlimited.

84 08 07 052

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NAC-TR-2359	2. GOVT ACCESSION NO. AD-A144	3. RECIPIENT'S CATALOG NUMBER 369
4. TITLE (and Subtitle) Update to the Free Floating Two-Dimensional Extensible Cable System Model (FF2E)		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Kevin L. Houser		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Avionics Center Anti-Submarine Warfare Engineering Branch 955 Indianapolis, IN 46218		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 18 May 1984
		13. NUMBER OF PAGES 20
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

1. Free Floating Cable System, 2. Sonobuoy, 3. Ocean Current Profile

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Free Floating Two-Dimensional Extensible Cable System Model (FF2E) was modified to account for velocity dependent drag forces on surface floats, non-linear stretch characteristics of cable, and true current versus depth calculation relative to the 90% ocean current profile. Two new subroutines incorporating interpolation schemes for empirical data have been added to the program. A third subroutine has been modified to allow direct calculation of current versus depth when the 90% current profile is used.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UPDATE TO THE FREE FLOATING TWO-DIMENSIONAL EXTENSIBLE CABLE SYSTEM MODEL (FF2E)

NAVAL AVIONICS CENTER
6000 E. 21ST STREET
INDIANAPOLIS, IN 46218

PREPARED BY: Kevin Houser
KEVIN L. HOUSER

APPROVED BY:
MICHAEL L. HADEN

TABLE OF CONTENTS

UPDATE TO THE FREE FLOATING TWO-DIMENSIONAL EXTENSIBLE CABLE SYSTEM MODEL (FF2E)

PURPOSE

The purpose of this report is to explain the modifications made to the FF2E computer program to more accurately predict operational behavior of a drifting sonobuoy. Each modification will be discussed in detail including the new data inputs required to execute the program.

BACKGROUND

The FF2E computer program, references (1) and (2), was developed to predict the steady-state behavior of a cable system in an ocean environment. The program, as referenced in sonobuoy specifications, is used to predict the operating conditions of a sonobuoy. The two conditions generally emphasized are the operating depth of the hydrophone and the velocity of the hydrophone, relative to the current at the hydrophone's operating depth. For the purpose of this report, the relative hydrophone velocity will be referred to as the "flow velocity". The flow velocity for a sonobuoy design, in a given two-dimensional current profile, can be predicted as a function of the drag characteristics of the sonobuoy system. The predicted flow velocity is used to establish the test scenario for evaluating mechanically-induced noise. The level of mechanically-induced noise is a function of the flow velocity; therefore, an accurate flow prediction is required. With increasing emphasis being placed on reducing mechanically-induced noise, a more accurate model of a drifting sonobuoy is required.

OBJECTIVE

The objective of the task was to develop a more accurate means of modeling the behavior of a drifting sonobuoy. The task was divided into four major areas:

a. Develop a subroutine which would find the true drag force on a surface unit given the flow velocity past the surface unit.

b. Incorporate a change to the program which will allow the use of the analytical formula for the 90% ocean current profile of reference (3) to calculate flow velocity at any given depth.

c. Add a subroutine which would compensate for the non-linear stretch characteristics of a cable or compliance in the sonobuoy suspension system.

d. Make a modification to the subroutine "BODY" which corrects an inherent interpolation error.

SUMMARY OF CHANGES

a. An interpolation scheme has been included in the subroutine "VAR" which allows the true drag force on a surface unit to be found. This is accomplished by interpolation of empirically measured values relating drag forces relative to flow velocities on the surface unit. If no table of values is to be input, the program will operate as previously written.

b. A modification to the subroutine "CUR" has been incorporated which will allow direct calculation of flow velocity given any depth in the 90% ocean current profile. When this profile is used, input of current velocity data points is not required. If the 90% profile is not used, the program will operate as previously written, by interpolation between current velocity data points.

c. The subroutine "STRETH", identified by reference (4), has been added to the program to account for the non-linear stretching characteristics of cables and compliances. This subroutine utilizes an input table of tension values versus AE values, where "A" is equal to the cross-sectional area and "E" is equal to Young's Modulus. If a cable has a constant value of AE the program will operate as previously written. The input for AE has been altered to allow for the input of tables.

d. A modification has been made to the subroutine "BODY" which corrects an error in output when a velocity or angle value falls within the first segment of the lift drag table.

DISCUSSION

Because of Fleet operational requirements, increased emphasis has been placed on reducing mechanically induced noise in sonobuoys. Therefore, emphasis is being placed on upgrading and refining the analytical tools used to evaluate the suspension systems.

Due to this fact, modifications to the FF2E program have been incorporated to create a more accurate prediction of a drifting sonobuoy. In the following paragraphs each modification will be discussed. Included in this report is an improved data input scenario which contains the added input parameters for the modifications. In Appendix I there is a complete listing of the improved FF2E program.

SUBROUTINE "VAR"

In the present version of the FF2E program, drag of the surface unit is determined using the flow velocity past the unit at a depth of one-half the draft, where the draft is equal to the distance from sea surface to the bottom of the surface unit. As the program iterates, it searches for an equilibrium of the buoyancy of the upper unit and the effective weight of the hydrophone and suspension system by varying the drift velocity and sonobuoy draft. Because the FF2E program uses a mathematical equation to model the drag force of the upper unit, the calculated drag force may not exactly model the empirically derived drag force of the actual hardware. This inherent error creates a false equilibrium of forces and ultimately could effect the flow velocity of the hydrophone.

The subroutine "VAR" was added to allow the true drag of the surface unit to be found given the relative flow of the upper unit. This subroutine utilizes a reference table of values which contain measured drag forces versus flow velocities. In each iteration of the subroutine "STEADY" the subroutine "VAR" is called with values of drag, flow, draft, diameter, and drag coefficient. Using the value of flow equal to the flow velocity at one-half the draft, the subroutine finds the drag force by interpolating between empirically derived data points. This value of drag force is then compared with the calculated drag force. If the difference between these two values is less than .001 lbs., the program will return and continue with the iterations of the subroutine "STEADY". For differences greater than .001 lbs., the subroutine calculates the surface unit drag coefficient based on the empirically measured drag force value. This coefficient is then used to vary the original surface unit drag coefficient until correlation between the empirically measured drag force and the calculated drag force is achieved.

DATA INPUT CHANGES FOR SUBROUTINE "VAR"

Three variables have been added to the data input scheme. The first is NFOSB. NFOSB is the number of drag forces to be entered in the drag versus flow velocity table. A maximum of 15 values may be entered in the table. The second variable is FOSB(k), where k=1 to NFOSB. FOSB(k) corresponds to the drag force values to be input in the table. FOSB(k) should range from 0 lbs. to a magnitude which is greater than the expected drag on the surface unit. The final variable is VOSB(k), which is the flow velocity in knots that corresponds to the force values entered in FOSB(k). If no table is to be entered for the surface unit drag, then NFOSB should be set equal to zero, and no values for FOSB(k) and VOSB(k) are entered.

DATA OUTPUT FOR SUBROUTINE "VAR"

The program will print out the corresponding value of FOSB and VOSB in a table format. After the program has reached the steady state solution, it will print out the drag force on the surface unit as, "the computed horizontal component of tension". Also, the program prints out, "the CD of the surface unit", and "the flow past the surface unit".

SUBROUTINE CUR

In the present FF2E program, subroutine "CUR" is called to perform a linear interpolation of the input current profile points. The purpose of this subroutine is to find a flow velocity for a given depth in the ocean profile. Depending on the step-size and number of current profile points entered, the amount of error in the interpolation of depths close to the surface will vary, since the current profile is not a straight line at shallow depths.

In the modified program, the analytical formula which describes the 90% profile, has been added to subroutine "CUR". If the user wishes to use the 90% profile, then the number of current profile points, NCUR, must be set equal to zero and XX(I) and YY(I) values are not entered. If NCUR is not zero, then the values for the current profile must be entered.

SUBROUTINE "STRETH"

The present FF2E computer program assumes that the cable and compliance of a sonobuoy suspension system will have linearly stretching characteristics. In the past, the problem of a non-linear stretching cable has been over-ridden by inputting the stretched length of the cable into FLC(k) and choosing an AE value of high magnitude.

In the modified program, subroutine "STRETH", identified by refrence (4), has been added to account for cables with non-linear stretch characteristics. The subroutine utilizes a reference table of values which relates cable tension to a value of AE. The AE value is calculated from the following equation:

$$AE = \frac{Lo\ (T-Tref)}{L-Lo}$$

Where,
A = cross sectional area (ft^2)
E = Young's Modulus (lb/ft^2)
L = stretched length of cable (ft)
Lo = unstretched length of cable (ft)
T = tension applied to cable (lb)
Tref = reference tension of unstretched cable (lb)

DATA INPUT CHANGES FOR "STRETH"

Four variables have been added to the input data parameters. The first is NAE which corresponds to the number of points to be entered for the relation of tension to AE for each cable and compliance. NAE has a range from 1 to 15. FAE(I,K) and AE(I,K) are the corresponding tension and AE values for a cable over which the tension-strain curve is non-linear, where I=1 to NCAB and K=1 to NAE. For cables with a linear tension-strain curve, NAE will be set equal to 1, and the fourth variable, AE(I,1) would be equal to a constant AE value. Values for AE should be entered for the entire range of tensions over which the tension-strain curve is non-linear. For tension values outside the range of the table, AE will be set equal to the endpoint value.

DATA OUTPUT CHANGES FOR "STRETH"

The program will print out the corresponding Tension-AE values in tabular format as they are entered in the input. For cables with linear stretch characteristics the constant value of AE will be printed out.

SUBROUTINE "BODY"

A modification to the subroutine "BODY", reference (4), has been provided which corrects an error in this subroutine. Previously, when a velocity or angle value fell within the first segment of the lift drag table, the first value of the table was chosen rather than interpolating between the first and second values of the table. As an example, assume the following table is in the program:

ANGLE

			0		10		20	
V		-	L	D	L	D	L	D
E	1	-	4	1	7	4	10	7
L		-						
O	2	-	5	2	8	5	11	8
C		-						
I	3	-	6	3	9	6	12	9
T		-						
Y		-						

The new data points may be computed as follows:

STEP 1: Determine lift at the desired velocity and at a given angle by interpolation

$$\frac{(V_3 - V_1)\ (V_2 - V_1)}{L_3 - L_1}\ (V_2 - V_1) + L_1 = L2$$

where V_2 = desired velocity
V_1 = next lowest velocity data point
V_3 = next highest velocity data point
L_2 = calculated lift for desired velocity
L_1 = lift value for next lowest velocity
L_3 = lift value for next highest velocity

Example: $\frac{(2-1)\ (1.5 - 1)}{(5 - 4)} + 4 = L_2 = 4.5$ for angle = 0°, and velocity = 1.5

$\frac{(2 - 1)\ (1.5 - 1)}{(8 - 7)} + 7 = L'_2 = 7.5$ for angle = 10°, and velocity = 1.5

STEP 2: Determine lift for the desired angle at the desired velocity by interpolation:

$$\frac{(L'_2 - L_2)\ (A_2 - A_1)}{A_3 - A_1} + L_2 = L$$

where A_1 = next lowest angle in table
A_3 = next highest angle in table
L = value for lift at the desired velocity and angle

Example: $\frac{(7.5 - 4.5)\ (10 - 0)}{5-0} + 4.5 = 6$ for angle = 5° and velocity = 1.5

STEP 3: Compute drag by substituting "D" for "L" and repeating steps 1 and 2.

The new data points for the example are found to be lift = 6 and drag = 3 for velocity = 1.5 and angle = 5°.

INPUT DATA FORMAT AND DESCRIPTION

A description of the input parameters of the modified FF2E program is shown below:

CARD 1

COLUMN 1-3

1. NCASES - number of sets of data to be ran. NCASES is set equal to 1 for one set of data to be ran. NCASES is an integer.

CARD 2

COLUMN 1-3

2. NCUR - number of current profile points to be entered. The maximum number of points is 30. If NCUR=0, the 90% profile is assumed. NCUR is an integer.

COLUMN 4-6

3. NCAB - number of cable segments bounded by endpoints or system components. NCAB is an integer.

COLUMN 7-9

4. NHPHS - number of hydrophones. NHPHS is an integer.

COLUMN 10-12

5. NTAB - number of lift/drag tables. If no tables are used then NTAB=0. NTAB is an integer.

CARD 3

COLUMN 1-12

6. DAI - diameter of buoy antenna in inches. DAI is a floating point number.

COLUMN 13-24

7. LA - length of buoy antenna in feet. LA is a floating point number.

COLUMN 25-36

8. CDA - drag coefficient in air of buoy antenna. CDA is a floating point number.

COLUMN 37-48

9. TBH - *horizontal component of tension in pounds* applied to lower unit. TBH is a floating point number.

10. TBV - vertical component of tension is pounds applied to lower unit. TBV is a floating point number. TBH, TBV should equal 0. for free floating system.

CARD 4

COLUMN 1-12

11. DB - diameter of cylindrical buoy in feet. DB is a floating point number.

COLUMN 13-24

12. LB - length of cylindrical buoy in feet. LB is a floating point number.

COLUMN 25-36

13. CDB1 - drag coefficient of submerged portion of surface buoy. CDB1 is a floating point number.

COLUMN 37-48

14. CDB2 - drag coefficient of surface buoy above sea surface. CDB2 is a floating point number.

COLUMN 49-60

15. WB - weight of surface buoy in air in pounds. WB is a floating point number.

COLUMN 61-72

16. UWINDK - velocity of wind in knots. UWINDK is a floating point number.

CARD 5

COLUMN 1-3

17. NFOSB - number of values on the force vs. flow curve. NFOSB = 0 to 15 and is an integer.

CARD 6

COLUMN 1-12, 13-24, etc.

18. FOSB(K) - force values on the force/flow curve for the surface unit in pounds. FOSB(k) are floating point numbers, where K=1, NFOSB.

CARD 7

COLUMN 1-12, 13-24, etc.

19. VOSB(K) - flow values on the force/flow curve for the surface unit in knots. VOSB(K) are floating point numbers, where K=1, NFOSB.

Note: For variables 18 and 19, a maximum of 6 values may be read in on one card. As many additional cards as required may be used.

CARD 8

COLUMN 1-12

20. CDAPK - drag coefficient times the normal drag area of package hung just below the surface buoy in feet squared. CDAPK is a floating point number.

COLUMN 13-24

21. WPAK - wet weight of package hung just below the surface buoy in pounds. WPAK is a floating point number.

CARD 9

COLUMN 1-12, 13-24, etc.

22. FLC(K) - length of each cable in feet. FLC(K) is a floating point number, where K=1, NCAB.

CARD 10

COLUMN 1-12, 13-24, etc.

23. NPR(K) - number of segments each cable is divided. NPR(K) are integer values, where K=1, NCAB.

CARD 11

COLUMN 1-12, 13-24, etc.

24. DCI(K) - diameter of each cable in inches. DCI(K) are floating point numbers, where K=1, NCAB.

CARD 12

COLUMN 1-12, 13-24, etc.

25. CVFAC(K) - conversion factor for non-circular cross section of cable. CVFAC is a floating point number, where K=1, NCAB.

CARD 13

COLUMN 1-12, 13-24, etc.

26. WC(K) - weight per unit length of each cable in water in pounds per foot. WC(K) are floating point numbers where K=1, NCAB.

CARD 14

COLUMN 1-12, 13-24, etc.

27. CDC(K) - coefficient of drag on each cable normal to the cable. CDC(K) are floating point numbers, where K=1, NCAB.

CARD 15

COLUMN 1-12, 13-24, etc.

28. TREF(K) - reference tension in each cable in pounds. If FLC(K) is an unstretched length, set TREF(K) = 0. TREF(K) are floating point numbers.

CARD 16

COLUMN 1-12, 13-24, etc.

29. (K) - Poisson's ratio for each cable. P(K) are floating point numbers.

CARD 17

COLUMN 1-3

30. NAE(K) - number of points on the tension/AE curve for each cable. NAE(K) are integers. For a cable without a table NAE=1.

CARD 18

COLUMN 1-12, 13-24, etc.

31. AE(K,1) - AE value for cable with linear stretch characteristics with NAE=1. AE(K,1) is a floating point number.

CARD 18a

COLUMN 1-12, 13-24, etc.

32. FAE(K,I) - tension values over which AE is changing in pounds. FAE(K,I) are floating point numbers, where K=1, NCAB and I=1, NAE.

CARD 18b

COLUMN 1-12, 13-24, etc.

33. AE(K,I) - corresponding values of AE relative to the values of card 18a. AE(K,I) are floating point numbers.

CARD 19

COLUMN 1-12, 13-24, etc.

34. CDABD(K) - Drag coefficient times normal drag area of body at the bottom of each cable in feet squared. CDABD(K) are floating point numbers, where K=1, NCAB.

CARD 20

COLUMN 1-12, 13-24, etc.

35. WBD(K) - wet weight of body at the bottom of each cable in pounds. WBD(K) are floating point numbers, where K=1, NCAB.

CARD 21

COLUMN 1-3, 4-6, etc.

36. NBOD(K) - lift/drag table number which applies to the body at the bottom of each cable. If no body table, then NBOD=0. NBOD(K) are integer numbers, where K=1, NCAB.

CARD 22

COLUMN 1-2, 13-24, etc.

37. XX(I) - depth of each current point in feet. XX(I) are floating point numbers, where I=1, NCUR.

CARD 23

COLUMN 1-12, 13-24, etc.

38. YY(I) - current velocity at each point corresponding to depths of XX(I). Velocities are in knots and are floating point numbers, where I=1, NCUR.

Note: If NCUR=0, then values for XX(I), YY(I) are not input.

CARD 24

COLUMN 1-3

39. NPHI(N) - number of angle values for which lift/drag tables are generated. NPHI(N) are integer numbers.

COLUMN 4-6

40. NU(N) - number of velocity values for which lift/drag tables are generated. NU(N) are integer numbers.

CARD 25

41. PHIM(N,I) - angle in degrees of suspension cable from vertical for which lift/drag forces are generated. PHIM(N,I) are floating point numbers.

CARD 26

COLUMN 1-12, 13-24, etc.

42. U(N,J) - Velocity in knots for which lift/drag forces are generated. U(N,J) are floating point numbers.

CARD 27

COLUMN 1-12, 13-24, etc.

43. DDRAG(M) - drag values in pounds for a velocity and angle. DDRAG(M) are floating point numbers.

CARD 28

COLUMN 1-12, 13-24, etc.

44. DLIFT(M) - lift values in pounds for a velocity and angle. DLIFT(M) are floating point numbers.

Note: For variables 39 to 42, N=1 to NTAB, I=1 to NPHI(N), and J=1 to NU(N). For variables 43 and 44, M=1 to NP, where NP=NU(N) times NPHI(N).

REFERENCES

1. Wang, Henry T., and Moran, Thomas L., "Analysis of the Two-Dimensional Steady-State Behavior of Extensible Free Floating Cable Systems", NSRDC Report No. 3721, Oct 1971.

2. McEachern, James F., "A Modification to the Free Floating Extensible Cable System Computer Model(FF2E) to Consider Lift and Drag Forces on Intermediate Bodies", NADC Report No. NADC80178-30, 7 May 1980.

3. Holler, R. A. "Ocean Current Profile Definition", Research on Sonobuoy Configuration Annual Report 1976, pp 109-151, Naval Air Development Center, 1 Sep 1977.

4. Magnavox Report, "Updates and Inputs to the FF2E Cable Model", 20 August 1979.

APPENDIX I
FF2E LISTING

```
C     PROGRAM (INPUT,OUTPUT,INPUT5,OUTPUT6)
       CALL FPARAM (1,160)
       COMMON /BLK1/DB,DA,LB,LA,WB,CDB1,CDB2,FTANG,UDRIFT,H,DELTAS,
      &PRINTI,UWIND,CDA,EPSLON,TBH,TBV,NCAB,NHPHS,CDAPK,WPAK,EP2
       COMMON /BLK2/ XX(30), YY(30), NCUR
       COMMON / BLK5/ NPR(100),DC(100),WC(100),FLC(100),CDC(100),
      &TREF(100),P(100),CDABD(100),WBD(100),DCI(100),WCA(100),
       &WCB(100),YYK(30)
        COMMON /BLK6/ PHIM(10,7),U(10,10),L(10,10,7),D(10,10,7),NBOD(100),
       &NPHI(10),NU(10)
        COMMON /BLK7/ FAE(100,15),AE(100,15),NAE(100),JAM
        COMMON /BLK8/ NFOSB,FOSB(15),VOSB(15)
        DIMENSION CVFAC(100),CDINIT(100),DDRAG(75),DLIFT(75)
        REAL L,LA,LB
  500   FORMAT(6F12.6)
  502   FORMAT(6F12.4)
  504   FORMAT(4I3,5F12.5)
  506   FORMAT(2F12.5)
  508   FORMAT(4F12.5)
  510   FORMAT(2I5,2X,F12.5)
  512   FORMAT(24I3)
C
        PRINT," "
        PRINT," WOULD YOU LIKE INPUT PROMPTERS DISPLAYED ? (YES=1,NO=0)"
        READ(5,512) R
        IF(R.EQ.0) GO TO 2
C
C
C         NCASES IS THE NUMBER OF SETS OF DATA TO BE RAN
C
        PRINT,"INPUT NCASES"
    2   READ(5,512) NCASES
C
        IF(R.EQ.0) GO TO 4
        PRINT," "
C
C
C     IF NCUR = 0,THE 90% PROFILE IS ASSUMED
        PRINT,"IF NCUR = 0,THE 90% PROFILE IS ASSUMED "
C       NCAB = # OF CABLE SEGMENTS TO BE ENTERED.
C       NHPHS = # OF HYDROPHONES.
C       NTAB = # OF LIFT/DRAG TABLES.
        PRINT,"INPUT NCUR,NCAB,NHPHS,NTAB"
      4 READ(5,512) NCUR,NCAB,NHPHS,NTAB
C
        IF(P.EQ.0) GO TO 6
C
C
C        DAI = DIAMETER OF ANTENNA (IN)
C        LA = LENGTH OF ANTENNA (FT)
C        CDA = DRAG COEF. OF ANTENNA
C        TBH,TBV = HORIZONTAL AND VERTICAL COMPONENT OF TENSION (LBS)
C
        PRINT,
        PRINT," INPUT DAI LA CDA TBH,TBV"
      6 READ(5,500) DAI,LA,CDA,TBH,TBV
C
        IF(R.EQ.0) GO TO 8
C
C
C       DB = DIAMETER OF BUOY (FT)
```

```
      LB = LENGTH OF BOUY (FT)
      CDB1 = DRAG COEF. OF SUBMERGED PORTION OF BUOY
      CDB2 = DRAG COEF. OF BOUY PORTION ABOVE SEA SURFACE
      WB = WEIGHT OF SURFACE BUOY IN AIR (LBS)
      UWINDK = VELOCITY OF WIND (KNOTS)

      PRINT,
      PRINT ," INPUT DB,LB,CDB1,CDB2,WB,UWINDK"
    8 READ(5,500) DB,LB,CDB1,CDB2,WB,UWINDK

      IF(R.EQ.0) GO TO 10


      PRINT,
       NFOSB = # OF POINTS ON THE FORCE/FLOW CURVE OF SURFACE BUOY
      PRINT,"INPUT NFOSB"
   10 READ(5,512) NFOSB
      N=NFOSB
      IF(N.EQ.0) GO TO 16

      IF(R.EQ.0) GO TO 12


      PRINT,
       FOSB = FORCE VALUES ON FORCE/FLOW CURVE (LBS)
      PRINT,"INPUT FOSB(K)"
   12 READ(5,502) (FOSB(K),K=1,N)

      IF(R.EQ.0) GO TO 14


      PRINT,
       VOSB = FLOW VALUES ON THE FORCE/FLOW CURVE (KNOTS)
      PRINT,"INPUT VOSB(K)"
   14 READ(5,502) (VOSB(K),K=1,N)

   16 IF(R.EQ.0) GO TO 18


      PRINT,
       CDAPK = DRAG COEF. TIMES DRAG AREA OF PACKAGE BELOW SURFACE BUOY (FT*
              FT)
       WPAK = WET WEIGHT OF PACKAGE BELOW SURFACE BUOY (LBS)
      PRINT," INPUT CDAPK,WPAK"
   18 READ(5,500) CDAPK,WPAK

      IF(R.EQ.0) GO TO 20


       FLC = LENGTH OF EACH CABLE (FT)
      PRINT,
      PRINT," INPUT FLC(K)"
   20 READ(5,502) (FLC(K),K=1,NCAB)

      IF(R.EQ.0) GO TO 22


       NPR = # OF SEGMENTS INTO WHICH EACH CABLE IS DIVIDED
      PRINT,
      PRINT," INPUT NPR(K)"
   22 READ(5,512) (NPR(K),K=1,NCAB)

      IF(R.EQ.0) GO TO 24

```

```
C
C     DCI = DIAMETER OF EACH CABLE (IN)
      PRINT,
      PRINT," INPUT DCI(K)"
   24 READ(5,500) (DCI(K),K=1,NCAB)
C
      IF(R.EQ.0) GO TO 26
C
C
C     CVFAC = CONVERSION FACTOR FOR NON-CIRCULAR CROSS-SECTION OF CABLE
      PRINT,
   PRINT," INPUT CVFAC(K)"
   26 READ(5,500) (CVFAC(K),K=1,NCAB)
C
      IF(R.EQ.0) GO TO 28
C
C
C     WC = WEIGHT PER FOOT OF EACH CABLE IN WATER (LB/FT)
      PRINT,
      PRINT," INPUT WC(K)"
   28 READ(5,500) (WC(K),K=1,NCAB)
C
      IF(R.EQ.0) GO TO 30
C
C
C     CDC = DRAG COEF. OF EACH CABLE
      PRINT,
      PRINT," INPUT CDC(K)"
   30 READ(5,500) (CDC(K),K=1,NCAB)
C
      IF(R.EQ.0) GO TO 32
C
C
C     TREF = REFERENCE TENSION IN EACH CABLE (LB)
      PRINT,
      PRINT," INPUT TREF(K)"
   32 READ(5,502) (TREF(K),K=1,NCAB)
C
      IF(R.EQ.0) GO TO 34
C
C
C     P = POISSON RATIO FOR EACH CABLE
      PRINT,
      PRINT," INPUT P(K)"
   34 READ(5,500) (P(K),K=1,NCAB)
C
      IF(R.EQ.0) GO TO 36
C
C
C     NAE = # OF POINTS ON THE TENSION/AE CURVE
      PRINT,
      PRINT,"INPUT NAE(K)"
   36 READ (5,612) (NAE(K),K=1,NCAB)
      DO 45 K=1,NCAB
      N=NAE(K)
      IF(N.GT.1) GO TO 40
C
      IF(R.EQ.0) GO TO 38
C     INSERT ONE VALUE PER LINE
      PRINT,
      PRINT,"INPUT AE(K,1)"
   38 READ(5,502) AE(K,1)
C
      GO TO 45
```

```
 40  CONTINUE
C
     IF(R.EQ.0) GO TO 42
     PRINT,
C
C     FAE = TENSION VALUES ON THE TENSION/AE CURVE (LBS)
     PRINT,"INPUT FAE(K,LL)"
  42 READ(5,502) (FAE(K,LL),LL=1,N)
     IF(R.EQ.0) GO TO 44
C
C     AE = AE VALUES ON THE TENSION/AE CURVE
     PRINT,
     PRINT,"INPUT AE(K,LL)"
  44 READ(5,502) (AE(K,LL),LL=1,N)
 45  CONTINUE
C
     IF(R.EQ.0) GO TO 46
C
C
C     CDABD = DRAG COEF. TIMES DRAG AREA OF BODY (FT*FT)
     PRINT,
     PRINT," INPUT CDABD(K)"
  46 READ(5,500) (CDABD(K),K=1,NCAB)
C
     IF(R.EQ.0) GO TO 48
C
C
C     WBD = WEIGHT OF EACH BODY IN WATER (LBS)
     PRINT,
     PRINT," INPUT WBD(K)"
  48 READ(5,500) (WBD(K),K=1,NCAB)
C
      IF(R.EQ.0) GO TO 50
C
C
C      NBOD - LIFT/DRAG TABLE #
      PRINT,
      PRINT," INPUT NBOD(K)"
  50 READ(5,512) (NBOD(K),K=1,NCAB)
C
      IF(R.EQ.0) GO TO 52
C
C
C      XX = DEPTH OF EACH CURRENT POINT (FT)
      PRINT,
  52 IF(NCUR .EQ. 0) GO TO 58
      IF(R.EQ.0) GO TO 54
      PRINT," INPUT XX(K)"
  54 READ(5,502) (XX(K),K=1,NCUR)
      IF(P.EQ.0) GOTO 56
C
C
C       YY = VELOCITY AT EACH CURRENT POINT (KNOTS)
      PRINT,
      PRINT," INPUT YYY(K)"
  56 READ(5,500) (YYK(K),K=1,NCUR)
  58 IF(NTAB.LE.0)GO TO 75
C
C
      DO 73 N=1,NTAB
      SGU=0.
C
      IF(P.EQ.0) GO TO 60
C
```

```
C
C      NPHI = # OF ANGLE VALUES IN A LIFT/DRAG TABLE
C      NU = # OF CURRENT VALUES IN A LIFT/DRAG TABLE
      PRINT,
      PRINT," INPUT NPHI(N),NU(N)"
   60 READ(5,512) NPHI(N),NU(N)
      NPT=NPHI(N)
      NUT=NU(N)
C
      IF(R.EQ.0) GO TO 62
C
C
C      PHIM = ANGLE FROM VERTICAL IN THE LIFT/DRAG TABLE
      PRINT,
      PRINT," INPUT PHIM(N,I)"
   62 READ(5,502) (PHIM(N,I),I=1,NPT)
C
      IF(R.EQ.0) GO TO 64
C
C
C     U = VELOCITIES IN THE LIFT/DRAG TABLES (KNOTS)
      PRINT,
      PRINT," INPUT U(N,J)"
   64 READ(5,502) (U(N,J),J=1,NUT)
      NP=NPHI(N)*NU(N)
C
      IF(R.EQ.0) GO TO 66
C
C
C      DDRAG = DRAG VALUES FOR GIVEN VELOCITY AND ANGLE (LBS)
      PRINT,
      PRINT," INPUT DDRAG(M)"
   66 READ(5,500) (DDRAG(M),M=1,NP)
C
      IF(R.EQ.0) GO TO 68
C
C
C      DLIFT = LIFT VALUES FOR GIVEN VELOCITY AND ANGLE (LBS)
      PRINT,
      PRINT," INPUT DLIFT(M)"
   68 READ(5,500) (DLIFT(M),M=1,NP)
      DO 69 I=1,NUT
      SGU =SGU+U(N,I)
      DO 69 J=1,NPT
      INDX=J+(I-1)* NPHI(N)
      D(N,I,J)= DDRAG(INDX)
   69 L(N,I,J)=DLIFT(INDX)
C      COMPUTE AVERAGE CDA FOR BODY FROM TABULATED DATA
      SGD=0.
      SGUF= SGU*1.688/NU(N)
      DO 71 ID=1,NP
   71 SGD =SGD+DDRAG(ID)
      SGDA=SGD/NP
   73 CDINIT(N)=2.*SGDA/(1.9905*SGUF**2)
   75 EPSLON=0.0001
      EP2=0.0001
      FTANG=.020
      DA=DAI/12.0
      DO 77 J=1,NCAB
      DC(J)=DCI(J)/12.
C  ASSIGN INITIAL CDABD FROM AVERAGE TABULATED DRAG AND VELOCITY DATA
      IF(NBOD(J).LE.0)GO TO 77
      CDABD(J)=CDINIT(NBOD(J))
   77 CONTINUE
```

```
      DO 119 ICASE=1,NCASES
C      FIND THE MAXIMUM AND MINIMUM VALUES OF THE CURRENT
      TOTL=0.
      DO 79 J=1,NCAB
   79 TOTL=TOTL+FLC(J)
      TOTL=1.3*TOTL
      IF(NCUR .EQ. 0) GO TO 83
      UMAXK=-1000.
      UMINK=1000.
      DO 81 I=1,NCUR
      IF(YYK(I).GT.UMAXK) UMAXK=YYK(I)
      IF(YYK(I).LT.UMINK) UMINK=YYK(I)
      IF(XX(I).GT.TOTL) GO TO 82
   81 CONTINUE
   82 UMAX=1.688*UMAXK
      UMIN=1.688*UMINK
      IF(NCUR .GT. 0) GO TO 84
   83 UMAXK=1.853
      UMAX=1.688*UMAXK
C      CALL CUR TO CALCULATE UMIN AT DEPTH=TOTL
      CALL CUR(TOTL,UMIN)
   84 CONTINUE
  600 FORMAT(// " SURFACE UNIT DRAG...."/)
  601 FORMAT(3X,"    FORCE(LB) ",15F8.3)
  602 FORMAT(3X,"VS. FLOW(KTS) ",15F8.3)
  603 FORMAT(1H1,64H*****LISTING OF CABLE AND OCEAN ENVIRONMENT CHARACTE
     &RISTICS*****)
  605 FORMAT(1X,5X,13HDIAMETER (FT),46X,F8.5 )
  604 FORMAT(1X,17HSURFACE BUOY.....)
  606 FORMAT(6X,24HDIAMETER OF ANTENNA (IN),36X,F7.5)
  607 FORMAT(6X,11HLENGTH (FT),44X,F12.5)
  608 FORMAT(6X,22HLENGTH OF ANTENNA (FT),37X,F8.5)
  609 FORMAT(6X,45HDRAG COEFFICIENT FOR BUOY (SUBMERGED PORTION),15X,F7.
     &5)
  610 FORMAT(6X,44HDRAG COEFFICIENT FOR BUOY (SURFACED PORTION),16X,F7.5
     &)
  611 FORMAT(6X,28HDRAG COEFFICIENT FOR ANTENNA,32X,F7.5)
  612 FORMAT(1X,18HWIND SPEED (KNOTS),45X,F9.5)
  613 FORMAT(1X,18HOCEAN PROFILE.....)
  614 FORMAT(5X,1X,10HDEPTH (FT),10X,15HCURRENT (KNOTS))
  615 FORMAT(1X,8X,F8.2,13X,F8.4)
  616 FORMAT(4X,"MAXIMUM CURRENT (FT/S) = ",F8.4,4X,"MINIMUM CURRENT
     &(FT/S) = ",F8.4)
  617 FORMAT(1X,5X,19HWEIGHT IN AIR (LBS),40X,F8.5)
  618 FORMAT(1X)
  619 FORMAT(4X,"THE TRUE 90% PROFILE IS ASSUMED")
  620 FORMAT(1H1,10HRUN NUMBER,1X,I3)
  621 FORMAT(1X,5X,31HBOTTOM HORIZONTAL TENSION (LBS),27X,F9.5)
  622 FORMAT(1X,5X,29HBOTTOM VERTICAL TENSION (LBS),29X,F9.5)
  623 FORMAT(6X,16HCABLE PROPERTIES,84X,15HBODY PROPERTIES)
  624 FORMAT(1X,3HNUM,2X,10HLENGTH(FT),14X,10HT REF(LBS),
     &3X,8HDIAM(IN),3X,10HWT/FT(LBS),3X,9HDRAG COEF,2X,
     &13HPOISSON RATIO,2X,10HVOL FACTOR,4X,9HCDA(FTSQ),4X,7HWT(LBS),3X,5
     &HTABLE)
  625 FORMAT(1X,I3,F12.4,12X,F12.4,4F12.6,2X,F9.6,5X,F11.6,1X,F11.6,
     &3X,I2)
  626 FORMAT(1X,5X,21HNUMBER OF HYDROPHONES,37X,I3)
  627 FORMAT(6X,25HPACKAGE DRAG AREA (FT SQ),30X,F12.5)
  628 WRITE(6,603)
  629 FORMAT(6X,29HPACKAGE WEIGHT IN WATER (LBS),26X,F12.5)
  630 FORMAT(// " CABLE ELASTICITY....."/)
  631 FORMAT(I3/3X" CONST. AE ",F10.2)
  632 FORMAT(I3/3X " FORCE (LB) ",15F8.2)
  633 FORMAT(    3X " VS. AE      ",15F8.2)
```

```
 634  FORMAT(/,26X,28HBODY LIFT AND DRAG TABLE NO.,I3,52HLIFT(LBS),
     &DRAG(LBS) VS ANGLE(DEG) AND VELOCITY(KNOT)
 635  FORMAT(13X,6(F6.3,4HDEG.,8X),F6.3,4HDEG.)
 636  FORMAT(13X,6(F6.3,4HDEG.,8X))
 637  FORMAT(13X,5(F6.3,4HDEG.,8X))
 638  FORMAT(13X,4(F6.3,4HDEG.,8X))
 639  FORMAT(13X,3(F6.3,4HDEG.,8X))
 640  FORMAT(13X,2(F6.3,4HDEG.,8X))
 641  FORMAT(13X,F6.3,4HDEG.)
 642  FORMAT(1X,8HVELOCITY,2X,7(4HDRAG,5X,4HLIFT,5X))
 643  FORMAT(1X,8HVELOCITY,2X,6(4HDRAG,5X,4HLIFT,5X))
 644  FORMAT(1X,8HVELOCITY,2X,5(4HDRAG,5X,4HLIFT,5X))
 645  FORMAT(1X,8HVELOCITY,2X,3(4HDRAG,5X,4HLIFT,5X))
 646  FORMAT(1X,8HVELOCITY,2X,4(4HDRAG,5X,4HLIFT,5X))
 647  FORMAT(1X,8HVELOCITY,2X,2(4HDRAG,5X,4HLIFT,5X))
 648  FORMAT(1X,8HVELOCITY,2X,1(4HDRAG,5X,4HLIFT,5X))
 649  FORMAT(1X,8H********,2X,7(4H****,5X,4H****,5X))
 650  FORMAT(1X,8H********,2X,6(4H****,5X,4H****,5X))
 651  FORMAT(1X,8H********,2X,5(4H****,5X,4H****,5X))
 652  FORMAT(1X,8H********,2X,4(4H****,5X,4H****,5X))
 653  FORMAT(1X,8H********,2X,3(4H****,5X,4H****,5X))
 654  FORMAT(1X,8H********,2X,2(4H****,5X,4H****,5X))
 655  FORMAT(1X,8H********,2X,4H****,5X,4H****)
 656  FORMAT(1X,3(F6.3,3X))
 657  FORMAT(1X,F6.3,3X,2(F6.3,3X,F6.3,3X))
 658  FORMAT(1X,F6.3,3X,3(F6.3,3X,F6.3,3X))
 659  FORMAT(1X,F6.3,3X,4(F6.3,3X,F6.3,3X))
 660  FORMAT(1X,F6.3,3X,5(F6.3,3X,F6.3,3X))
 661  FORMAT(1X,F6.3,3X,6(F6.3,3X,F6.3,3X))
 662  FORMAT(1X,F6.3,3X,7(F6.3,3X,F6.3,3X))
      WRITE (6,618)
      WRITE (6,618)
      WRITE (6,604)
      CONTINUE
      WRITE (6,605) DB
      WRITE (6,606) DAI
      WRITE(6,617) WB
      WRITE (6,607) LB
      WRITE (6,608) LA
      WRITE (6,609) CDB1
      WRITE (6,610) CDB2
      WRITE (6,611) CDA
      WRITE(6,627) CDAPK
      WRITE(6,629) WPAK
      WRITE (6,618)
      WRITE(6,612) UWINDK
      WRITE (6,618)
      N=NFOSB
      IF(N.EQ.0) GO TO 86
      WRITE(6,600)
      WRITE(6,618)
      WRITE(6,601) (FOSB(K),K=1,N)
      WRITE(6,602) (VOSB(K),K=1,N)
      WRITE(6,618)
 86   IF(NCUR.EQ.0) GOTO 88
      WRITE(6,618)
      WRITE(6,613)
      WRITE (6,614)
      WRITE(6,615) (XX(I),YYK(I),I=1,NCUR)
      WRITE(6,618)
      IF(NCUR.GT.0) GO TO 90
 88   WRITE(6,619)
      WRITE(6,618)
      WRITE(6,619)
```

```
      WRITE(6,618)
 90   WRITE(6,616) UMAX,UMIN
      WRITE (6,618)
      WRITE(6,623)
      WRITE(6,624)
      DO 91 K=1,NCAB
      WRITE(6,625) K,FLC(K),TREF(K),DCI(K),WC(K),CDC(K),P(K),
     &CVFAC(K),CDABD(K),WBD(K),NBOD(K)
 91   CONTINUE
      WRITE(6,630)
      DO 95 K=1,NCAB
      N=NAE(K)
      IF(N.GT.1) GO TO 93
      WRITE(6,631) K,AE(K,1)
      GO TO 95
 93   WRITE(6,632) K,(FAE(K,I),I=1,N)
      WRITE(6,633) (AE(K,I),I=1,N)
 95   CONTINUE
      IF(NTAB.LE.0)GO TO 112
      DO 111 N=1,NTAB
      WRITE(6,634) N
      NT=NU(N)
      GO TO (96,98,100,102,104,106,108) , NPHI(N)
 96   WRITE(6,641) PHIM(N,1)
      WRITE(6,648)
      WRITE(6,655)
      DO 97 I=1,NT
 97   WRITE(6,656) U(N,I),D(N,I,1),L(N,I,1)
      GO TO 111
 98   WRITE(6,640) (PHIM(N,J),J=1,2)
      WRITE(6,644)
      WRITE(6,647)
      WRITE(6,654)
      DO 99 I=1,NT
 99   WRITE(6,657) U(N,I),(D(N,I,J),L(N,I,J),J=1,2)
      GO TO 111
 100  WRITE(6,639) (PHIM(N,J),J=1,3)
      WRITE(6,645)
      WRITE(6,653)
      DO 101 I=1,NT
 101  WRITE(6,658) U(N,I),(D(N,I,J),L(N,I,J),J=1,3)
      GO TO 111
 102  WRITE(6,638) (PHIM(N,J),J=1,4)
      WRITE(6,646)
      WRITE(6,652)
      DO 103 I=1,NT
 103  WRITE(6,659) U(N,I),(D(N,I,J),L(N,I,J),J=1,4)
      GO TO 111
 104  WRITE(6,637) (PHIM(N,J),J=1,5)
      WRITE(6,644)
      WRITE(6,651)
      DO 105 I-1,NT
 105  WRITE(6,660) U(N,I),(D(N,I,J),L(N,I,J),J=1,5)
      GO TO 111
 106  WRITE(6,636) (PHIM(N,J),J=1,6)
      WRITE (6,643)
      WRITE(6,650)
      DO 107 I=1,NT
 107  WRITE(6,661) U(N,I),(D(N,I,J),L(N,I,J),J=1,6)
      GO TO 111
 108  WRITE(6,635) (PHIM(N,J),J=1,7)
      WRITE(6,642)
      WRITE(6,649)
      DO 109 I=1,NT
```

```
109   WRITE(6,662) U(N,I),(D(N,I,J),L(N,I,J),J=1,7)
111   WRITE(6,618)
112   WRITE(6,618)
      WRITE(6,626) NHPHS
      WRITE(6,621) TBH
      WRITE(6,622) TBV
      WRITE(6,620) ICASE
C      CONVERT FROM KNOTS TO FEET PER SECOND
      UWIND=1.688*UWINDK
      IF(NCUR.EQ.0) GO TO 114
      DO 113 I=1,NCUR
      YY(I)=1.688*YYK(I)
113   CONTINUE
114   CONTINUE
      IF(NTAB.LE.0)GO TO 116
      DO 115 N=1,NTAB
      NT=NU(N)
      DO 115 K=1,NT
      U(N,K)=1.688*U(N,K)
115   CONTINUE
116   DO 117 J=1,NCAB
      ALPHA=CVFAC(J)*0.25*64.043*3.14159
      WCB(J)=ALPHA*DC(J)*DC(J)
      WCA(J)=WC(J)+WCB(J)
117   CONTINUE
      IRUN=1
C    CALL THE ROUTINE STEADY TO CALCULATE THE CONFIGURATION.
      CALL STEADY(IRUN,UMAX,UMIN)
119   CONTINUE
      STOP
      END
C
      SUBROUTINE STEADY(IRUN,DLIMIT,UDMIN)
C      THIS ROUTINE CALCULATES THE STEADY STATE SCOPE
      DIMENSION CRELKT(400)
      DIMENSION FLIFT(10),BDRAG(10)
      DIMENSION S(400),X(400),Y(400),PHI(400),PHID(400),T(400)
      DIMENSION XX(400),YY(400),Y0(5),PHIV(400),BPHI(400),SE(400)
      DIMENSION BPHIV(400),HGRID(400),UGRID(400),SAR(400)
      COMMON /BLK1/ DB,DA,LB,LA,WB,CDB1,CDB2,FTANG,UDRIFT,H,DELTAS,
     &PRINTI,UWIND,CDA,EPSLON,TBH,TBV,NCAB,NHPHS,CDAPK,WPAK,EP2
      COMMON /BLK3/ FIRST
      COMMON /BLK4/ DRAG,WPLA,WPLB,FFTANG,DRIFT,TREFC,PC
      COMMON /BLK5/ NPR(100),DC(100),WC(100),FLC(100),CDC(100),
     &TREF(100),P(100),CDABD(100),WBD(100),DCI(100),WCA(100),
     &WCB(100)
      COMMON /BLK6/ PHIM(10,7),U(10,10),L(10,10,7),D(10,10,7),NBOD(100),
     &NPHI(10),NU(10)
      COMMON /BLK7/ FAE(100,15),AE(100,15),MAE(100),JAM
      COMMON /BLK8/ NFOSB,FOSB(15),VOSB(15)
      REAL LA,LB
      DATA PI,RHO,RHOAIR,GAMMA,RADIAN
     &/3.14159,1.9905,.002378,64.043,57.29578  /
341   FORMAT(1X,9HITERATION,1X,10HDRIFT(KTS),1X,9HDRAFT(FT),1X,
     &9HDRAG(BOT),2X,7HWT(BOT),4X,10HERROR(HOR),1X,11HERROR(VERT),1X,
     &8HTEN(BOT),1X,8HPHI(BOT),5X,6HTZEROX,4X,6HTZEROY,4X,5HDELTA)
200   FORMAT(1X,22HASSUMED DRIFT SPEED IS,1X,F7.5,1X,6HKNOTS.)
201   FORMAT(1X,25HASSUMED BUOY DRAFT, H, IS,1X,F7.5,1X,5HFEET.)
202   FORMAT(1X)
203   FORMAT(1X,4X,9HS REF(FT),4X,9HS STR(FT),6X,5HX(FT),8X,5HY(FT),
     &6X,8HPHI(DEG),6X,6HT(LBS),7X,9HPHIV(DEG),6X,8HCREL(KT),5X,9HLIFT
     &(LBS),5X,9HDRAG(LBS))
204   FORMAT(1X,3X,F8.2,5X,F8.2,5X,F9.2,1X,F8.2,5X,F8.2,5X,F8.2,6X,
     &F8.2,6X,F8.4)
```

```
205  FORMAT(1H1,16HITERATION NUMBER,1X,I3)
206  FORMAT(1X,36HTHE TOTAL WEIGHT OF BOTTOM WEIGHT IS,
    &1X,F9.5,1X,4HLBS.)
207  FORMAT(1X,45HTHE COMPUTED VERTICAL COMPONENT OF TENSION IS,1X,F9.5
    &,1X,4HLBS.)
208  FORMAT(1X,38HTHE TOTAL DRAG OF THE BOTTOM WEIGHT IS,
    &1X,F9.5,1X,4HLBS.)
209  FORMAT(1X,47HTHE COMPUTED HORIZONTAL COMPONENT OF TENSION IS,1X,
    &F9.5,1X,4HLBS.)
210  FORMAT(1H1,70H*CONFIGURATION OF THE LOWER ARRAY. POINT (0.0,0.0)
    & IS THE BOTTOM*)
211  FORMAT(1X,5X,6H S(FT),8X,5HX(FT),8X,5HY(FT),6X,8HPHI(DEG),
    &6X,8HPSI(DEG))
212  FORMAT(1X,3X,F8.2,4(5X,F8.2))
213  FORMAT(1X,18HTHE ANGLE THETA IS,1X,F5.2,1X,8HDEGREES.)
214  FORMAT(1X,48HTHE MAXIMUM PERPENDICULAR DISTANCE FROM CHORD IS,1X,F
    &5.2,1X,5HFEET.)
215  FORMAT(1X,15HBOTTOM ANGLE IS,1X,F7.2,5X,17HBOTTOM TENSION IS,1X,
    & F6.2)
216  FORMAT(1X,I3,3X,F11.7,F11.7,9F11.5)
220  FORMAT(1X,42HREVERSAL IN SIGN BETWEEN DELTAU AND ERRORH)
222  FORMAT(1X,28HSTART OF SIMULTANEOUS SCHEME)
223  FORMAT(1X,25HSTART OF STAGGERED SCHEME)
230  FORMAT(110X,24HCOMPUTED BODY PROPERTIES)
231  FORMAT(1X,3X,F8.2,5X,F8.2,5X,F9.2,4X,F8.2,5X,F8.2,5X,F8.2,6X,
    &F8.2,6X,F8.4,6X,F8.3,6X,F8.3)
232  FORMAT(1X,3X,F8.2,5X,F8.2,5X,F9.2,4X,F8.2,5X,F8.2,5X,F8.2,6X,
    &F8.2,6X,F8.4,7X,5H*BODY,I3,5X,8HNO LIFT*)
     WRITE(6,202)
     WRITE(6,341)
     DLLIMIT=DLIMIT
     UDDMIN=UDMIN
     WBOT=WBD(NCAB)
     GPRBSQ=GAMMA*PI*.25*DB*DB
     GPIOV4=GAMMA*PI*.25
     ILAST=0
     HMIN=WB/GPRBSQ
     UDRIFT=UDDMIN+0.5*(DLIMIT-UDDMIN)
     DLIMIT=1.2*DLLIMIT
C  LET INITIAL BUOYANCY BE THE WEIGHT OF EVERYTHING UNDER THE BUOY.
     XNPHS=NHPHS
     DELTA=1.
     BCY=0.
     DO 253 J=1,NCAB
     BCY=BCY+FLC(J)*WC(J)+WBD(J)
253  CONTINUE
     BCY=BCY+WPAK+TBV
     IF(BCY.LE.0.) BCY=0.
     H=(BCY+WB)/GPRBSQ
     UMAX=DLIMIT
     UMIN=UDDMIN
     UMIN1=UMIN
     UMAX1=UMAX
     WRITE(6,202)
     PRV=0.
     PRH=0.
     ABSERH=0.
     ABSERV=0.0
     PERH=15.
     PERV=15.
     EPRIME=100.
     BRSLT=EPRIME
     DEN4=LB*CDB1*DB+CDAPK
     DO 20 J=1,NCAB
```

```
      DEN4=DEN4+CDC(I)*D2(I)*...
      DEN4=DEN4+CDAE(I)
20    CONTINUE
      DENS=RHO*DEN4
      IB=0
      JSEN=1
      K2=0
      K3=0
      I2MANT=0
      IRUN=1
      IND=1
      KIT=0
      KUSTOP=22
      IFRST=11
      KUD=0
      KH=0
      KREV=0
      DHFAC=0.6
      FFERV=1000.
      HMINF=HMIN
      HMAXF=LB
      F=0.0
2     CONTINUE
      IF(H.LT.HMIN) H=1.01*HMIN
      BCY=GFRBSQ*H-WB-WFAK
      HTEMP=H
8     FIRST=-100.0
C     CONVERT FROM FEET PER SECOND TO KNOTS.
      IF(UDRIFT.LT. 0.0) UDRIFT=0.0
      UDKNTS=.5924*UDRIFT
      DEP=0.5*H
      CALL CUR(DEP,COFY)
      CZERO=COFY
      FIRST=100.0
      CREL=COFY-UDRIFT
      DRAGW=.5*RHOAIR*DB*CDB2*(UWIND-UDRIFT)*ABS(UWIND-UDRIFT)*(LB-H)
     $+.5*RHOAIR*CDA*(UWIND-UDRIFT)*ABS(UWIND-UDRIFT)*LA*DA
9     DRAGB=.5*RHO*DB*CDB1*CREL*ABS(CREL)*H
      CALL CUR(H,COFY)
      CRELP=COFY-UDRIFT
      DRGPK=0.5*RHO*CDAPK*CRELP*ABS(CRELP)
      DRAGB=DRAGB+DRGPK
      IF(NFOSB.EQ.0) GO TO 10
C
      CALL VAR(DRAGB,CREL,H,DB,CDB1,CD)
      IF(CD.EQ.CDB1) GO TO 10
      CDB1=CD
      GO TO 9
10    DRAGB=DRAGB+DRAGW
C     INITIAL TENSION IS THE RESULTANT OF BUOYANCY AND DRIFT.
      T(1)=SQRT(BCY*BCY+DRAG*DRAG)
C     INITIAL ANGLE IS THE ANGLE WHOSE TANGENT IS BUOYANCY DRAG OF BODY.
      PHI(1)=ATAN2(BCY,DRAG)
      ...
      PHI(1) ... RADIAN
      ...
      SE(1)=0.0
      Y(1)=...
      NLAST=0
      ...
      ...
      DO 240 J=1,NIAS
      DRAG=0.5*RHO*...
      START ...
      ...
```

Copy available to DTIC does not permit fully legible reproduction

```
      WPLB=WCB(J)
      TREFC=TREF(J)
      JAM=J
      PC=P(J)
      FNP=NPR(J)
      SPA=FLC(J)/FNP
      N1=NLAST+2
      NLAST=N1+NPR(J)-1
      DO 250 M=N1,NLAST
      MINDEX=M
      Y0(1)=T(M-1)
      Y0(2)=PHI(M-1)
      Y0(3)=X(M-1)
      Y0(4)=Y(M-1)
      Y0(5)=SE(M-1)
      SS=S(M-1)
      CALL KUTMER(5,SS,Y0,EPSLON,SPA,START,HCX,EP2)
      T(M)=Y0(1)
      PHI(M)=Y0(2)
      PHID(M)=PHI(M)*RADIAN
      IF(KIT-1) 952,951,952
 951  IF((PHID(M).GT.125.).AND.(K2.LE.4)) GO TO 800
      IF((PHID(M).LT.0.).AND.(K2.LE.4)) GO TO 820
 952  X(M)=Y0(3)
      Y(M)=Y0(4)
      SE(M)=Y0(5)
      S(M)=SS
 250  CONTINUE
      CALL CUR(Y(NLAST),COFY)
      CREL=COFY-UDRIFT
      IF(NBOD(J).LE.0) GO TO 3000
      CALL BODY(CREL,PHID(NLAST),CDABD(J),NBOD(J),WBD(J),FLIFT(J),BDRAG(
     &J),IRUN,JAM)
 3000 DRAGH=0.5*RHO*CDABD(J)*CREL*ABS(CREL)
      XCOMP=DRAGH+T(NLAST)*COS(PHI(NLAST))
      YCOMP=-WBD(J)+T(NLAST)*SIN(PHI(NLAST))
      T(NLAST+1)=SQRT(XCOMP**2+YCOMP**2)
      PHI(NLAST+1)=ATAN2(YCOMP,XCOMP)
      PHID(NLAST+1)=PHI(NLAST+1)*RADIAN
      X(NLAST+1)=X(NLAST)
      Y(NLAST+1)=Y(NLAST)
      S(NLAST+1)=S(NLAST)
      SE(NLAST+1)=SE(NLAST)
 240  CONTINUE
 95   MPRINT=NLAST
      THORIZ=T(MPRINT)*COS(PHI(MPRINT))
      TVERT=T(MPRINT)*SIN(PHI(MPRINT))
      CALL CUR(Y(MPRINT),COFY)
      I2MANY=0
      CREL=COFY-UDRIFT
      WBOT=WBD(NCAB)
      DRAGBT=0.5*RHO*CDABD(NCAB)*CREL*ABS(CREL)
      IF(ABS(DRAGBT).LT.0.001) DRAGBT=0.001
C   CHECK BOTTOM CONDITIONS WITH REALITY.
      PPERV=PRV
      PPERH=PRH
      ERRORV=TVERT-WBOT-TBV
      ERRORH=THORIZ+DRAGBT-TBH
      PRV=ERRORV
      PRH=ERRORH
      ABSERH=ABS(ERRORH)
      ABSERV=ABS(ERRORV)
      RESULT=SQRT(ERRORH**2+ERRORV**2)
 799  RATIO1=ABS(ERRORH/DRAGBT)
```

```
      RATIO2=ABS(ERRORV/(WBOT+TBV))
      DRGTBH=DRAGBT+TBH
      RATIO3=ABS(ERRORH/DRGTBH)
      WRITE(6,202)
      WRITE(6,216)INO,UDKNTS,HTEMP,DRAGBT,WBOT,ERRORH,ERRORV,T(MPRINT),
     &PHID(MPRINT),DRAGB,BCY,DELTA
      IF(RESULT-BRSLT) 903,903,906
 903  BH=H
      BUDR=UDRIFT
      BRSLT=RESULT
 906  IF((RATIO3.LE..02).AND.(RATIO2.LE..02)) GO TO 150
      IF(ABS(DRGTBH)-0.30) 2055,2055,2056
 2055 IF((RATIO3.LE.0.10).AND.(RATIO2.LE..02)) GO TO 150
 2056 CONTINUE
 98   IRUN=IRUN+1
      INO=IRUN
      UTEMP=UDRIFT
      IF(IB.GT.1) GO TO 150
      IF(IRUN.GT.50) GO TO 111
      GO TO 112
 111  IF(F-ERRORH) 139,960,139
 139  F=ERRORH
 112  IF(INO.GT.400) GO TO 960
      IF(KIT-1)  4860,650,4860
 4860 IF(KIT-2) 120,121,121
 121  KUD=KUD+1
      IF((RATIO3.LE.0.02)) GO TO 102
      IF((KUD.GT.KUSTOP).AND.(RATIO2.GT.0.02)) GO TO 102
      IF(((PRH/PPERH).GT.1.).AND.(KUD.GE.2)) GO TO 2040
 120  IF((RATIO3.LE.0.02).AND.(KIT.EQ.0)) GO TO 1651
      IF((KUD.GT.KUSTOP).AND.(KIT.EQ.0)) GO TO 1651
 303  IF(ERRORH.GT.0.0) GO TO 100
 101  UDRIFT=.5*(UDRIFT+UMIN)
      UMAX=UTEMP
      GO TO 2
 100  UDRIFT=.5*(UDRIFT+UMAX)
      UMIN=UTEMP
      GO TO 2
 102  KH=KH+1
      KUD=0
      WRITE(6,202)
      IF((-ERRORV/PRERV).GT.0.5) DHFAC=0.5*DHFAC
      IF((ERRORV/PRERV).GT.0.7) DHFAC=1.5*DHFAC
      PRERV=ERRORV
      IF((ERRORV.GT.0.0).AND.(PHID(MPRINT).LT.180.)) GO TO 2002
 104  H=HTEMP+DHFAC*ABSERV/GPRBSQ
      IF(H.GE.HMAXP) H=HTEMP+0.75*(HMAXP-HTEMP)
      HHT=H/HTEMP
      HTH=HTEMP/H
      HMINP=HTEMP
      UMAX=(HHT+.05)*UDRIFT
      UMIN=(HTH-.05)*UDRIFT
      GO TO 2032
 2002 H=HTEMP-DHFAC*ABS(ERRORV)/GPRBSQ
      IF(H.LT.(0.5*(HTEMP+HMIN))) H=0.5*(HTEMP+HMIN)
      IF(H.LE.HMINP) H=HTEMP-0.75*(HTEMP-HMINP)
      HMAXP=HTEMP
      HHT=H/HTEMP
      HTH=HTEMP/H
      UMAX=(HTH+.05)*UDRIFT
      UMIN=(HHT-.05)*UDRIFT
 2032 PREVH=HTEMP
      PREVU=UDRIFT
      UDRIFT=1.01*UDRIFT
```

```
      UMAX1=UMAX
      UMIN1=UMIN
      GO TO 2
 2040 KREV=KREV+1
      IF(KREV.GT.10) GO TO 960
      WRITE(6,220)
      GO TO 303
 1651 KIT=1
      UD1=UDRIFT
      H1=H
      EV1=ERRORV
      UMAX=UMAX+ABS(ERRORV)*UDRIFT*.03
      UMIN=UMIN-ABS(ERRORV)*UDRIFT*.03
      IF(WBOT.LT.0.) GO TO 1001
      WRITE(6,202)
      WRITE(6,222)
       GO TO 650
   99  FIRST=-100.
      DO 97 I=1,MPRINT
      CALL CUR(Y(I),COFY)
      FIRST=100.
      CRELKT(I)=(COFY-UDRIFT)*0.5924
      PHIV(I)=90.0-PHID(I)
 97   CONTINUE
      FIRST=-100.
      HHALF=.5*HTEMP
      CALL CUR(HHALF,COFY)
      FHALF=(COFY-UDRIFT)*.5924
      FIRST=100.
C
      WRITE(6,205) INO
      WRITE(6,200) UDKNTS
      WRITE (6,209) DRAGB
      WRITE(6,201) HTEMP
      WRITE (6,207) BCY
      WRITE(6,219) FHALF
  219 FORMAT(1X,"FLOW PAST THE SURFACE UNIT IS",F8.4,1X,"KNOTS")
      WRITE(6,224) CDB1
  224 FORMAT(1X,"CD OF SURFACE UNIT IS",F8.4)
      WRITE(6,202)
      WRITE(6,230)
      WRITE(6,203)
      I=0
      K=0
 80   I=I+1
      IF(I.GE.MPRINT)GO TO 81
      IF(S(I).GE.S(I+1))GO TO 81
      WRITE(6,204)S(I),SE(I),X(I),Y(I),PHID(I),T(I),PHIV(I),CRELKT(I)
      GO TO 80
 81   K=K+1
      IF(NBOD(K).LE.0)GO TO 83
      WRITE(6,231)S(I),SE(I),X(I),Y(I),PHID(I),T(I),PHIV(I),CRELKT(I),FL
     &IFT(K),BDRAG(K)
      GO TO 87
 83   WRITE(6,232)S(I),SE(I),X(I),Y(I),PHID(I),T(I),PHIV(I),CRELKT(I),K
 87   IF(I.LT.MPRINT)GO TO 80
      WRITE(6,202)
      TOTV=WBOT+TBV
      WRITE(6,206) TOTV
      WRITE(6,207) TVERT
      WRITE(6,202)
      TOTH=DRAGBT-TBH
      WRITE(6,208) TOTH
      WRITE(6,209) THORIZ
```

```
C
      GO TO 130
  650 K2=0
      IF((IRUN.GT.35).AND.(RESULT.GT.EPRIME).AND.(EPRIME.GT.0.20)) GO
     &TO 1001
      IF(RESULT.GT.EPRIME) GO TO 660
  901 PERV=ERRORV
      PERH=ERRORH
  905 EPRIME=RESULT
      PUDRFT=UDRIFT
      PH=H
      DELTA=1.
      USEN=1.
  907 DELTAH=-DELTA*ABSERV*ERRORV/(RESULT*GPRBSQ)
      DELTAU=USEN*DELTA*ERRORH*ABSERH/(RESULT*DEN5)
      DELTAU=DELTAU/UDRIFT
  721 IF(DELTAU) 651,651,652
  651 UMAX=UDRIFT
      UMIN=UDRIFT+DELTAU
      GO TO 653
  652 UMAX=UDRIFT+DELTAU
      UMIN=UDRIFT
  653 UDRIFT=UDRIFT+DELTAU
      H=H+DELTAH
      HINT=0.7
      IF(H.LT.(HTEMP-HINT*(HTEMP-HMIN))) H=(HTEMP-HINT*(HTEMP-HMIN))
  969 GO TO 2
  660 IF((DELTA.LT.0.05).OR.(USEN.GT.500.)) GO TO 901
      IF((DELTA.LT.0.3).AND.(IRUN.LT.25)) GO TO 901
      IF(EPRIME.LT.0.1.AND.IRUN.LT.30) GO TO 901
      EHPH=PRH/PERH
      IF((EHPH.GT.1.).AND.(ABS(PRH).GT.ABS(PRV))) GO TO 901
      AEHPH=ABS(EHPH)
      ERRORH=PERH
      ERRORV=PERV
      ABSERV=ABS(ERRORV)
      ABSERH=ABS(ERRORH)
      RESULT=EPRIME
      UDRIFT=PUDRFT
      H=PH
      ARVPV=ABS(PRV/PERV)
  914 DELTA=0.5*DELTA
      USEN=1.
      GO TO 907
 1001 KIT=5
      HTEMP=H1
      UTEMP=UD1
      UDRIFT=UD1
      ERRORV=EV1
      KUD=0
      WRITE(6,202)
      WRITE(6,223)
      GO TO 102
  960 IB=15
      H=BH
      UDRIFT=BUDR
      GO TO 2
  800 UMAX=UDRIFT
      I2MANY=I2MANY+1
      MPRINT=MINDEX
      IF(I2MANY.GT.7) GO TO 962
      UDRIFT=.5*(UDRIFT+UMIN)
      GO TO 2
  820 UMIN=UDRIFT
```

```
      I2MANY=I2MANY+1
      MPRINT=MINDEX
      IF(I2MANY.GT.7) GO TO 962
      UDRIFT=.5*(UDRIFT+UMAX)
      GO TO 2
962   K2=K2+1
      K3=K3+1
      IF(K3.GT.15) GO TO 1ØØ1
      I2MANY=Ø
      UMAX=UMAX+Ø.Ø3*UDRIFT
      UMIN=UMIN-Ø.Ø3*UDRIFT
      GO TO 2
15Ø   ILAST=1Ø
      GO TO 99
13Ø   XX(1)=Ø.Ø
      YY(1)=Ø.Ø
      SAR(1)=Ø.Ø
      BPHI(1)=PHID(MPRINT)
      IF(NHPHS.LE.1) GO TO 133
      IARRAY=NHPHS
      DO 255 MK=1,NHPHS
      KK=MK-1
      NMKK=NCAB-KK
255   IARRAY=IARRAY+NPR(NMKK)
      DO 131 I=1,IARRAY
      II=MPRINT-I
      BPHI(I+1)=PHID(II)
      XX(I+1)=-X(MPRINT)+X(II)
      YY(I+1)=Y(MPRINT)-Y(II)
      SAR(I+1)=S(MPRINT)-S(II)
131   CONTINUE
      THETA1=ATAN2(YY(IARRAY+1),XX(IARRAY+1))
      THETAD=9Ø.Ø-RADIAN*THETA1
      RMAX=Ø.Ø
      I32=IARRAY+1
      DO 132 I=2,I32
      R=SQRT(XX(I)**2+YY(I)**2)
      THETA2=ATAN2(YY(I),XX(I))
      Z=R*ABS(SIN(THETA2-THETA1))
      IF(Z.GT. RMAX)  RMAX=Z
132   CONTINUE
      DO 14Ø I=1,I32
      BPHIV(I)=9Ø.Ø-BPHI(I)
14Ø   CONTINUE
      WRITE(6,2Ø2)
      WRITE(6,213) THETAD
      WRITE(6,214) RMAX
133   CONTINUE
      RETURN
      END
C
C      THIS ROUTINE CALCULATES FLOW FOR A GIVEN DEPTH.
      SUBROUTINE CUR(X, FOFX)
      COMMON /BLK2/ XX(3Ø), YY(3Ø), NCUR
      COMMON /BLK3/ FIRST
      IF(NCUR .EQ. Ø) GO TO 85
      IF(FIRST.LT. Ø.Ø) I=1
      IF(X.LT.Ø.) GO TO 8Ø
      IF(( X.GE.XX(I)).AND.(X.LE.XX(I+1))) GO TO 3Ø
      IF((X.GE.XX(I-1)).AND.(X.LE.XX(I)) ) GO TO 4Ø
      IF((X.GE.XX(I+1)).AND.(X.LE.XX(I+2))) GO TO 5Ø
1Ø    I=1
2Ø    IF(X.LE.XX(I+1)) GO TO 3Ø
      I=I+1
```

```
      GO TO 20
   30 FOFX=YY(I)+((YY(I+1)-YY(I))/(XX(I+1)-XX(I)))*(X-XX(I))
      RETURN
   40 I=I-1
      GO TO 30
   50 I=I+1
      GO TO 30
   60 FOFX=YY(I)
      RETURN
   85 XM=X*.3048
      IF(XM.GT.45.6 .AND. XM .LT. 1000.) GO TO 86
      IF(XM.GE. 1000.) GO TO 87
      A=(77/(130 + XM))
      B=(45.6/(XM+.002))
      VM=A+(.036*ALOG(B))
      GO TO 88
   86 VM=(77/(130+XM))
      GO TO 88
   87 VM=.068
   88 FOFX=VM*3.2806
      RETURN
      END
C
      SUBROUTINE DAUX(S,IN,DE)
      DIMENSION DE(5)
      COMMON /BLK4/ DRAG,WPLA,WPLB,FFTANG,DRIFT,TREFC,PC
      COMMON /BLK7/ FAE(100,15),AE(100,15),NAE(100),JAM
      REAL IN(5)
      CALL CUR(IN(4),COFY)
      IF(NAE(JAM).EQ.1) AEC=AE(JAM,1)
      IF(NAE(JAM).GT.1) CALL STRETH(IN(1),AEC)
      CREL=COFY-DRIFT
      CABSC=CREL*ABS(CREL)
      E=(IN(1)-TREFC)/AEC
      DE(5)=1.+E
      PCE=(1./(1.+E))**PC
      DRAP=DRAG*PCE
      F2=PCE*PCE*DE(5)
      WPUL=WPLA-WPLB*F2
      DE(3)=-COS(IN(2))*DE(5)
      DE(4)=SIN(IN(2))*DE(5)
      DE(1)=DPAP*CABSC*SIGN(FFTANG,COS(IN(2)))*DE(5)-WPUL*SIN(IN(2))
      DE(2)=-(DPAP*CABSC*SIN(IN(2))*ABS(SIN(IN(2))*DE(5))+WPUL*
     &COS(IN(2)))/IN(1)
      RETURN
      END
C
      SUBROUTINE KUTMER(N,T,Y0,EPS,H,FIRST,HCX, A)
C       KUTMER ROUTINE REVISED FOR IVODE    JAN 30,1964
      DIMENSION Y0(23),Y1(23),Y2(23),F0(23),F1(23),F2(23)
      IF(FIRST)20,10,20
   10 HC=H
      IPLOC=1
      FIRST=1.
   20 LOC=0
      HCX=HC
   32 CALL DAUX (T,Y0,F0)
   39 DO 40 I=1,N
   42 Y1(I)=Y0(I)+(HC/3.)*F0(I)
      CALL DAUX (T+HC/3.,Y1,F1)
      DO 50 I=1,N
   50 Y1(I)=Y0(I)+(HC/6.)*F0(I)+(HC/6.)*F1(I)
      CALL DAUX (T+HC/3.,Y1,F1)
      DO 60 I=1,N
```

```
 60   Y1(I)=Y0(I)+HC/8.*F0(I)+.375*HC*F1(I)
      CALL DAUX (T+HC/2.,Y1,F2)
      DO 70 I=1,N
 70   Y1(I)=Y0(I)+HC/2.*F0(I)-1.5*HC*F1(I)+2.*HC*F2(I)
      CALL DAUX (T+HC,Y1,F1)
      DO 80 I=1,N
 80   Y2(I)=Y0(I)+HC/6.*F0(I)+.66666667*HC*F2(I)+(HC/6.)*F1(I)
      INC=0
      DO 110 I=1,N
      ZZZ=ABS (Y1(I))-A
      IF(ZZZ) 85,87,87
 85   ERROR = ABS(.2*(Y1(I)-Y2(I)) )
      IF(ERROR-A)100,100,90
 87   ERROR=ABS (.2-.2*Y2(I)/Y1(I))
      IF (ERROR-EPS)100,100,90
 90   CONTINUE
      KYSCIE=12
      CZATP=2.**KYSCIE
      [illegible]CZATP*ABS(HC)-ABS(H)
      IF [illegible] 91,95,95
 91   WRITE(6,92) T,ERROR,I
 92   FORMAT(21H RELATIVE ERROR AT X= 1P1E12.3,3H IS F10.6/ 6H I= ,I4)
      [illegible]
      [illegible]
 95   [illegible]
      [illegible]*IPLOC
      [illegible]LOC
      [illegible]
      [illegible]
100   IF([illegible]64.-EPS)110,110,101
101   [illegible]
110   CONTINUE
111   T=T+HC
      DO 112 I=1,N
112   [illegible]=Y2(I)
      LOC=LOC+1
      IF(LOC-IPLOC)120,210,210
120   IF([illegible])210,130,210
130   IF(L[illegible]-(LOC/2)*2)210,140,210
140   IF(IPLOC-1)210,210,200
200   HC=2.*HC
      LOC=LOC/2
      IPLOC=IPLOC/2
210   IF(IPLOC-LOC)20,220,30
220   RETURN
      END
C
C   THIS ROUTINE CALCULATES LIFT AND DRAG ON INTERMEDIATE BODIES.
      SUBROUTINE BODY(V,PHIH,CDA,N,W,LIFT,DRAG,IPUN,J)
      DIMENSION WI(100)
      COMMON /[illegible]/ PHIM(10,7),U(10,10),L(10,10,7),D(10,10,7),NCOD(100),
     &NPHI(10),NU(10)
      REAL L,LL,LU,LIFT
      IF([illegible].GT.1)GO TO 5
      WI([illegible])=[illegible]
  5   [illegible]=WI([illegible])
      I=0
      K=0
      VA=ABS(V)
      PHIV=90.-PHIH
      PHIMIN=PHIM(N,1)
      PHIMAX=PHIM(N,NPHI(N))
      UMIN=U(N,1)
      UMAX=U(N,NU(N))
```

```
C     USE FRESH WATER DENSITY
      RHO=1.94
      IF(PHIV.LE.PHIMIN)GO TO 20
      K=1
      IF(PHIV.GE.PHIMAX)GO TO 16
 13   IF((PHIV.GT.PHIM(N,K)).AND.(PHIV.LE.PHIM(N,K+1)))GO TO 18
      K=K+1
      GO TO 13
 16   K=NPHI(N)
      GO TO 20
 18   RPHI=(PHIV-PHIM(N,K))/(PHIM(N,K+1)-PHIM(N,K))
 20   IF(VA.LE.UMIN)GO TO 30
      I=1
      IF(VA.GE.UMAX)GO TO 26
 23   IF((VA.GT.U(N,I)).AND.(VA.LE.U(N,I+1)))GO TO 28
      I=I+1
      GO TO 23
 26   I=NU(N)
      GO TO 30
 28   RV=(VA-U(N,I))/(U(N,I+1)-U(N,I))
 30   IF((K.GE.NPHI(N)).OR.(K.LT.1))GO TO 32
      IF((I.GE.NU(N)).OR.(I.LT.1))GO TO 60
C
C       ........BOTH PHIV AND VA ARE WITHIN THE LIFT/DRAG TABLE
C
      LL=RPHI*(L(N,I,K+1)-L(N,I,K))+L(N,I,K)
      LU=RPHI*(L(N,I+1,K+1)-L(N,I+1,K))+L(N,I+1,K)
      LIFT=RV*(LU-LL)+LL
      DL=RPHI*(D(N,I,K+1)-D(N,I,K))+D(N,I,K)
      DU=RPHI*(D(N,I+1,K+1)-D(N,I+1,K))+D(N,I+1,K)
      DRAG=RV*(DU-DL)+DL
      GO TO 100
 32   IF((I.GE.NU(N)).OR.(I.LT.1))GO TO 40
C
C       .....VA IS WITHIN THE LIFT/DRAG TABLE, PHIV IS NOT
C
      IF(K.LT.1) K=1
      LIFT=RV*(L(N,I+1,K)-L(N,I,K))+L(N,I,K)
      DRAG=RV*(D(N,I+1,K)-D(N,I,K))+D(N,I,K)
      GO TO 100
C
C       .....BOTH VA AND PHIV ARE OUTSIDE THE LIFT/DRAG TABLE
C
 40   IF(I.LT.1) I=1
      IF(K.LT.1) K=1
      LIFT=L(N,I,K)
      DRAG=D(N,I,K)
      GO TO 100
C
C       .....PHIV IS WITHIN THE LIFT/DRAG TABLE, VA IS NOT
C
 60   IF(I.LT.1) I=1
       LIFT=RPHI*(L(N,I,K+1)-L(N,I,K))+L(N,I,K)
       DRAG=RPHI*(D(N,I,K+1)-D(N,I,K))+D(N,I,K)
 100   CDA=2.*DRAG/(RHO*VA**2.)
       W=W-LIFT
       RETURN
       END
C
C     THIS ROUTINE CORRELATES SURFACE UNIT DRAG TO CORRESPONDING FLOW.
       SUBROUTINE VAR(F1,FLOW,HO,DB,CDB1,CD)
       COMMON /BLK8/ NFOSB,FOSB(15),VOSB(15)
       N=NFOSB
       FLOWK=.5924*FLOW
```

```
      IF(FLOWK.LE.0.) GO TO 30
      IF(FLOWK.LT.VOSB(1)) GO TO 11
      IF(FLOWK.GT.VOSB(N)) GO TO 12
      I=1
    5 IF(FLOWK.LE.VOSB(I+1)) GO TO 10
      I=I+1
      GO TO 5
   10 F2=FOSB(I)+(FOSB(I+1)-FOSB(I))/(VOSB(I+1)-VOSB(I))*
     &(FLOWK-VOSB(I))
      GO TO 13
   11 F2=FOSB(1)
      GO TO 13
   12 F2=FOSB(N)
   13 CD1=2*F2/(1.9905*FLOW*FLOW*DB*HO)
      EDRG=F2-F1
      IF(ABS(EDRG).LE..001) GO TO 30
      IF(F2-F1) 16,16,17
   16 CD=CDB1-ABS((CD1-CDB1)/2)
      GO TO 35
   17 CD=CDB1+ABS((CD1-CDB1)/2)
      GO TO 35
   30 CD=CDB1
   35 RETURN
      END
C
C   THIS ROUTINE CALCULATES AE GIVEN FORCE VALUES.
      SUBROUTINE STRETH(X,AEC)
      COMMON /BLK7/ FAE(100,15),AE(100,15),NAE(100),JAM
      J=JAM
      N=NAE(J)
      IF(X.LE.FAE(J,1)) GO TO 80
      IF(X.GE.FAE(J,N)) GO TO 90
      I=1
   30 IF(X.LE.FAE(J,I+1)) GO TO 70
      I=I+1
      GO TO 30
   70 AEC=AE(J,I)+(AE(J,I+1)-AE(J,I))/(FAE(J,I+1)-FAE(J,I))*
     &(X-FAE(J,I))
      RETURN
   80 AEC=AE(J,1)
      RETURN
   90 AEC=AE(J,N)
      RETURN
      END
```